Soft Comput (2014) 18:743–756
DOI 10.1007/s00500-013-1085-8

METHODOLOGIES AND APPLICATION

Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization

Ronghua Shang · Licheng Jiao · Yujing Ren · Lin Li · Luping Wang

Published online: 20 July 2013

Abstract The existing algorithms to solve dynamic multiobjective optimization (DMO) problems generally have difficulties in non-uniformity, local optimality and non-convergence. Based on artificial immune system, quantum evolutionary computing and the strategy of co-evolution, a quantum immune clonal coevolutionary algorithm (QICCA) is proposed to solve DMO problems. The algorithm adopts entire cloning and evolves the theory of quantum to design a quantum updating operation, which improves the searching ability of the algorithm. Moreover, coevolutionary strategy is incorporated in global operation and coevolutionary competitive operation and coevolutionary cooperative operation are designed to improve the uniformity, the diversity and the convergence performance of the solutions. The results on test problems and performance metrics compared with ICADMO and DBM suggest that QICCA has obvious effectiveness and advantages which shows great capability of evolving convergent, diverse and uniformly distributed Pareto fronts.

Keywords Dynamic multiobjective optimization (DMO) · Immune clonal operation · Quantum updating operation · Coevolutionary · Pareto optimal front

1 Introduction

In the real world, a number of multiobjective optimization problems exist in the environment changing over time, and this kind of problems are called dynamic multiobjective optimization (DMO) problem (Farina et al. 2004). As a result of the important role of DMO problem in practical application, the study of algorithms for DMO problems is very necessary (Back 1998).

Although there are some evolutionary multiobjective optimization algorithms to solve static multiobjective optimization (SMO) problems, e.g., (Deb et al. 2002; Zitzler and Thiele 2005; Nebro et al. 2007), the research and promotion of DMO problems are still on the preliminary stage. Branke (2002) presented a multi-population strategy, which is an effective method to construct dynamic intelligent optimization algorithms. Farina et al. (2004) proposed a set of DMO test problems and the correlative solution: Direction-based method (DBM). In 2005, an immune clonal algorithm for DMO was presented by Shang et al. (2005), which was based on immune clonal mechanism. As far as the authors know, the latest work reported for DMO is the dynamic competitive-cooperation coevolutionary algorithm (dCOEA) proposed by Goh and Tan (2009), which is focused on the competitive-cooperation strategy in co-evolution.

In this paper, we design a new algorithm: quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization (QICCA) which can deal with DMO problems. From the view of the adaptability of artificial immune system (AIS) and the diversity of antibodies (de Castro and Timmis 2002a,b; Gong et al. 2006; Liu et al. 2010), QICCA incorporates the quantum rotation gate strategy in quantum evolutionary algorithm (Maravall and de Lope 2007; Jiao et al. 2008) to design a quantum updating operation, which improves the searching ability in the evolutionary process and achieves better results. In addition, QICCA employs the coevolutionary mechanism in global search, which takes advantage of the multi-population strategy in Jiao et al. (2006) and Goh and Tan (2007) to

Communicated by Y.-S. Ong.

R. Shang (✉) · L. Jiao · Y. Ren · L. Li · L. Wang
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China
e-mail: rhshang@mail.xidian.edu.cn

enhance the competitive and cooperative relationship and the exchange of information among different populations as well as the enlargement of the search region of the algorithm. We consider the U-measure proposed by Leung and Wang (2003) as the competitive operation of the algorithm and design a new cooperative operation to improve the uniformity and the diversity performance of the population. The test results on five test problems in Farina et al. (2004) of the new algorithm, which are compared with DBM and ICADMO, show that QICCA has reached better results and effectively improves the performance of the population. Moreover, we test different operations on a test problem, which shows that every component of QICCA has its contribution to the performance of the population.

This paper is organized as follows: we briefly introduce the key concepts used in the field of DMO in Sect. 2. The detailed framework and the designed operations of the new algorithm QICCA are in Sect. 3. Thereafter, in Sect. 4, we present the compared results of the proposed algorithm with other two algorithms and the results with different operations, which prove the feasibility of QICCA. In Sect. 5, we give the concluding remarks and future work.

2 Problem statement and basic definitions

Definition 1 Consider a DMO problem which includes a time t, an n-dimensional decision variable $\boldsymbol{x}$, M-objectives, and the constraint conditions. The objective functions and the constraint conditions are the functions of decision variable $\boldsymbol{x}$. The mathematical model of the DMO problem is formulated as:

$$\begin{cases} \min \quad \boldsymbol{f} = \{f_1(\boldsymbol{x}, t), f_2(\boldsymbol{x}, t) \ldots, f_M(\boldsymbol{x}, t)\} \\ s.t. \quad \boldsymbol{g}(\boldsymbol{x}, t) \leq 0, \ \boldsymbol{h}(\boldsymbol{x}, t) = 0 \end{cases} \tag{1}$$

where $t \in [t_0, t_s]$ denotes the time variable, $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) \in \boldsymbol{X}(t)$ is the variable vector (decision variable), $\boldsymbol{f} = \{f_1(\boldsymbol{x}, t), f_2(\boldsymbol{x}, t), \ldots, f_M(\boldsymbol{x}, t)\} \in \boldsymbol{Y}(t)$ is the objective vector. $\boldsymbol{X}(t)$ and $\boldsymbol{Y}(t)$ are called decision space and objective space, respectively. $\boldsymbol{g}(\boldsymbol{x}, t)$ and $\boldsymbol{h}(\boldsymbol{x}, t)$ are the constraints, which determine the feasible region (Wang and Dang 2008).

Definition 2 $\boldsymbol{X}_f(t)$ denotes the set of feasible decision variables, i.e., decision variables fulfilling the constraints.

$$\boldsymbol{X}_f(t) = \{\boldsymbol{x} \in \boldsymbol{X}(t) \,|\, \boldsymbol{g}(\boldsymbol{x}, t) \leq 0, \ \boldsymbol{h}(\boldsymbol{x}, t) = 0\} \tag{2}$$

The mapping of $\boldsymbol{X}_f(t)$ is the feasible region of objective space, which is denoted as $\boldsymbol{Y}_f(t) = \boldsymbol{f}(\boldsymbol{X}_f(t))$ (Jiao et al. 2006).

Definition 3 For any two objective vectors $\boldsymbol{u}$ and $\boldsymbol{v}$

$$\begin{aligned} &\boldsymbol{u} = \boldsymbol{v}, \text{ iff } \forall i \in \{1, 2, ..., k\}, u_i = v_i \\ &\boldsymbol{u} \geq \boldsymbol{v}, \text{ iff } \forall i \in \{1, 2, \ldots, k\}, u_i \geq v_i \\ &\boldsymbol{u} > \boldsymbol{v}, \text{ iff } \boldsymbol{u} \geq \boldsymbol{v} \in \boldsymbol{u} \neq \boldsymbol{v} \end{aligned} \tag{3}$$

Definition 4 For any two decision vectors $\boldsymbol{a}$ and $\boldsymbol{b}$

$$\boldsymbol{a} \succ \boldsymbol{b} (a \text{ dominates } \boldsymbol{b}), \tag{4}$$

where for every $i \in \{1, 2, \ldots, M\}$, $f_i(\boldsymbol{a}) \leq f_i(\boldsymbol{b})$ and for at least one $k \in \{1, 2, \ldots, M\}, s.t. f_k(\boldsymbol{a}) < f_k(\boldsymbol{b})$.

Definition 5 At some moment t, a feasible decision vector $\boldsymbol{x}^* \in \boldsymbol{X}_f(t)$ is the Pareto optimal if there is no other $\boldsymbol{x} \in \boldsymbol{X}_f(t)$ that dominates $\boldsymbol{x}^*$. Moreover, the set of all Pareto optimal decision vectors in $\boldsymbol{X}_f(t)$ is called Pareto set, denoted as $\boldsymbol{X}_p(t) \cdot \boldsymbol{Y}_p(t) = \boldsymbol{f}(\boldsymbol{X}_p(t))$ is the set of all Pareto optimal objective vectors, which stands for the Pareto front (Jiao et al. 2006).

On the basis of Definition 5, there can be four different types of DMO problems according to the changes of $\boldsymbol{X}_p(t)$ and $\boldsymbol{Y}_p(t)$ (Farina et al. 2004).

- Type 1, where $\boldsymbol{X}_p(t)$ changes while $\boldsymbol{Y}_p(t)$ remains unchanged.
- Type 2, where both $\boldsymbol{X}_p(t)$ and $\boldsymbol{Y}_p(t)$ change.
- Type 3, where $\boldsymbol{X}_p(t)$ does not change, while $\boldsymbol{Y}_p(t)$ changes.
- Type 4, where both $\boldsymbol{X}_p(t)$ and $\boldsymbol{Y}_p(t)$ remain invariant, although the DMO problem is changing over time.

3 Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization

3.1 Algorithm description

Quantum immune clonal coevolutionary algorithm (QICCA) is proposed for DMO. QICCA makes use of the immune clonal function and clonal selection function in AIS to achieve the large-scale evolution towards the excellent schema of the optimal antibodies (Coello and Cortes 2002; de Castro and Timmis 2002a,b; Shang et al. 2012). With the consideration of the improvement of the uniformity and the diversity of the population, we implement the multi-population strategy in co-evolution theory (Jiao et al. 2006) and take advantage of the competitive and cooperative relationship to enhance the exchange of the information among different populations. In addition, inspired by the characteristics of quantum computing, in this new algorithm, we design a quantum updating operation, and the design of the quantum rotation angle in the quantum rotation gate (Jiao et al. 2008; Li et al. 2012b) realizes the improvement of the searching ability of the population and the better distribution of the Pareto optimal solutions. The flow chart of QICCA at a fixed time t is shown in Fig. 1.

It can be seen from Fig. 1 that QICCA mainly combines the immune clonal operation, the quantum updating operation, and the coevolutionary operation. The three main operations are described in Sects. 3.2, 3.3 and 3.4.

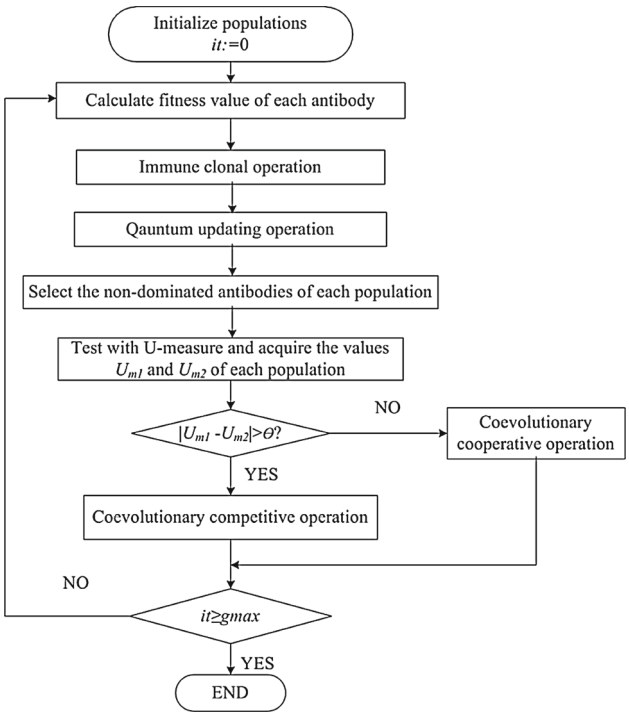


Fig. 1 Flow chart of QICCA

3.2 Immune clonal operation

3.2.1 Immune clonal proliferation operation

In immunology, the word "clone" generally means the asexual propagation, which means the asexual propagation can descend a group of identical cells from a single common ancestor, for example, members of bacterial colony can be arisen from a single original cell as the result of mitosis. Based on the theory of AIS, the clonal operation realizes the enlargement of the space and provides the foundation of the global search for attain new population and algorithms. Moreover, the clonal operation provides the condition for the restructuring strategies, such as partial restructuring and entire restructuring, which improves the information exchange among different antibodies. Actually, clone is the duplication of single antibody to multiple antibodies (Coello and Cortes 2002; Gong et al. 2008; Shang et al. 2012), which can achieve the self-adapting expansion of antibodies. Considering this advantage, the clonal proliferation operation of new algorithm improves the performance of the solutions by appropriately increasing the scale of obtained Pareto optimal solutions. Particularly, we assume that the size of the population $\boldsymbol{P}$ is n, $\boldsymbol{P} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, the implementation procedure of clonal proliferation operation is as follows:

$$\begin{aligned}\boldsymbol{P}' = T_c^C(\boldsymbol{P}) &= T_c^C\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \\ &= T_c^C\{\boldsymbol{x}_1\} + T_c^C\{\boldsymbol{x}_2\} + \cdots + T_c^C\{\boldsymbol{x}_n\} \\ &= \{\boldsymbol{x}_1^1, \boldsymbol{x}_1^2, \ldots, \boldsymbol{x}_1^q\} + \{\boldsymbol{x}_2^1, \boldsymbol{x}_2^2, \ldots, \boldsymbol{x}_2^q\} \\ &\quad + \cdots + \{\boldsymbol{x}_n^1, \boldsymbol{x}_n^2, \ldots, \boldsymbol{x}_n^q\} \end{aligned} \tag{5}$$

where T_c^C is the clonal proliferation operation, q is the clonal proportion, in QICCA, $q = 6$.

3.2.2 Clonal selection operation

The clonal selection is the contrary operation of clonal proliferation operation, which can select better antibodies to form a new population from the sub-population. Different from other selection operation of general evolutionary computing, in AIS, clonal selection means the process that selecting the better antibodies from offsprings which have been immune clonal proliferated from the antibodies and the correlative parents to generate new population (Coello and Cortes 2002; Shang et al. 2012). The implementation procedure of clonal selection operation is as follows:

$$\begin{aligned}\boldsymbol{P}'' &= T_S^C(\boldsymbol{P}') = T_S^C(\{\boldsymbol{x}_1^{1'}, \boldsymbol{x}_1^{2'}, \ldots, \boldsymbol{x}_1^{q'}\} \\ &\quad + \cdots + \{\boldsymbol{x}_n^{1'}, \boldsymbol{x}_n^{2'}, \ldots, \boldsymbol{x}_n^{q'}\}) \\ &= T_S^C(\{\boldsymbol{x}_1^{1'}, \boldsymbol{x}_1^{2'}, \ldots, \boldsymbol{x}_1^{q'}, \ldots, \boldsymbol{x}_n^{1'}, \boldsymbol{x}_n^{2'}, \ldots, \boldsymbol{x}_n^{q'}\}) \\ &= \{\boldsymbol{x}_1', \boldsymbol{x}_2', \ldots, \boldsymbol{x}_n'\} \end{aligned} \tag{6}$$

where T_S^C is the clonal selection operation. QICCA divides the antibodies into dominated ones and nondominated ones. An antibody is selected or not depends on whether it is a non-dominated solution, and only the non-dominated antibodies are selected in the antibody population (Shang et al. 2005). In QICCA, for any antibody $\boldsymbol{x}_i^{j'} \in \boldsymbol{P}'$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, q\}$, $\boldsymbol{x}_i^{j'}$, is called a non-dominated antibody in the current iteration iff:

$$\begin{aligned}&\neg\exists \boldsymbol{x}_r^{p'} \neq \boldsymbol{x}_i^{j'} \in \boldsymbol{P}', r \in \{1, 2, \ldots, n\}, \\ &\quad p \in \{1, 2, \ldots, q\} : \boldsymbol{x}_r^{p'} \succ \boldsymbol{x}_i^{j'} \end{aligned} \tag{7}$$

Otherwise, $\boldsymbol{x}_i^{j'}$ is called a dominated antibody.

Based on Eq. (7), the antibodies in $\boldsymbol{P}'$ are divided into two parts: $\boldsymbol{P}'_{non}$ with N_{non} non-dominated antibodies and $\boldsymbol{P}'_{dom}$ with N_{dom} dominated antibodies. Note that $N_{non} + N_{dom} = q * n$. In QICCA, If $N_{non} > n$, then select n antibodies according to the antibody population updating (APU) strategy proposed by Shang et al. (2012). If $N_{non} < n$, then select $n - N_{non}$ antibodies in $\boldsymbol{P}'_{non}$ randomly to compose a population denoted by $\boldsymbol{P}'_{n-N_{non}}$ and $\boldsymbol{P}'' = \boldsymbol{P}'_{non} \cup \boldsymbol{P}'_{n-N_{non}}$.

3.3 Quantum updating operation

In order to improve the search ability of the algorithm, we design a new operation called quantum updating operation, which can attain global detection and local exploitation abil-

ity. This ability can achieve the intelligence when implementing the searching ability, that is, the operation can decide to choose global detection or local exploitation when running the program. Currently we consider the population evolutes forward under the operation of the evolutionary algorithm. Therefore, the algorithm is designed to take local exploitation operation after the global detection operation.

As a result of the implementation of the basic functions of immune clone mechanism in QICCA, the real coding is adopted to design quantum updating operation. At first, the control variable is mapped to the unit interval. Afterwards, we will calculate the rotation angle and the direction of the rotation. Finally, the variables are mapped to the original interval. Generally, the expressive form of the quantum rotation gate (Pulmannnova 2001; Yu et al. 2010; Li et al. 2012a,b) is:

$$G(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \tag{8}$$

where φ is the rotation angle of $G(\varphi)$.

Through the implement of $G(\varphi)$, for some quantum bit $\begin{bmatrix} \alpha^i \\ \beta^i \end{bmatrix}$, the new quantum bit will be changed into:

$$\begin{bmatrix} \alpha^{i'} \\ \beta^{i'} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \times \begin{bmatrix} \alpha^i \\ \beta^i \end{bmatrix} \tag{9}$$

In this paper, the rotation angle of the quantum updating operation is designed as:

$$\varphi = \lambda \times d(i, j) \tag{10}$$

where, $d(i, j) \in \{-1, 0, 1\}$ is the searching direction of the rotation, which changes randomly. $d(i, j) = 1$ represents the anticlockwise searching direction, $d(i, j) = -1$ indicates the clockwise searching direction and $d(i, j) = 0$ means there is no changes. The illustration is shown in Fig. 2.

In Fig. 2, λ is the self-adapting step size, which determines the searching precision of the algorithm. We implement the strategy that adjusting the step size dynamically near the better solutions to strengthen the searching ability of the algorithm. $U(.\ ,\ .)$ is the uniform random number. Particularly, we define the λ as follows:

$$\lambda = U\left(k \times b^{\frac{it-1}{g\max-1}}, b^{\frac{it-1}{g\max-1}}\right) \tag{11}$$

In this formula, *it* is the current iteration number, while *gmax* is the total iteration number, both *it* and *gmax* determine the searching range of λ. When *it* is quite small, the self-adapting step size will be very large, and the operation can achieve a comprehensive fast global search, of which the searching range is [0, k]; when *it* gradually increases to *gmax*, the self-adapting step size will reduce to a small value and the operation can realize the improvement of local searching ability by reducing the step size near better antibodies. The searching range of local search is [0, b].

Due to the real coding applied in the new algorithm, the practical form of the quantum rotation gate in QICCA is:

$$G(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ 0 & 0 \end{bmatrix} \tag{12}$$

Based on the analysis above, we can get the procedure of quantum updating operation as follows:

Algorithm 1: Quantum updating operation

For a decision variable $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, where $x_i \in [lower, upper]$, $i = 1, 2, \ldots, n$.

Step 1: Every x_i is mapped into [0, 1] to get $x_{i[0,1]} = \frac{x_i - lower}{upper - lower}$, $i = 1, 2, \ldots, n$.

Step 2: Implement quantum real coding to get: $\boldsymbol{x}' = \begin{pmatrix} x_{1[0,1]} & x_{2[0,1]} & \cdots & x_{n[0,1]} \\ \sqrt{1-(x_{1[0,1]})^2} & \sqrt{1-(x_{2[0,1]})^2} & \cdots & \sqrt{1-(x_{n[0,1]})^2} \end{pmatrix}$.

Step 3: For some bit i of $\boldsymbol{x}'$, we implement quantum rotation operation, i.e., computing the rotation angle $\varphi = \lambda \times d(i, j)$ and the random rotation direction $d(i, j)$ to get: $x'_{i[0,1]} = \cos(\varphi) \times x_{i[0,1]} - \sin(\varphi) \times \sqrt{1-(x_{i[0,1]})^2}$, thus $\boldsymbol{x}'$will be updated into $\boldsymbol{x}''$.

Step 4: If the updated bit is out of [0, 1], revise it by dichotomy to get: $\boldsymbol{x}'_{i[0,1]} = \frac{(x'_{i[0,1]} + x_{i[0,1]})}{2}$, repeat the operation until the bit is within [0, 1].

Step 5: The updated antibody $\boldsymbol{x}''$ will be mapped into $[lower, upper]$ by computing $\boldsymbol{x}'''_i = \boldsymbol{x}''_i \times (upper - lower) + lower$. After this step, we will get the final updated $\boldsymbol{x}'''$.

As a consequence, we can see that the quantum rotation angle of the quantum rotation gate can update by the dynamic change of the self-adapting step size.

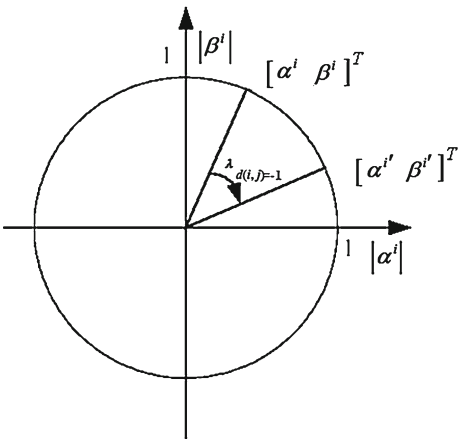


Fig. 2 The quantum updating operation

3.4 Coevolutionary operation

The resource in nature is very limited; therefore, the worse populations will be obsoleted via competition, while the better populations will be conserved for evolution. In addition,

there is another relationship among different populations, which is called the cooperation relationship. Although the individuals of various populations are quite different, the environment of the populations is similar. Thus, the consideration of the cooperation for diverse populations is necessary, which can contribute to the evolution for all populations. Nevertheless, most evolutionary algorithms only consider one population, which will overemphasize the evolution of the antibodies within the population that the Pareto optimal solutions could not spread on the Pareto front. On behalf of avoiding the loss of the diversity of the Pareto optimal solutions, QICCA employs the coevolutionary operation to maintain its performance. Co-evolution is based on the mutual restrictive, interdependent and mutual coordinated relationship of different populations (Jiao et al. 2006). In coevolutionary operation, the operations take advantage of the competitive and cooperative relationship among multiple populations. In QICCA, the difference of the performance for different populations is taken into account to realize coevolutionary operation. In this operation, we have implemented two operations: competitive operation and cooperative operation. The competitive operation is designed to improve the uniformity of the algorithm, while the cooperative operation is designed to improve the diversity of the algorithm. The details of two operations are shown in Sects. 3.4.1 and 3.4.2 respectively.

3.4.1 Coevolutionary competitive operation

For the sake of improving the performance of the optimal solutions for the population, we design a coevolutionary competitive operator which uses the U-measure in Wang and Dang (2008) as the criterion. U-measure tests the uniformity of the Pareto optimal solution distribution. U-measure is stated in Table 1.

As the U-measure value of the population stands for the uniformity and the spread of the Pareto optimal solutions, thus the less value means that the distribution of the solutions is more uniform and more extensive. In QICCA, we use the difference between the U-measure values of two populations as the criterion which determines the selection of the operation, i.e., when the absolute value of the difference ($|U_{m1} - U_{m2}|$) is greater than a proper value θ, there is a clear difference of the uniformity between two populations and we will implement coevolutionary competitive operation, otherwise coevolutionary cooperative operation.

Assume that there are j independently evolved populations $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, ..., $\boldsymbol{P}_j$, the U-measure of the ith population $\boldsymbol{P}_i$ is denoted as U_{mi}. The coevolutionary competitive operation is formulated as:

$$\boldsymbol{P}_{new} = \boldsymbol{P}_t, U_{mt} = \min_{2 \le i \le j} U_{mi},\ t \in \{2, 3, \ldots, j\} \tag{13}$$

where $\boldsymbol{P}_{new}$ is the new population, $\boldsymbol{P}_t$ is the population with the smallest U-measure value among $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_j$.

Note that the new population is the population with better uniformity and spread after the annexation operation, the coevolutionary competitive operation can select better population and enhance the performance of the Pareto optimal solutions.

3.4.2 Coevolutionary cooperative operation

According to what is stated above, when the absolute value of the difference ($|U_{m1} - U_{m2}|$) is less than a proper value θ, we will implement coevolutionary cooperative operation (Jiao et al. 2006). With the consideration of the absolutely independent evolvement of the two populations $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, QICCA makes use of the search in different regions in decision space to realize the coevolutionary cooperative operation. Particularly, the new population $\boldsymbol{P}_{new} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$ generated from the cooperative operation is designed as:

$$\boldsymbol{y}_i = \frac{(\boldsymbol{y}_i^1 + \boldsymbol{y}_i^2)}{2} \quad i = 1, 2, \ldots, n \tag{14}$$

where $\boldsymbol{y}_i^1 = \boldsymbol{r}_i^2 + U(-1, 1) \cdot (\boldsymbol{r}_i^2 - \boldsymbol{x}_i^1)$, $\boldsymbol{y}_i^2 = \boldsymbol{r}_i^1 + U(-1, 1) \cdot (\boldsymbol{r}_i^1 - \boldsymbol{x}_i^2)$, $U(.,.)$ is the uniform random numbers, $\boldsymbol{x}_i^1$ and $\boldsymbol{x}_i^2$ are the random antibodies of P_1 and P_2. $\boldsymbol{r}_i^1$ and $\boldsymbol{r}_i^2$ are the antibodies randomly chosen from $\boldsymbol{X}_{p1}(t)$ and $\boldsymbol{X}_{p2}(t)$.

With the coevolutionary cooperative operation, two populations can exchange information and enlarge the search

Table 1 U-measure

Problem	Two objectives problem	M objectives problem
U-measure (U_m)	$U_m = d_{std} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(d_{i,i+1} - d_{mean})^2}$ where $d_{mean} = \frac{1}{N+1}\sum_{i=0}^{N} d_{i,i+1}$	$U_m = d_{std} = \sqrt{\frac{1}{2MN-1}\sum_{g\in T}\sum_{r=1}^{2M}(d_r - d_{mean})^2}$ where $d_{mean} = \frac{1}{2MN}\sum_{g\in T}\sum_{r=1}^{M}(d_{2r-1} - d_{2r})$

where $d_{i,i+1}$ is the distance of every pair of adjacent points and d_{mean} is the mean distance. T is a set of Pareto solutions in the objective space, each point $g \in T, r \in [1, M]$ and d_r is the distance of this point to its neighbors. U_m can measure the uniformity and spread for the points in T. For a more detailed description the interested reader is referred to Leung and Wang (2003) and Wang and Dang (2008)

region of the algorithm, which makes the most of the difference between various populations to increase the diversity of the population.

Based on the principles introduced above, the procedure of coevolutionary operation is shown as follows:

Algorithm 2: Coevolutionary operation

Given two populations $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ and get the Pareto optimal solutions $\boldsymbol{POS}_1$ for $\boldsymbol{P}_1$ and $\boldsymbol{POS}_2$ for $\boldsymbol{P}_2$; an empty population $\boldsymbol{P}_3$; parameter θ; the number of the antibodies randomly chosen for Pareto neighborhood k; $i = 1$.

Step 1: Select two antibodies randomly from $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, denoted by $\boldsymbol{x}_i^1$ and $\boldsymbol{x}_i^2$ and select two antibodies optimal solutions randomly from $\boldsymbol{POS}_1$ and $\boldsymbol{POS}_2$, denoted by $\boldsymbol{r}_i^1$ and $\boldsymbol{r}_i^2$;

Step 2: Calculate the U-measure values of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, denoted as U_{m1} and U_{m2};

Step 3: If $|U_{m1} - U_{m2}| \geq \theta$ turn to Step 4; otherwise, turn to Step 5

Step 4: If $U_{m1} \leq U_{m2}$, $\boldsymbol{P}_3 = \boldsymbol{P}_1$; otherwise, $\boldsymbol{P}_3 = \boldsymbol{P}_2$;

Step 5: Calculate $\boldsymbol{y}_i^1 = \boldsymbol{r}_i^2 + U(-1, 1) \cdot (\boldsymbol{r}_i^2 - \boldsymbol{x}_i^1)$ and $\boldsymbol{y}_i^2 = \boldsymbol{r}_i^1 + U(-1, 1) \cdot (\boldsymbol{r}_i^1 - \boldsymbol{x}_i^2)$ and get $\boldsymbol{y}_i = \frac{(\boldsymbol{y}_i^1 + \boldsymbol{y}_i^2)}{2}$;

Step 6: If $i = k$, $\boldsymbol{P}_3 = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_k\}$; otherwise, $i = i + 1$ and turn to Step 1.

As shown in Algorithm 2, when the difference between the U-measure values for two populations is greater than θ, the algorithm implements coevolutionary competitive operation and the new population $\boldsymbol{P}_3$ is determined by the parent with less U-measure value. Forasmuch as the new population has better uniformity and spread, the quality of the solutions in the objective space is improved, while when the difference is less than θ, the algorithm will choose coevolutionary cooperative operation. Therefore, with the cooperative operation the populations will exchange their information to expend the search area of the algorithm. Hence, making the most of the difference between different populations can improve the uniformity.

3.5 The QICCA algorithm

On the basis of the designed operations above, the procedure of QICCA is followed.

Algorithm 3: Main loop for QICCA

Step 1: Initial iteration number $it := 0$, $t := 0$, the maximum iteration is $gmax$, the maximum time step is T. Initialize two populations A and B with a size of N.

Step 2: If $t < T$, go to **Step 3**; otherwise, stop.

Step 3: Calculate the fitness value of each antibody and select $\boldsymbol{Y}_f(t)$ for A and B, denoted as A_0 and B_0 respectively.

Step 4: Implement immune clonal operation on A_0 and B_0 to generate two new populations A_1 and B_1.

Step 5: Implement quantum updating operation on A_1 and B_1 and select the nondominated solutions of each population. Afterwards, reload the solutions into A_1 and B_1.

Step 6: Test A_1 and B_1 with U-measure and acquire the measure values U_{m1} and U_{m2}.

Step 7: If $|U_{m1} - U_{m2}| > \theta$, implement coevolutionary competitive operation; Otherwise, implement coevolutionary cooperative operation.

Step 8: If $it < gmax$, $it := it + 1$, go to **Step 3**; otherwise, output the resulting population at time step t and go to **Step 9**.

Step 9: $t := t + 1$, $it := 0$, go to **Step 2**.

4 Experiment results and discussion

In this paper, the test problems are taken from Farina et al. (2004). Furthermore, we select two algorithms to compare with QICCA, especially the uniformity, the diversity and the convergence performance of the solutions distribution: (1) Direction-based method (DBM), which is an immediate extension of the static direction-based search method and the multiobjective search algorithm is run in the time between one time-dependent change to another (Farina et al. 2004). (2) Immune clonal coevolutionary algorithm for dynamic multiobjective optimization (ICADMO), the one that improves the existing clonal strategies to take entire cloning and divide the antibody populations into dominated antibodies and nondominated antibodies by Pareto-dominance strategy. Next, we present the box plots (Chambers et al. 1983) on the metric S, MS and GD on 30 independent runs for these two algorithms. In addition, we give the results with different component of QICCA, which proves the feasibility of their function and contribution.

4.1 Performance metrics

At every moment, the DMO problem is a standard multiobjective optimization problem. Generally, a multiobjective optimization problem is evaluated by the uniformity metric, the diversity metric and the convergence metric. As a result, we select three metrics to test the performance of the new algorithm: space metric (S), most spread metric (MS) and generation distance metric (GD).

4.1.1 Uniformity test: space metric (spacing, S (Van Veldhuizen and Lamont 2000))

Space metric is to measure the uniformity of the distribution of the Pareto optimal solutions. Formally:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\bar{d} - d_i)^2} \tag{15}$$

where $d_i = \min_j(|f_2^i(\boldsymbol{x}) - f_1^j(\boldsymbol{x})| + |f_2^i(\boldsymbol{x}) - f_2^j(\boldsymbol{x})|)$, $i, j = 1, \ldots, n$, $\bar{d}$ is the mean value of d_i. n is the number of the antibodies on Pareto-front.

It can be seen from Eq. (15) that the smaller the metric is, the more uniform the distribution of the Pareto optimal solution will be. When S = 0, the Pareto optimal solution is the most uniform.

4.1.2 Diversity test: most spread metric (MS (Goh and Tan 2007))

Most spread metric is to measure how well the true Pareto-front (PF_{true}) is covered by the evolved Pareto-front (PF_{known}). Formally:

$$MS = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\frac{\min((\overline{PF_{true}})_i, (\overline{PF_{known}})_i) - \max((\underline{PF_{true}})_i, (\underline{PF_{known}})_i)}{(\overline{PF_{true}})_i - (\underline{PF_{true}})_i}} \tag{16}$$

where $\overline{(PF_{known})_i}$ denotes the maximum of the ith objective function on PF_{known}, while $\underline{(PF_{known})_i}$ is the minimum of the ith objective function on PF_{known}. Similarly, $\overline{(PF_{true})_i}$ and $\underline{(PF_{true})_i}$ is the maximum and the minimum of the ith objective function on PF_{true} respectively.

It can be seen from Eq. (16) that the higher the metric is, the larger the coverage area will be and the better the diversity of the solutions will be. The value MS = 1 means that the coverage area of the evolved Pareto-front over the true Pareto-front is 100 %.

4.1.3 Approximation property test: generational distance metric (GD (Van Veldhuizen and Lamont 2000))

Generational distance metric provides a good measure of the distance between the evolved Pareto-front (PF_{known}) and the true Pareto-front (PF_{true}). Mathematically, this metric is a function of antibody distance given as:

$$GD = \frac{1}{n}\left(\sum_{i=1}^{n} d_i^p\right)^{1/p} \tag{17}$$

where $p = 2$, n is the number of antibodies. d_i is the Euclidean distance between the objective function vector of the ith antibody on PF_{known} and the nearest antibody on PF_{true}.

Equation (17) suggests that a smaller value of the metric reflects a better convergence performance of the evolved Pareto-front to the true Pareto-front will be.

4.2 Parameter settings

On all test problems, the parameter values of QICCA are as follows: The size of the population: $N = 300$. Overall iteration number: $gmax = 150$; The expected number of nondominated antibody population N_n is related to the test problem. The cloning proportion: $N_c = 6$. In every test DMO problem, we define a fixed maximum runtime T. On these settings, the new algorithm can achieve relatively good results.

4.3 Results and comparisons

4.3.1 FDA1

$$\begin{cases} \min\ F_1(\boldsymbol{x}_{\mathrm{I}}) = x_1 \\ \min\ F_2(\boldsymbol{x}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{II}}) = g(\boldsymbol{x}_{\mathrm{II}}) \cdot h(F_1, g) \\ where,\ g(\boldsymbol{x}_{\mathrm{II}}) = 1 + \sum_{x_i \in \boldsymbol{x}_{\mathrm{II}}} (x_i - G(t))^2 \\ h(F_1, g) = 1 - \sqrt{\frac{F_1}{g}},\ G(t) = \sin(0.5\pi t),\ t = \frac{1}{n_\tau}\left\lfloor \frac{\tau}{\tau_T} \right\rfloor \end{cases} \tag{18}$$

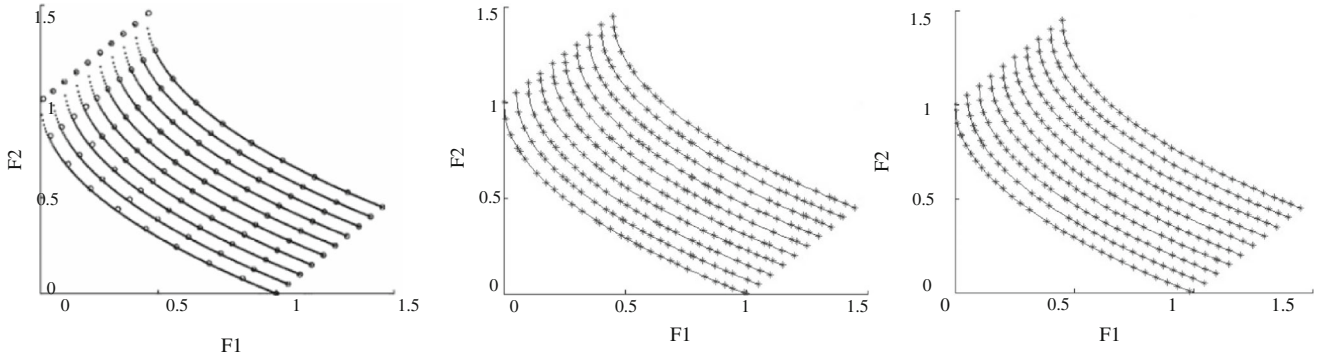
where $\boldsymbol{x}_I = (x_1) \in [0, 1]$, $\boldsymbol{x}_{II} = (x_2, \ldots, x_n) \in [-1, 1]$, $n = 20$, $\tau_T = 5$, $n_t = 10$. In this problem, $G(t)$ changes over time, hence $\boldsymbol{X}_p(t)$ will change over time as well, while $\boldsymbol{Y}_p(t)$ remains invariant. The $\boldsymbol{Y}_p(t)$ is $F_2 = 1 - \sqrt{F_1}$ at any time. The Pareto optimal solutions of each algorithm and Pareto-fronts are shown in Fig. 3.

For this problem, QICCA has shown an overall advantage over other algorithms. Particularly, it can be seen from Fig. 3 that the proposed algorithm has attained a better distribution of solutions than DBM and ICADMO. DBM fails to acquire the Pareto optimal solutions in the upper left of the Pareto-fronts, while ICADMO does not have a good uniformity.

The box plots of the metrics in Fig. 4 show that the QICCA has better metric statistical results than ICADMO on every moment. On one side, the minimum, mean value and maximum of metric S for QICCA are all smaller than those of ICADMO. On the other side, although the minimum of metric MS for QICCA on some moments is smaller than the corresponding value of ICADMO, the maximum and mean value of the proposed algorithm are both the larger ones. In addition, it is obvious that the new algorithm reaches a better result on convergence of the population, which shows the improvement of the convergence.

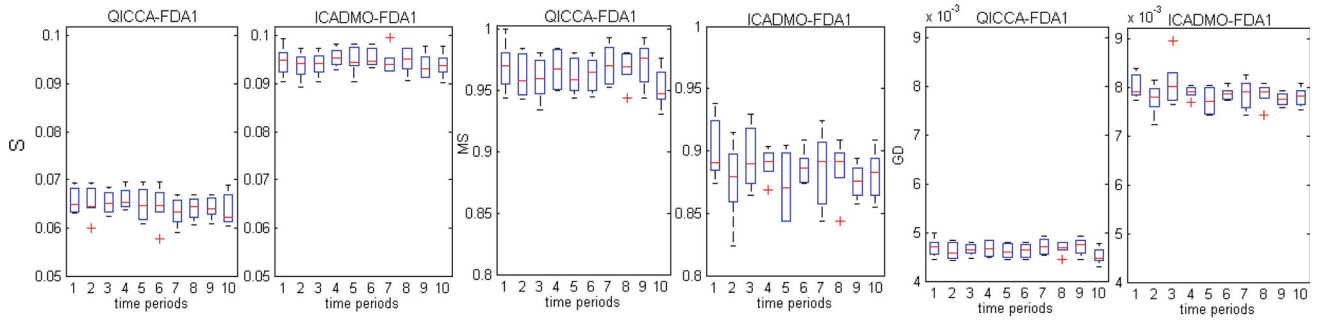
4.3.2 FDA2

$$\begin{cases} \min\ F(\boldsymbol{x}_I) = x_1 \\ \min\ F(\boldsymbol{x}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{II}}) = g(\boldsymbol{x}_{\mathrm{II}}) \cdot h(\boldsymbol{x}_{\mathrm{III}}, F_1, g) \\ where,\ g(\boldsymbol{x}_{\mathrm{II}}) = 1 + \sum_{x_i \in x_{II}} x_i^2 \\ h(\boldsymbol{x}_{\mathrm{III}}, F_1, g) = 1 - \left(\frac{F_1}{g}\right)^{(H(t) + \sum_{x_i \in x_{\mathrm{II}}} (x_i - H(t))^2)} \\ H(t) = 0.75 + 0.7\sin(0.5\pi t),\ t = \frac{1}{n_\tau}\left\lfloor \frac{\tau}{\tau_T} \right\rfloor \end{cases}, \tag{19}$$



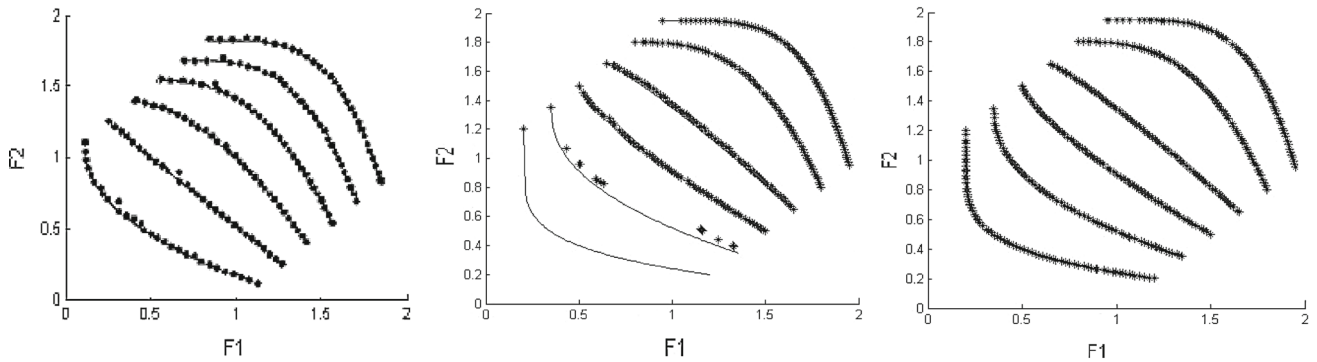
(a) Results of DBM **(b)** Results of ICADMO **(c)** Results of QICCA

Fig. 3 The $\boldsymbol{Y}_{\boldsymbol{p}}(t)$ of FDA1 and results of each algorithm



(a) Box plots of metric S **(b)** Box plots of metric MS **(c)** Box plots of metric GD

Fig. 4 *Box plots* of each metric

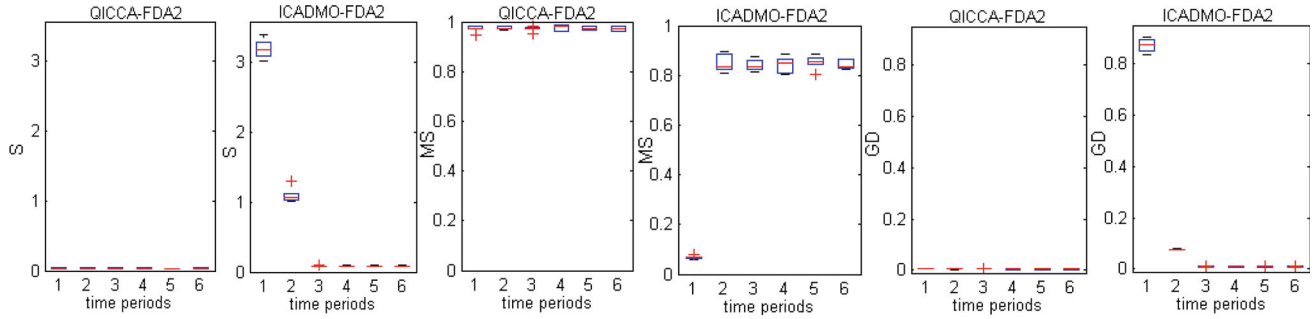


(a) Results of DBM **(b)** Results of ICADMO **(c)** Results of QICCA

Fig. 5 The $\boldsymbol{Y}_{p}(t)$ of FDA2 and results of each algorithm

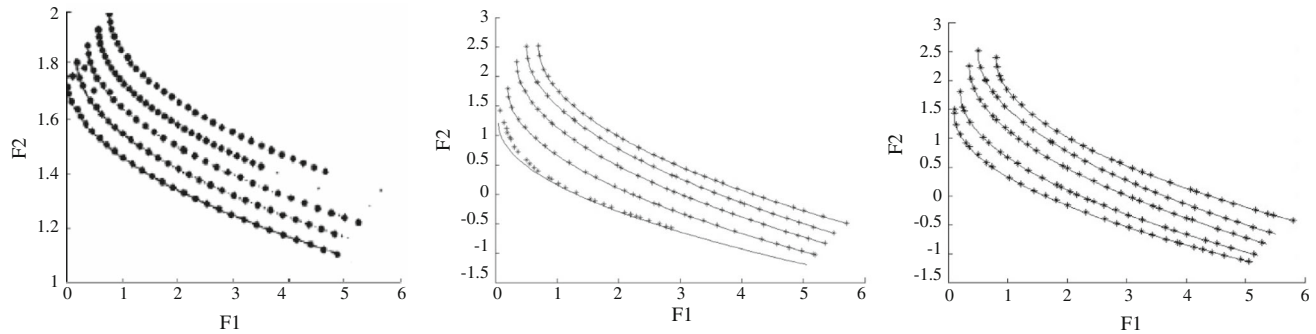
where $\boldsymbol{x}_{\mathrm{I}} = (x_1) \in [0, 1], \boldsymbol{x}_{\mathrm{II}}, \boldsymbol{x}_{\mathrm{III}} \in [-1, 1], |\boldsymbol{x}_{\mathrm{II}}| = |\boldsymbol{x}_{\mathrm{III}}| = 15$, $n = 31$, $\tau_T = 5$, $n_t = 10$. In FDA2, $\boldsymbol{Y}_p(t)$ swings from a convex shape to a non-convex shape with the change of $H(t)$, while $\boldsymbol{X}_p(t)$ remains invariant. In every moment, the $\boldsymbol{Y}_p(t)$ of FDA2 is $F_2 = 1 - F_1^{H(t)+15(1+H(t))^2}$. Figure 5 gives the Pareto optimal solutions of each algorithm and Pareto-fronts of FDA2.

The test results shown in Fig. 5 suggest that DBM and ICADMO fail to maintain good uniformity and diversity on FDA2. Moreover, the results of DBM tend to have difficulty on convergence, while the distribution of the Pareto optimal solutions on the first moment and on the second moment of ICADMO is much worse than expectation. Furthermore, ICADMO does not preserve the uniformity and the convergence that there is a large distance between the attained solutions and the true Pareto-front. QICCADOM guarantees better performance on every metric and reaches a broader spread, which achieves a successful improvement.



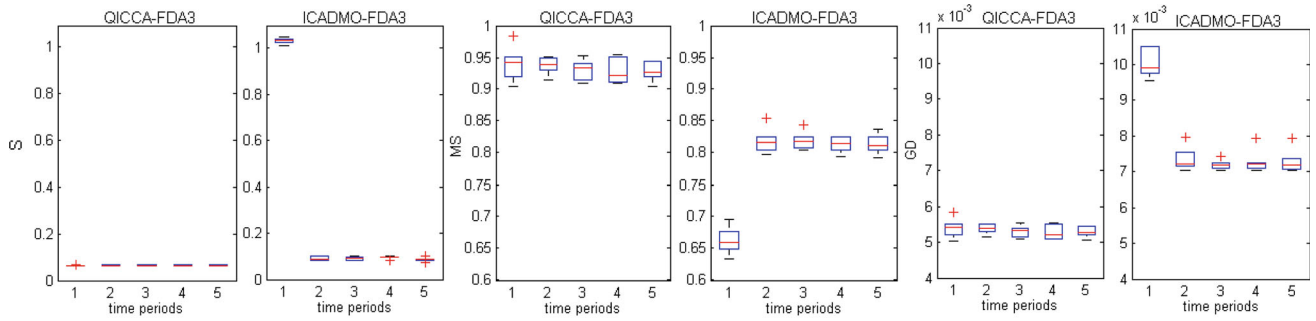
(a) Box plots of metric S **(b)** Box plots of metric MS **(c)** Box plots of metric GD

Fig. 6 *Box plots* of each metric



(a) Results of DBM **(b)** Results of ICADMO **(c)** Results of QICCA

Fig. 7 The $\boldsymbol{Y_p}(t)$ of FDA3 and results of each algorithm



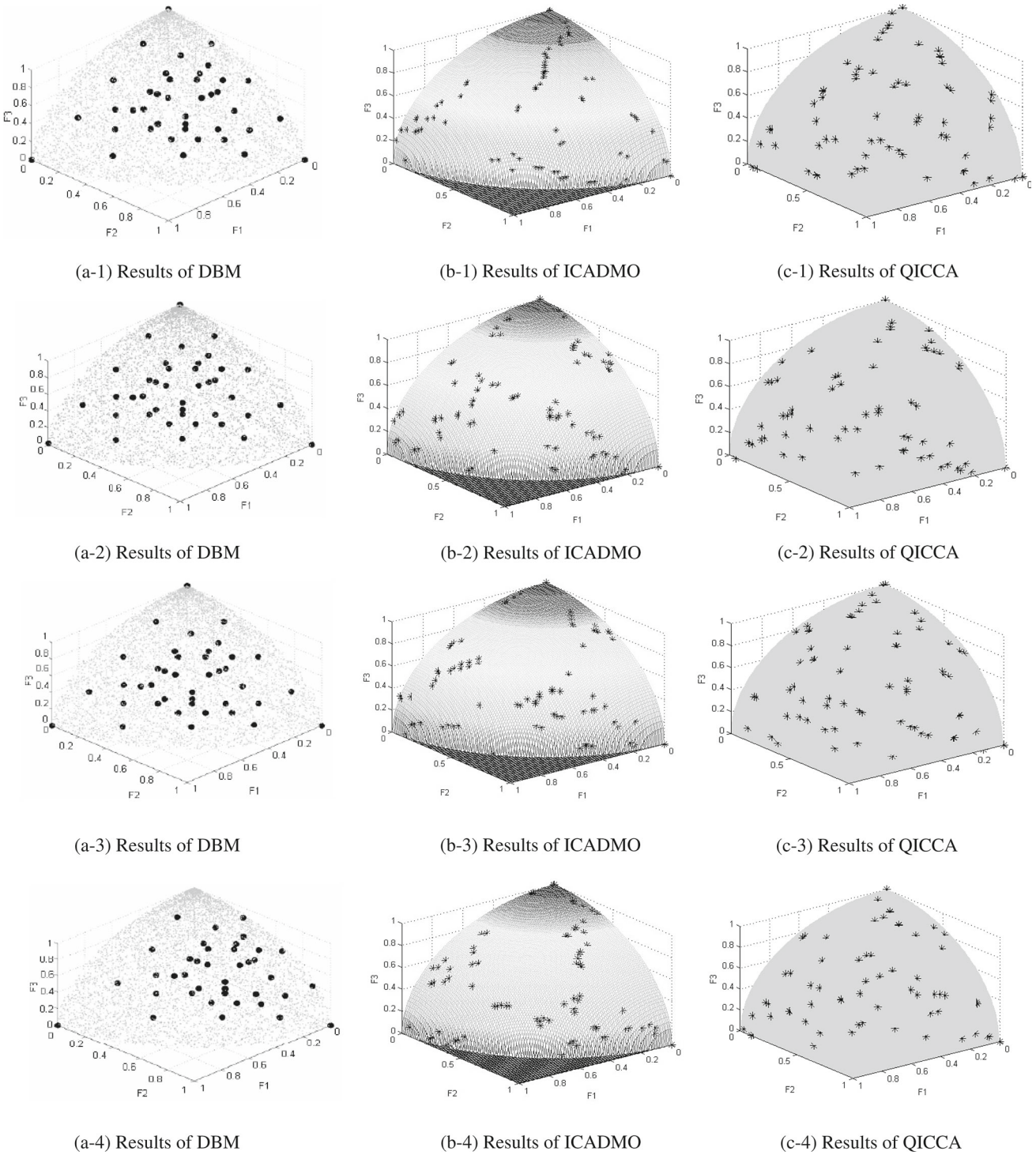
(a) Box plots of metric S **(b)** Box plots of metric MS **(c)** Box plots of metric GD

Fig. 8 *Box plots* of each metric

The box plots depicted in Fig. 6 indicate that QICCA provides similar behavior on every moment for both metric S and metric MS. The value of metric S on the first moment of the new algorithm is much smaller than the value of ICADMO and the test values on other moments are all quite small. The value of metric MS of ICADMO is between 0–0.1 on the first moment, the points generated by ICADMO cover a quite small percentage of the true Pareto-front. Additionally, the solutions of ICADMO fail to converge to the Pareto-fronts on every moment, while QICCA shows better convergence. The results suggest that the new algorithm performs better than ICADMO on FDA2.

4.3.3 FDA3

$$\begin{cases} \min\ F_1(\boldsymbol{x}_{\mathrm{I}}) = \sum_{x_i \in x_I} x_i^{F(t)} \\ \min\ F_2(\boldsymbol{x}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{II}}) = g(x_{\mathrm{II}}) \cdot h(F_1, g) \\ where,\ g(\boldsymbol{x}_{\mathrm{II}}) = 1 + G(x) + \sum_{x_i \in x_{\mathrm{II}}} (x_i - G(t))^2 \\ h(F_1, g) = 1 - \sqrt{\frac{F_1}{g}},\ G(t) = \sin(0.5\pi t), \\ F(t) = 10^{2\sin(0.5\pi t)},\ t = \frac{1}{n_\tau}\left\lfloor \frac{\tau}{\tau_T} \right\rfloor \end{cases} \tag{20}$$



(a-1) Results of DBM (b-1) Results of ICADMO (c-1) Results of QICCA

(a-2) Results of DBM (b-2) Results of ICADMO (c-2) Results of QICCA

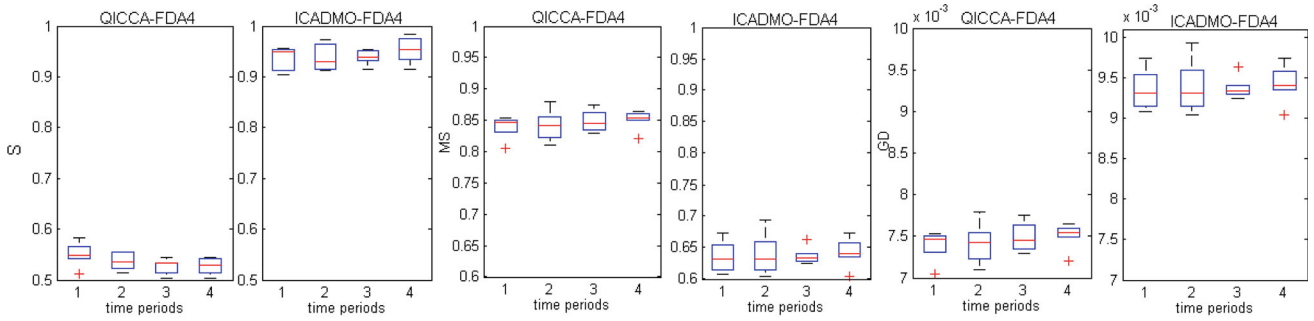
(a-3) Results of DBM (b-3) Results of ICADMO (c-3) Results of QICCA

(a-4) Results of DBM (b-4) Results of ICADMO (c-4) Results of QICCA

Fig. 9 The $\boldsymbol{Y}_p(t)$ of FDA4 and results of each algorithm

where $\boldsymbol{x}_{\mathrm{I}} \in [0,1], \boldsymbol{x}_{\mathrm{II}} \in [-1,1], n = 31, \tau_T = 5, n_t = 10, |\boldsymbol{x}_I| = 5, |x_{\mathrm{II}}| = 25$. In this test problem, both $\boldsymbol{X}_p(t)$ and $\boldsymbol{Y}_p(t)$ change over time, and the solutions density of $\boldsymbol{Y}_p(t)$ varies over time as well. In any time, the $\boldsymbol{Y}_p(t)$ is $F_2 = (1+G(t)) \times (1-\sqrt{F_1})$. The Pareto optimal solutions of each algorithm and Pareto-fronts of FDA3 are shown in Fig. 7.

It can be seen from Fig. 7 that QICCA provides a better distribution of points than other two algorithms. On the forth moment, DBM fails to find the Pareto optimal solutions. ICADMO seems to have difficulties to preserve the balance of uniformity and the diversity. Figure 7 also shows a better convergence performance of QICCA.



(a) Box plots of metric S **(b)** Box plots of metric MS **(c)** Box plots of metric GD

Fig. 10 *Box plots* of each metric

Comparisons with each measure in Fig. 8 clearly suggest that the ICADMO does not have good results on each metric on the first moment, and could not maintain good performance on other moments. In contrast, the value of metric S for the new algorithm is between 0–0.05, which is much smaller than ICADMO. The value of metric MS for QICCA is greater than 0.9 on every moment, while the value for ICADMO is just 0.6–0.85. The conclusion shows that QICCA has improved a lot on every aspect.

4.3.4 FDA4

$$\begin{cases} \min_x F_1(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\prod_{i=1}^{M-1}\cos\left(\frac{x_i\pi}{2}\right) \\ \min_x F_2(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\left(\prod_{i=1}^{M-2}\cos\left(\frac{x_i\pi}{2}\right)\right)\sin\left(\frac{x_{M-1}\pi}{2}\right) \\ \min_x F_M(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\sin\left(\frac{x_1\pi}{2}\right) \\ where,\ g(\boldsymbol{x}_{\mathrm{II}}) = \sum_{x_i\in x_{II}}(x_i - G(t))^2, \\ \quad k = 2:M-1, G(t) = |\sin(0.5\pi t)|, t = \frac{1}{n_\tau}\left\lfloor\frac{\tau}{\tau_T}\right\rfloor \end{cases} \tag{21}$$

where $\boldsymbol{x}_{\mathrm{II}} = (x_M, \ldots, x_n), x_i \in [0,1](i = 1:n), n = 12, \tau_T = 5, n_t = 10, |\boldsymbol{x}_{\mathrm{II}}| = 10$. In FDA4, the $\boldsymbol{X}_p(t)$ changes with t, while $\boldsymbol{Y}_p(t)$ preserves unchanged. The $\boldsymbol{Y}_p(t)$ is $\sum_{i=1}^{M}(F_i^*)^2 = 1$. The Pareto optimal solutions of each algorithm and Pareto-fronts of FDA4 are shown in Fig. 9. We have selected four moments of the test results for each algorithm.

For this three objectives problem, with the changing time, DBM tends to have difficulties on preserving a good diversity and a broad spread. It is also observed that the attained solutions of ICADMO fail to maintain the uniformity of population. The proposed QICCA shows better performance on every moment.

In Fig. 10, the box plots of every metric show the clear advantages of QICCA. The test result of metric S provided by Fig. 10a presents a value of 0.5–0.6 for metric S of the proposed algorithm, while the value of ICADMO is greater than 0.9. In Fig. 10b, the MS value of QICCA is between 0.8–0.9, which is greater than ICADMO. Moreover, the value of GD of the new algorithm is between 0.6–0.7, while the maximum, mean value and minimum of ICADMO are all greater than QICCA.

4.3.5 FDA5

$$\begin{cases} \min_x F_1(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\prod_{i=1}^{M-1}\cos\left(\frac{y_i\pi}{2}\right) \\ \min_x F_2(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\left(\prod_{i=1}^{M-2}\cos\left(\frac{y_i\pi}{2}\right)\right)\sin\left(\frac{y_1\pi}{2}\right) \\ \min_x F_M(\boldsymbol{x}) = (1+g(\boldsymbol{x}_{\mathrm{II}}))\sin\left(\frac{y_1\pi}{2}\right) \\ where,\ g(\boldsymbol{x}_{\mathrm{II}}) = G(t) + \sum_{x_i\in x_{\mathrm{II}}}(x_i - G(t))^2; \\ G(t) = |\sin(0.5\pi t)|; F(t) = 1 + 100\sin^4(0.5\pi t) \\ y_i = x_i^{F(t)},\ for\ i = 1:M-1; t = \frac{1}{n_\tau}\left\lfloor\frac{\tau}{\tau_T}\right\rfloor \end{cases} \tag{22}$$

where $\boldsymbol{x}_{\mathrm{II}} = (x_M, \ldots, x_n), x_i \in [0,1](i = 1:n), n = 12, \tau_T = 5, n_t = 10, |\boldsymbol{x}_{\mathrm{II}}| = 10$. For FDA5, the $\boldsymbol{Y}_p(t)$ is three-dimensional. Both $\boldsymbol{X}_p(t)$ and $\boldsymbol{Y}_p(t)$ change with time, and the density of the solutions on $\boldsymbol{Y}_p(t)$ varies with time. In any time, the $\boldsymbol{Y}_p(t)$ is. $\sum_{i=1}^{M}(F_i^*)^2 = 1 + G(t)$. The Pareto optimal solutions of each algorithm and Pareto-fronts of FDA5 are shown in Fig. 11. We have selected four moments of the test results for each algorithm.

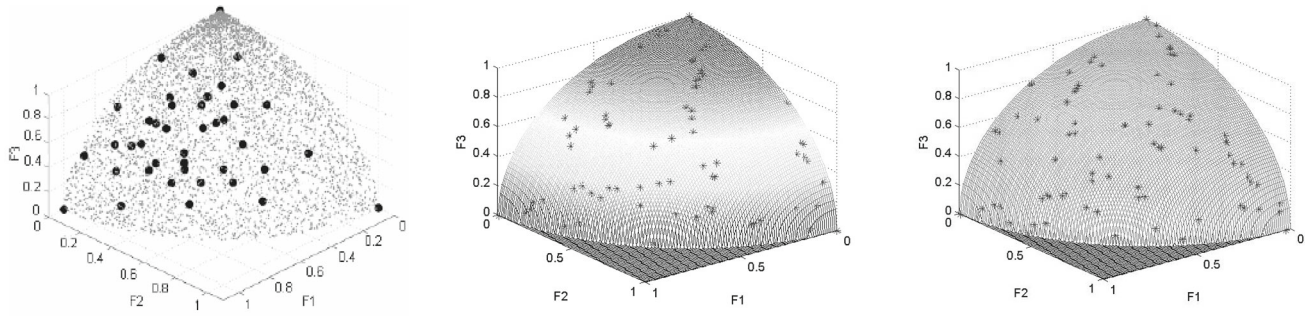
In this test problem, the solutions of ICADMO and QICCA show better diversity of DBM. However, QICCA has improved the uniformity of ICADMO, which can be seen on every moment.

In Fig. 12, the minimum, mean value and maximum of metric S for QICCA are all smaller than the results of ICADMO. In addition, the metric MS and GD of QICCA show better result than ICADMO. Therefore, QICCA has achieved better performance on FDA5.

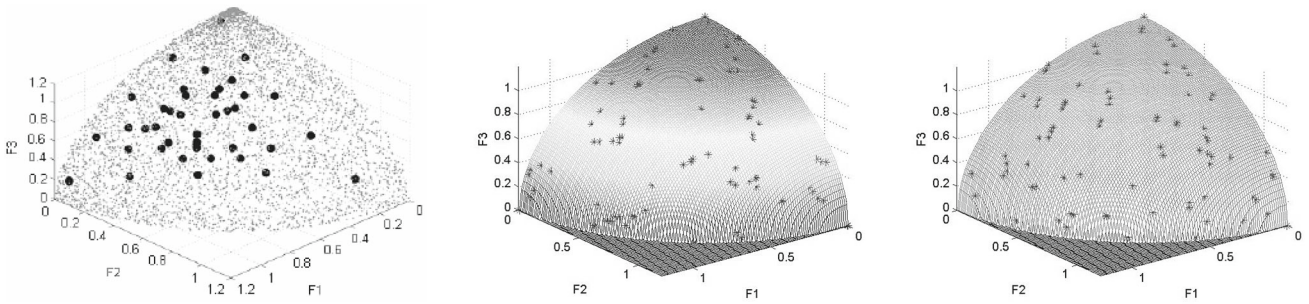
4.4 Results with different operations

In order to illustrate the necessity and function of every operation, we choose FDA2 to test .The results with different operations are shown in Fig. 13.

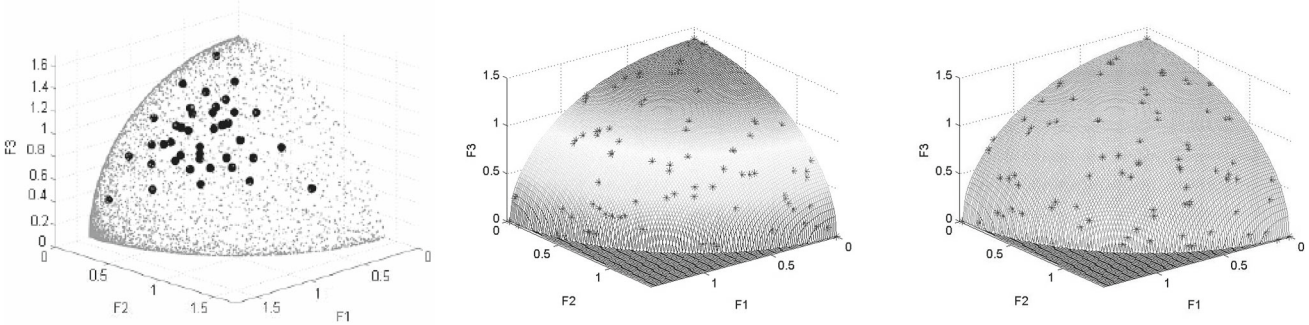
From Fig. 13a we can see that the algorithm which only employs clonal operations has the worst performance. The



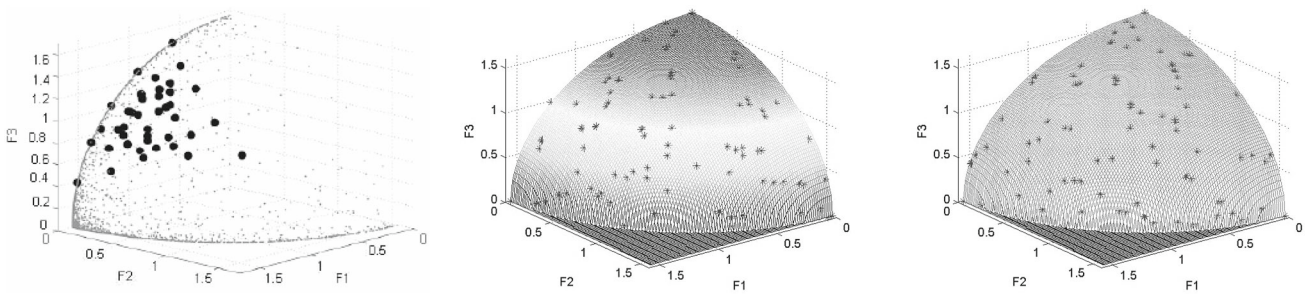
(a-1) Results of DBM (b-1) Results of ICADMO (c-1) Results of QICCA



(a-2) Results of DBM (b-2) Results of ICADMO (c-2) Results of QICCA



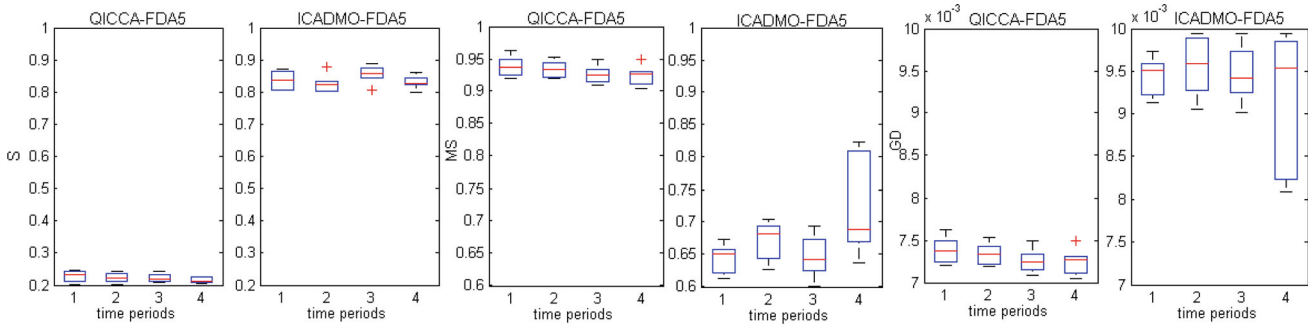
(a-3) Results of DBM (b-3) Results of ICADMO (c-3) Results of QICCA



(a-4) Results of DBM (b-4) Results of ICADMO (c-4) Results of QICCA

Fig. 11 The $\boldsymbol{Y_p}(t)$ of FDA5 and results of each algorithm

clonal operations have the ability to preserve Pareto optimal solutions on the last four moments, but they fail to converge to the true Pareto front on the first moment and the second moment. Moreover, the algorithm with only clonal operations does not have good uniformity and diversity. It is shown in Fig. 13b that the population has improved its convergence through both clonal operation and quantum updating operation. However, the solutions of the algorithm are not well distributed, especially on the first moment. It is clearly illustrated in Fig. 13c that the algorithm with clonal operations,



(a) Box plots of metric S **(b)** Box plots of metric MS **(c)** Box plots of metric GD

Fig. 12 *Box plots* of each metric

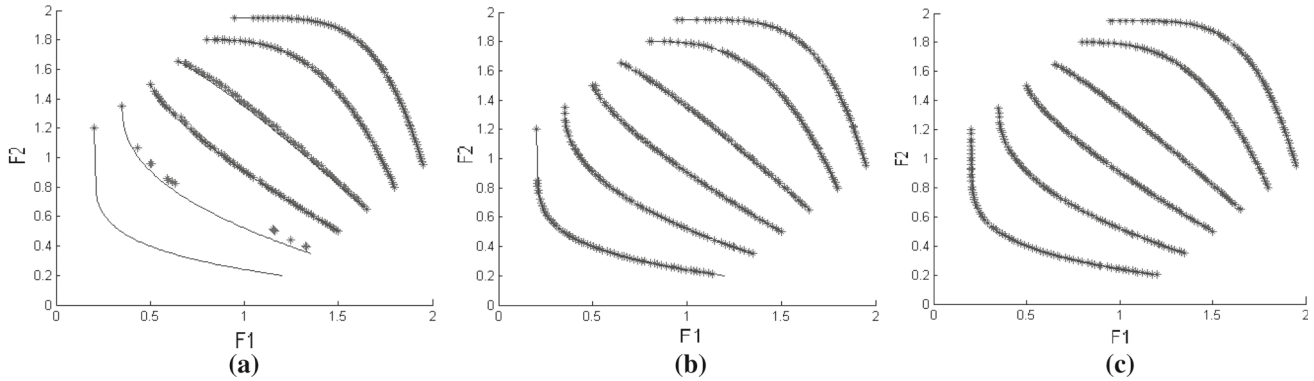


Fig. 13 Results with different operations on FDA2. **a** Result with clonal operations, **b** result with clonal operations and quantum operation, **c** result with clonal operations, quantum updating operation and coevolutionary operations

quantum updating operation and coevolutionary operations, namely QICCA has the best performance which has attained a well distributed solutions. In addition, the required solutions are all converged to their true Pareto front on every moment.

As a consequence of the experimental results discussed above, it is proved that QICCA has a better performance that the clonal operations attain better optimal ability, while the quantum updating operation effectively improves the convergence of the algorithm. Furthermore, the coevolutionary operations make the solutions more uniform and diverse.

5 Conclusion

In this paper, we have proposed QICCA, an improved dynamic multiobjective optimization algorithm that combines Immune Clonal Algorithm with the coevolutionary strategy and employed the theory of quantum immune computing. The new algorithm has designed a quantum updating operation, a coevolutionary competitive operation and a coevolutionary cooperative operation. In addition, we have revised the operations in immune clonal operation and presented an improved strategy of the performance of the population. Extensive numerical comparisons of QICCA with DBM and ICADMO have been carried on five different benchmark problems borrowed from the literature.

The key experimental results are:

- QICCA shows the best performance overall.
- DBM has good uniformity, but has difficulties to attain the boundary solutions on some problems.
- QICCA performs better than ICADMO on spacing metric, most spread metric and generation distance metric and has overwhelming advantage over DBM.
- Every component of QICCA contributes to the algorithm. The clonal operations make the algorithm attain better optimal ability, the quantum updating operation improves the searching ability and preserve the convergence of the population, and the coevolutionary operations performs better uniformity and diversity.

Furthermore, the future work is expected to concentrate on the improvement of the speed of the convergence and put the algorithm into practical optimization problem.

Acknowledgments We would like to express our sincere appreciation to the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Basic Research Program (973

Program) of China under Grant 2013CB329402, the National Natural Science Foundation of China, under Grants 61001202, 61203303,and 61272279, the National Research Foundation for the Doctoral Program of Higher Education of China, under Grants 20100203120008, the Fundamental Research Funds for the Central Universities, under Grant K5051302028, the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grant B07048, and the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

References

Back T (1998) On the behavior of evolutionary algorithms in dynamic fitness landscape. Proceedings of IEEE international conference on evolutionary computation, Anchorage, In, pp 446–451

Branke J (2002) Evolutionary optimization in dynamic environments. Kluwer Academic Publishers, Dordrecht

Chambers JM, Cleveland WS, Kleiner B et al (1983) Graphical methods for data analysis. Wadsworth Brooks/Cole, Pacific Grover

Coello CAC, Cortes NC (2002) An approach to solve multiobjective optimization problems based on an artificial immune system. In: Proceedings of 1st international conference artificial immune system. Int Center Adv Res Identificat Sci (ICARIS), pp 212–221

de Castro LN, Timmis J (2002a) Artificial immune systems: a new computational intelligence approach. Springer, Berlin, pp 1–357

de Castro LN, Timmis J (2002b) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6(3):239–251

Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8(5):425–442

Goh CK, Tan KC (2007) An investigation on noisy environments in evolutionary multiobjective optimization. IEEE Trans Evol Comput 11(3):354–381

Goh CK, Tan KC (2009) A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans Evol Comput 13(1):103–127

Gong MG, Du HF, Jiao LC (2006) Optimal approximation of linear systems by artificial immune response. Sci China Ser F Inf Sci 49(1):63–79

Gong MG, Jiao LC, Du HF et al (2008) Multiobjective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation. MIT Press, Cambridge, pp 225–255

Jiao LC, Li YY, Gong MG et al (2008) Quantum-inspired immune clonal algorithm for global optimization. IEEE Trans Syst Man Cybern Part B 38(5):1234–1253

Jiao LC, Liu J, Zhong WC (2006) An organizational coevolutionary algorithm for classification. IEEE Trans Evol Comput 10(1):67–80

Leung YW, Wang YP (2003) U-measure: a quality measure for multiobjective programming. IEEE Trans Syst Man Cybern Part A 33(3):337–343

Li YY, Shi HZ, Jiao LC et al (2012a) Quantum evolutionary clustering algorithm based on watershed applied to SAR image segmentation. Neurocomputing 87:90–98

Li YY, Xiang RR, Jiao LC et al (2012b) An improved cooperative quantum-behaved particle swarm optimization. Soft Comput 16:1061–1069

Liu RC, Sheng ZC, Jiao LC et al (2010) Immunodomaince based clonal selection clustering algorithm. IEEE congress on, evolutionary computation, pp 1–7

Maravall D, de Lope J (2007) Multi-objective dynamic optimization with genetic algorithms for automatic parking. Soft Comput 11:249–257

Nebro AJ, Alba E, Luna F (2007) Multi-objective optimization using grid computing. Soft Comput 11: 531–540

Pulmannnova S (2001) On the role of quantum structures in the foundations of quantum theory. Soft Comput, 135–136

Shang RH, Jiao LC, Gong MG et al (2005) Clonal selection algorithm for dynamic muitiobjective optimization. In: Hao Y, Liu JM, Wang YP et al. (eds) Proceedings of the 2005 international conference on computational intelligence and security. Lecture Notes in Computer Science, LNCS, vol 3801. Springer, Berlin, pp 846–851

Shang RH, Jiao LC, Liu F et al (2012) A novel immune clonal algorithm for MO problems. IEEE Trans Evol Comput 16(1):35–50

Van Veldhuizen DA, Lamont GB (2000) On measuring multiobjecitve evolutionary algorithm performance. In: Congress on evolutionary computation (CEC 2000), vol 1. IEEE Press, Piscataway, pp 204–211

Wang YP, Dang CY (2008) An evolutionary algorithm for dynamic multi-objective optimization. Appl Math Comput 205(1):6–18

Yu YF, Qian F, Liu HM (2010) Quantum clustering-based weighted linear programming support vector regression for multivariable nonlinear problem. Soft Comput 14:921–929

Zitzler E, Thiele L (2005) A simple mulf-membered evolution strategy to solve constraint optimization problems. IEEE Trans Evol Comput 9(1):1–17