J Heuristics (2011) 17: 303–351
DOI 10.1007/s10732-010-9136-0

Quantum-inspired evolutionary algorithms: a survey and empirical study

Gexiang Zhang

Received: 2 December 2009 / Revised: 1 June 2010 / Accepted: 13 June 2010 /
Published online: 29 June 2010

Abstract Quantum-inspired evolutionary algorithms, one of the three main research areas related to the complex interaction between quantum computing and evolutionary algorithms, are receiving renewed attention. A quantum-inspired evolutionary algorithm is a new evolutionary algorithm *for a classical computer* rather than for quantum mechanical hardware. This paper provides a unified framework and a comprehensive survey of recent work in this rapidly growing field. After introducing of the main concepts behind quantum-inspired evolutionary algorithms, we present the key ideas related to the multitude of quantum-inspired evolutionary algorithms, sketch the differences between them, survey theoretical developments and applications that range from combinatorial optimizations to numerical optimizations, and compare the advantages and limitations of these various methods. Finally, a small comparative study is conducted to evaluate the performances of different types of quantum-inspired evolutionary algorithms and conclusions are drawn about some of the most promising future research developments in this area.

Keywords Quantum-inspired evolutionary algorithm · Evolutionary computation · Quantum computing · Optimization

Glossary

QIEA	Quantum-inspired evolutionary algorithm
EDQA	Evolutionary-designed quantum algorithm
Q-bit	Quantum-inspired bit

This work is partially supported by the National Natural Science Foundation of China (60702026), the Scientific and Technological Funds for Young Scientists of Sichuan (09ZQ-026-040) and the Open Research Fund of Key Laboratory of Signal and Information Processing, Xihua University (SZJJ2009-003).

G. Zhang (✉)
School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, P.R. China
e-mail: zhgxdylan@126.com

EA	Evolutionary algorithm
GA	Genetic algorithm
Q-gate	Quantum-inspired gate
QEA	Quantum evolutionary algorithm
bQIEAcm	bQIEA with crossover and mutation operators
bQIEAn	bQIEA with a novel update method for Q-gates
bQIEAi	Hybrid algorithm of bQIEA and immune algorithms
DS-CDMA	Directed-sequence code-division multiple access
bQIEApso	Hybrid algorithm of bQIEA and PSO
bQIEAcga	Hybrid algorithm of bQIEA and CGA
bQIEAo	Original version of bQIEA
bQIEAh	Hybrid bQIEA
rQIEA	Real observation QIEA
EDA	Estimation of distribution algorithm
bQIEA	Binary observation QIEA
OMUD	Optimal multiuser detector
PGA	Polyploid GA
PSO	Particle swarm optimization
MFD	Matched filter detector
CGA	Conventional GA
iQIEA	QIEA-like algorithm

1 Introduction

The last twenty years have seen the application of various properties from quantum physics to building a new kind of computers, quantum computers (Nielsen and Chuang 2000; Glassner 2001a). In contrast to classical computers that deal with binary digits (*bits*), quantum computers work by manipulating quantum bits (*qubits*); these are the smallest units of information that can be stored in a two-state quantum computer (Hey 1999). Besides the usual '0' and '1' states, a qubit can also be in a superposition of these two states, so that a quantum particle may effectively be in lots of incompatible states at the same time (Nielsen and Chuang 2000). Each superposition, $|\psi\rangle$, can be represented as a linear sum of the basis states, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are numbers that denote the corresponding states' probability amplitudes. The values $|\alpha|^2$ and $|\beta|^2$ are the probabilities that the observation of a qubit in state $|\psi\rangle$ will render a '0' or '1' state, respectively (Glassner 2001b), and normalization requires that $|\alpha|^2 + |\beta|^2 = 1$. Various quantum gates such as the NOT gate, AND gate, OR gate, NAND gate, Hadamard gate and rotation gates can be applied to modify the state of a qubit (Hey 1999). A quantum system $|\psi_n\rangle$ with n qubits can represent 2^n states simultaneously (Grover 1999; Bennett and DiVincenzo 2000) as

$$|\psi_n\rangle = \sum_{j=1}^{2^n} C_j |S_j\rangle, \tag{1}$$

Fig. 1 Pseudocode algorithm for evolutionary algorithms (Bäck et al. 1997)

```
Begin
(i)     Initialize Q(t), t =0 ;
(ii)    Evaluate Q(t) ;
        While (not termination condition) do
(iii)     P(t) ← Vary Q(t) ;
(iv)      Evaluate P(t) ;
(v)       Q(t +1) ← Select P(t) ;
          t ← t + 1 ;
        End
End
```

where C_j is the probability amplitude of the jth state S_j described by the binary string $(x_1x_2\cdots x_n)$, where x_i, $i = 1, 2, \ldots, n$, is either 0 or 1. However, the system will "collapse" to a single state if a quantum state is observed. Since the early 1990s, some significant quantum algorithms including the quantum search algorithm (Grover 1997) and quantum factorization algorithm (Shor 1994), have been proposed to show that quantum computers are in some sense more powerful than classical computers at least with respect to solving some specific problems (Narayanan 1999; Kent and Williams 1999).

Inspired by natural selection (Darwin 1859) and molecular genetics (Burian 1996), evolutionary algorithms (EAs) define practical and robust optimization and search methodologies. As compared with conventional optimization methods, EAs provide a general approach to solving complex problems. Their global search capabilities, their flexibility, robust performance and adaptability are all considered as outstanding characteristics of EAs when searching for optimal solutions (Bäck et al. 1997). Although the primary ideas of evolutionary computation came from the influential works of Fraser (1957), Box (1957), Friedberg (1958), Friedberg et al. (1959) and Bremermann (1962) in the late 1950s, it was not until the 1970s that evolutionary computation started to be taken seriously. Evolutionary computation includes three main branches: genetic algorithms (GAs) introduced by Holland (1975); evolutionary programming, introduced by Fogel et al. (1966); and evolution strategies, introduced by Rechenberg (1973) and Schwefel (1975). The representation of chromosomes, the mutation and/or recombination operators, and selection/reproduction methods are the key features that differentiate these three approaches.

The pseudocode algorithm for a canonical EA (Bäck et al. 1997) is shown in Fig. 1, in which $Q(t)$ represents a population with n individuals at generation t. The individuals in $P(t)$ are evaluated by using an auxiliary *objective function*, and after applying the variation operators an offspring population $Q(t+1)$ evolves from the population $Q(t)$. The selection operation is applied to select better individuals in terms of encoded fitness values to make the population evolve forward.

The possible interplay between quantum computing and evolutionary computation has been explored since the late 1990s. Three kinds of algorithms have been identified in this context and are respectively called in this paper:

- Evolutionary-designed quantum algorithms (EDQAs): the automated synthesis of new quantum algorithms using evolutionary algorithms such as genetic programming and GAs has been explored in Spector et al. (1999), Koza et al. (2005), Grigorenko and Garcia (2001, 2002), Sahin and Tomak (2005).

– Quantum evolutionary algorithms (QEAs): QEAs focus on trying to implement evolutionary algorithms in a quantum computation environment (Spector et al. 1998; Rylander et al. 2000; Malossini et al. 2004; Sofge 2006; Sahin et al. 2005; Udrescu et al. 2006) in order to take advantage of quantum computation's exponential parallelism.
– Quantum-inspired evolutionary algorithms (QIEAs): QIEAs concentrate on generating new evolutionary algorithms using some concepts and principles of quantum computing such as standing waves (Moore and Narayanan 1995), interference (Zhou and Sun 2005), coherence (Pötz and Fabian 2006), etc.

This paper will focus on the QIEA. In recent years, research into QIEA, a new and promising branch of evolutionary algorithms, has become a rapidly expanding field. To date, however, there is no survey of the various types of QIEA, and furthermore, some confusing QIEA concepts that have appeared in the literature need to be clarified. Finally, some potential research developments aimed at advancing the theory of QIEAs and their applications will be discussed.

Quantum-inspired computation uses computational methods based on the concepts and principles of quantum mechanics, such as qubits, superposition, quantum gates and quantum measurement, in order to solve various problems in the context of a classical computing paradigm (Moore and Narayanan 1995). Like a quantum mechanical system, a quantum-inspired system can be regarded as a probabilistic system, in which the probabilities related to each state are utilized to describe the behavior of the system. QIEAs use quantum-inspired bits (Q-bits), quantum-inspired gates (Q-gates) and observation processes to specify their structure and steps. More specifically, Q-bits are applied to represent genotype individuals; Q-gates are employed to operate on Q-bits to generate offspring; and the genotypes and phenotypes are linked by a probabilistic observation process. In quantum mechanical systems, the act of observation causes a quantum particle to take on one and only one state in the measurement basis (i.e., one of the states $|0\rangle$ and $|1\rangle$) (Glassner 2001a). Similarly, a superposition state represented by Q-bits in QIEAs will become a single state in the process of observation.

QIEAs were firstly introduced by Narayanan and Moore in the 1990s to solve the traveling salesman problem (Narayanan and Moore 1996), in which the crossover operation was performed based on the concept of interference. The contribution of Narayanan and Moore signaled the potential advantage of introducing quantum computational parallelism into the evolutionary algorithm framework. No further attention was paid to QIEAs until a practical algorithm was proposed by Han and Kim (2000, 2002), but they are now viewed as an emergent theme in evolutionary computation. Albeit various variants of QIEA have been presented in the literature, they can be categorized into three types: binary observation QIEA (bQIEA) (Han and Kim 2000, 2002, 2004), real observation QIEA (rQIEA) (Zhang and Rong 2007c; Liu et al. 2008) and QIEA-like algorithms (iQIEA) (Abs da Cruz et al. 2004, 2006; Sailesh Babu et al. 2008). Inspired by the concepts of quantum computing, such as qubits and quantum gates, a QIEA has the following main characteristics.

– A QIEA adopts a new representation, *Q-bit representation*, to describe individuals of a population. Q-bit representation provides probabilistically a linear superposition of multiple states.

Items		EDAs	QIEAs
Differences	Motivation	Combining EAs with machine learning to investigate how to use the knowledge of current individuals to generate the offspring	Inspired by the concepts and principles in quantum computing, how to apply the data representation in quantum computing to specify the structure and algorithms of EAs
	Algorithm	Fig.3	Fig.3
	Representation	Binary representation	Q-bit representation
	Population	Need a larger number of individuals to avoid significant bias of joint probability	The population size can be smaller, even one individual
	Application	Mainly for combinatorial optimization problems	Both for combinatorial and numeric optimization problems
	Further development	Efficient estimation techniques for joint probability and numeric optimization	Exploring the effective application of other concepts (quantum registers, entanglement, interference, etc.) to EAs, to solve more complex problems including dependent variables

Fig. 2 Comparisons of QIEAs and EDAs

- A QIEA uses a *Q-gate*, which can guide the individuals toward better solutions (Han and Kim 2002), to generate the individuals at the next generation.
- A QIEA can exploit the search space for a global solution with a small number of individuals, even with one element (Han and Kim 2002).

The present rapidly increasing interest in QIEAs is reflected in the growth of the research community, but no survey on QIEAs has yet appeared in the specialized literature. So the main aim of this paper is to provide basic (mainly descriptive) information so as to allow newcomers to the area to get a clear understanding of key research problems and developments in the field, including those that are currently under way. The significant contribution of this work is to give a comprehensive survey of QIEAs and their applications and to point out some issues deserving special attention in this area. At the same time, this work provides researchers with a clear understanding of such concepts as EDQAs, QEAs, and other notions related to QIEAs. Furthermore, this paper is intended to advance QIEA-related theoretical research and to deepen the application of quantum computing techniques and concepts to evolutionary computation.

Since QIEAs can be regarded as a kind of *estimation of distribution algorithm* (EDA), a succinct comparison between QIEAs and EDAs (Santana et al. 2008; Pelikan et al. 2000, 2002; Baluja 1994; Baluja and Davies 1997; De Bonet et al. 1997; Harik 1999; Harik et al. 1998; Larrañaga et al. 2000; Mühlenbein and Mahnig 1998, 1999; Pelikan and Mühlenbein 1999; Platel et al. 2009) is presented in Fig. 2.

The relationship between EDAs and QIEAs is worth considering in more detail. As Figs. 2 and 3 show, EDAs need auxiliary selection methods and replacement strategies to be provided, which are unnecessary for QIEAs. While probability is a fundamental property of all EAs, and EDAs may use it in the course of selection and replacement, QIEAs employ it in the process of observation. However, if we restrict attention to QIEAs that utilize only one individual and multiple observations at each generation, then the two models are similar (Zhou and Sun 2005).

The rest of this paper is arranged as follows. Section 2 introduces QIEAs and their applications, and provides an overview of the QIEA work done so far. Section 3 presents a small comparative study. Experiments are carried out on benchmark

Algorithm for the EDA	*Algorithm for the QIEA*
(i) Generate N individuals randomly; **While** (not termination condition) **do** (ii) Select M ($M \leq N$) individuals using a selection approach to calculate their joint probability; (iii) Sample M individuals from the joint probability; (iv) Produce offspring using a replacement strategy; **End**	(i) Generate a population represented by Q-bits; **While** (not termination condition) **do** (ii) Probabilistic observation; (iii) Evaluate all individuals and select the best one; (iv) Produce offspring using Q-gates; **End**

Fig. 3 Comparisons of main schemes of EDAs and QIEAs

problems to evaluate the performances of different types of QIEA. Finally, some conclusions and some possible further developments are presented in Sect. 4.

2 Quantum-inspired evolutionary algorithms

Like conventional EAs, QIEAs are characterized by the representation of individuals, population diversity, and the use of a fitness evaluation mechanism (Han and Kim 2002). However, unlike the conventional framework for EAs, QIEAs describe individuals through Q-bit representation, and use of a Q-gate as an evolutionary operator to obtain fitter individuals by employing an observation process to connect the Q-bit representation with the optimization variables. This section summarizes the work done on QIEAs. First, Q-bit representation is introduced in Sect. 2.1. Then the bQIEA and several of its variants are discussed in Sect. 2.2, the rQIEA and its potential applications are presented in Sect. 2.3, and several QIEA-like algorithms are discussed in Sect. 2.4. Finally, a summary follows in Sect. 2.5.

2.1 Q-bit representation

In conventional EAs, encoding the solutions onto chromosomes uses many different representations, which may be generally grouped into three classes: symbolic, binary, and numeric (Hinterding 1999). In contrast, a QIEA uses the *Q-bit representation*, a novel probabilistic description of Q-bit individuals as strings of Q-bits. The Q-bit is the basic computing unit in a QIEA and is defined as a column vector

$$[\alpha \;\; \beta]^T, \tag{2}$$

where the numbers α and β satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. We often write Eq. 2 in quantum mechanical ket-notation, $\alpha|0\rangle + \beta|1\rangle$, and as in quantum theory, the values $|\alpha|^2$ and $|\beta|^2$ denote the probabilities that the Q-bit will be found in the '0' or '1' state, respectively (Han and Kim 2002). By a process of probabilistic observation, each Q-bit can be rendered into one binary bit. This observation process is shown in Fig. 4, in which x is the observed value of the Q-bit shown in Eq. 2. Whereas binary representation uses 0 or 1 to deterministically represent a bit, the Q-bit representation employs a Q-bit to describe a probabilistic linear superposition of 0 and 1, and this representation extends naturally to multi-Q-bit systems. For

Fig. 4 Observation process in the QIEA (Han and Kim 2002)

Begin

If $random[0,1) < |\alpha|^2$

Then $x \leftarrow 0$

Else $x \leftarrow 1$

End

example, consider a three-Q-bit system with three pairs of amplitudes, such as

$$\begin{bmatrix} \alpha_1 | \alpha_2 | \alpha_3 \\ \beta_1 | \beta_2 | \beta_3 \end{bmatrix} = \begin{bmatrix} \frac{-\sqrt{3}}{3} | \frac{\sqrt{2}}{3} | \frac{-\sqrt{5}}{3} \\ \frac{\sqrt{6}}{3} | \frac{\sqrt{7}}{3} | \frac{-2}{3} \end{bmatrix}, \tag{3}$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1,\ i = 1, 2, 3$. This represents a linear probabilistic superposition of $2^3 = 8$ states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$. In the process of observation, each of these eight states can be selected, and the associated probabilities are $|\alpha_1\alpha_2\alpha_3|^2 = 30/729$, $|\alpha_1\alpha_2\beta_3|^2 = 24/729$, $|\alpha_1\beta_2\alpha_3|^2 = 105/729$, $|\alpha_1\beta_2\beta_3|^2 = 84/729$, $|\beta_1\alpha_2\alpha_3|^2 = 60/729$, $|\beta_1\alpha_2\beta_3|^2 = 48/729$, $|\beta_1\beta_2\alpha_3|^2 = 210/729$ and $|\beta_1\beta_2\beta_3|^2 = 168/729$, respectively. Taking the signs of the various α_i, β_i, into account, the states of the system can be represented as

$$\begin{aligned} |\psi_3\rangle = {} & \frac{\sqrt{30}}{27}|000\rangle + \frac{\sqrt{24}}{27}|001\rangle - \frac{\sqrt{105}}{27}|010\rangle + \frac{\sqrt{84}}{27}|011\rangle \\ & + \frac{\sqrt{60}}{27}|100\rangle - \frac{\sqrt{48}}{27}|101\rangle + \frac{\sqrt{210}}{27}|110\rangle - \frac{\sqrt{168}}{27}|111\rangle. \end{aligned} \tag{4}$$

As this example illustrates, the Q-bit representation can represent a linear superposition of states probabilistically; thus a QIEA's population, i.e., the encoded genotypes, potentially map to a larger phenotype space than other EAs with binary representation.

2.2 bQIEA

The *binary observation QIEA* (bQIEA) was initially proposed in 2000 by Han and Kim (2000, 2002) to solve combinatorial optimization problems; we refer to their model as the *original bQIEA* (bQIEAo). Since then, various variants of the bQIEA have been developed and they can be classified into three categories: bQIEA with crossover and mutation operators (bQIEAcm), bQIEA with a novel update method for Q-gates (bQIEAn), and hybrid bQIEA (bQIEAh). In what follows these bQIEA variants will be introduced step by step.

2.2.1 bQIEAo

The structure of bQIEAo was first expounded in Han and Kim (2002), although preliminary work was initiated in Han and Kim (2000). As the kernel of several variants of bQIEA, bQIEAo will be explained in detail, since a clear understanding of bQIEA

```
Begin
    t ← 0
(i)     Initialize Q(t)
(ii)    Make P(t) by observing the states of Q(t)
(iii)   Evaluate P(t)
(iv)    Store the best solutions among P(t) into B(t) and the best solution b among B(t)
        While (not termination condition) do
            t ← t+1
(v)         Make P(t) by observing the states of Q(t -1)
(vi)        Evaluate P(t)
(vii)       Update Q(t) using Q-gates
(viii)      Store the best solutions among P(t) and B(t -1) into B(t) and the best solution b among B(t)
            If (migration condition)
(ix)            Migrate b or b_j^t to B(t) globally or locally, respectively
            End
        End
End
```

Fig. 5 Pseudocode algorithm for bQIEAo (Han and Kim 2002)

is required if we are also to understand the other bQIEA variants. The pseudocode algorithm for bQIEAo is illustrated in Fig. 5.

Each step of this algorithm is described below.

(i). In the "initialize $Q(t)$" step, a population $Q(0)$ with n multi-Q-bit individuals is generated, $Q(t) = \{\boldsymbol{q}_1^t, \boldsymbol{q}_2^t, \ldots, \boldsymbol{q}_n^t\}$, at the generation moment $t = 0$, where $\boldsymbol{q}_i^t$ $(i = 1, 2, \ldots, n)$ is an arbitrary individual in $Q(t)$, represented as

$$\boldsymbol{q}_i^t = \begin{bmatrix} \alpha_{i1}^t | \alpha_{i2}^t | \cdots | \alpha_{im}^t \\ \beta_{i1}^t | \beta_{i2}^t | \cdots | \beta_{im}^t \end{bmatrix}, \tag{5}$$

where m is the number of Q-bits used in each individual's representation, i.e., the string length of the Q-bit individual. The value α_{ij}^t and β_{ij}^t, $j = 1, 2, \ldots, m, t = 0$, are initialized to the same probability amplitude $1/\sqrt{2}$, so that all possible states are superposed with the same probability at the beginning.

(ii). By independently observing each Q-bit of $Q(t)$ (where at this stage $t = 0$), using the process described in Fig. 4, binary solutions in $P(t)$, $P(t) = \{\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_n^t\}$, are obtained, where each $\boldsymbol{x}_i^t$ $(i = 1, 2, \ldots, n)$ is a binary solution with m bits. Each bit '0' or '1' is the observed value of a Q-bit $[\alpha_{ij}^t \;\; \beta_{ij}^t]^T$ in $\boldsymbol{q}_i^t$, respectively, $j = 1, 2, \ldots, m$.

(iii). The binary solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \ldots, n)$ in $P(t)$ is evaluated thus obtaining its fitness.

(iv). In this step, all solutions in $P(t)$ are stored into $B(t)$, where $B(t) = \{\boldsymbol{b}_1^t, \boldsymbol{b}_2^t, \ldots, \boldsymbol{b}_n^t\}$ and $\boldsymbol{b}_i^t = \boldsymbol{x}_i^t$ $(i = 1, 2, \ldots, n)$ (again, at this stage, $t = 0$). Furthermore, the best binary solution $\boldsymbol{b}$ in $B(t)$ is also stored.

(v). This step is similar to step (ii). Observation of the states of $Q(t-1)$ produces the binary solutions in $P(t)$.

(vi). This step is similar to step (iii).

(vii). In this step, all the individuals in $Q(t)$ are modified by applying Q-gates. The bQIEAo use a *quantum rotation gate* as a Q-gate. To be specific, the jth Q-bit in the ith Q-bit individual $\boldsymbol{q}_i^t$, $j=1,2,\ldots,m$, $i=1,2,\ldots,n$, is updated by applying the current Q-gate $G_{ij}^t(\theta)$

$$G_{ij}^t(\theta)=\begin{bmatrix}\cos\theta_{ij}^t & -\sin\theta_{ij}^t \\ \sin\theta_{ij}^t & \cos\theta_{ij}^t\end{bmatrix}, \tag{6}$$

where θ_{ij}^t is an adjustable Q-gate rotation angle. Thus, the update procedure for the Q-bit $[\alpha_{ij}^t \;\; \beta_{ij}^t]^T$ can be described as

$$\begin{bmatrix}\alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1}\end{bmatrix}=G_{ij}^t(\theta)\begin{bmatrix}\alpha_{ij}^t \\ \beta_{ij}^t\end{bmatrix}, \tag{7}$$

where θ_{ij}^t is defined as

$$\theta_{ij}^t=s(\alpha_{ij}^t,\beta_{ij}^t)\Delta\theta_{ij}^t, \tag{8}$$

and $s(\alpha_{ij}^t,\beta_{ij}^t)$ and $\Delta\theta_{ij}^t$ are the sign and the value of θ_{ij}^t, respectively. The particular values used in bQIEAo are illustrated in Table 1, in which $f(\cdot)$ is the fitness function, $s(\alpha_{ij}^t,\beta_{ij}^t)$ depends on the sign of $\alpha_{ij}^t\beta_{ij}^t$, and b and x are certain bits of the searched best solution $\boldsymbol{b}$ and the current solution $\boldsymbol{x}$, respectively (Han and Kim 2002). It is worth pointing out that Table 1 was derived from a maximum problem and hence the condition $f(\boldsymbol{x})\geq f(\boldsymbol{b})$ should be replaced by $f(\boldsymbol{x})\leq f(\boldsymbol{b})$ if a minimum problem is to be considered.

(viii). This step is similar to step (iv). The better candidate between $\boldsymbol{x}_i^t$ in $P(t)$ and $\boldsymbol{b}_i^{t-1}$ in $B(t-1)$, $i=1,2,\ldots,n$, is selected and stored into $B(t)$. Simultaneously, the best candidate $\boldsymbol{b}$ in $B(t)$ is also stored.

(ix). This step includes local and global migrations, where a *migration* in this algorithm is defined as the process of copying $\boldsymbol{b}_j^t$ in $B(t)$ or $\boldsymbol{b}$ to $B(t)$. A global migration is realized by substituting $\boldsymbol{b}$ for all the solutions in $B(t)$, and a local migration is realized between each pair of neighboring solutions in $B(t)$, i.e., by substituting the better one of two neighboring solutions for the other solution. For more information about the migrations, see Han and Kim (2002).

The knapsack problem, a well-known NP-hard combinatorial optimization problem, was chosen by Han and Kim as a suitable application example to investigate the setting of parameters for, and the performance of, bQIEAo. Empirical guidelines for setting the Q-gate parameters were drawn up following extensive experiments, and to show the advantages of bQIEAo over conventional GAs (CGAs) with various crossover and mutation probabilities, a large number of experiments were conducted on the knapsack problems with different number of items. In Han and Kim (2002), the bQIEAo's convergence properties were also analyzed by observing the changing

Table 1 Lookup table of θ_{ij}^t, where $f(\cdot)$ is the fitness, $s(\alpha_{ij}^t, \beta_{ij}^t)$ is the sign of θ_{ij}^t, and b and x are certain bits of the searched best solution $\boldsymbol{b}$ and the current solution $\boldsymbol{x}$, respectively (Han and Kim 2002)

x	b	$f(\boldsymbol{x}) \geq f(\boldsymbol{b})$	$\Delta\theta_{ij}^t$	$s(\alpha_{ij}^t, \beta_{ij}^t)$	
				$\alpha_{ij}^t\beta_{ij}^t \geq 0$	$\alpha_{ij}^t\beta_{ij}^t < 0$
0	0	false	0	± 1	± 1
0	0	true	0	± 1	± 1
0	1	false	0.01π	$+1$	-1
0	1	true	0	± 1	± 1
1	0	false	0.01π	-1	$+1$
1	0	true	0	± 1	± 1
1	1	false	0	± 1	± 1
1	1	true	0	± 1	± 1

trends of probabilities of all solutions using a single Q-bit individual in the process of finding the optimal profit for a knapsack problem with ten items.

Han and Kim further studied the termination criterion, a modified Q-gate H_ϵ and the initial values' setting of Q-bits in Han and Kim (2004). Whereas EAs generally use the maximal number of generations as a termination condition, bQIEAo could employ a Q-bit convergence termination criterion due to its probability-based representation of the individuals. In terms of our notation in Eq. 5, the Q-bit convergence C_i in Han and Kim (2004) was defined as

$$C_i = \frac{1}{m}\sum_{j=1}^{m} ||\alpha_{ij}^t|^2 - |\beta_{ij}^t|^2| \tag{9}$$

and the termination criterion was of the form, $C_i \geq \lambda$, where λ is some appropriately selected number such as 0.9. This termination criterion gives a clear meaning to how much closely Q-bit individuals converge to 0 or 1. The introduction of the H_ϵ gate is to prevent the premature convergence of bQIEAo by keeping a Q-bit away from 0 or 1 to a certain degree. Moreover, how to merge prior knowledge into the initial values of Q-bits was also discussed to improve bQIEAo performance. The bQIEAo algorithm was applied to solve 6 numerical optimization problems in Han and Kim (2004), and experimental results compared with Yao et al. (1999) show that bQIEAo is competitive with classical evolutionary programming and fast evolutionary programming.

In Han and Kim (2006), a simplified model of the segment process bQIEAo was considered to analyze the convergence for exploitation, and Shannon entropy was used to investigate the exploration strategy. Theoretical analysis indicates that bQIEAo with a single Q-bit individual for ONE-MAX problem guarantees the global solution within the expected number of generations. The exploration mechanisms applied clearly demonstrate that bQIEAo starts with a global search and then automatically turns into a local search as the number of generations increases, due to its inherent probabilistic nature, which achieves to a good balance between exploration and exploitation.

Further studies considered parallelization (Han et al. 2001; Kim et al. 2006; Yang et al. 2003a, 2003b), the extension to multi-objective algorithm (Zhou and Sun 2005), and other advanced features of the bQIEAo (Li et al. 2004b, 2009; Zhang and Gao 2007a; Khorsand 2005; Chen et al. 2004; Han and Kim 2003a, 2003b; Platelt et al. 2007; Imabeppu et al. 2008). A summary of this research is shown in Fig. 12.

In the study of bQIEAo, attention was given not only to research issues, but also to applications. In Jang et al. (2004a), bQIEAo was applied to improve principal component analysis methods by optimizing the weight factors of distance measures, and consequently a bQIEAo-based classifier was presented to enhance face verification performances. Experiments carried out on the face and non-face images extracted from Aleix and Robert's face database (Martinez and Benavente 1998) show that the proposed classifier performs better than the distance-from-face space classifier and the maximum likelihood classifier both in terms of the face verification rate and the false alarm rate. In Kim et al. (2003), a bQIEAo-based disk allocation method was proposed for distributing buckets of a binary Cartesian product file among an unrestricted number of disks to maximize concurrent disk I/O. The experimental results show that the introduced method achieves equal or shorter average query response times and 3.2–11.3 times faster convergence speed than those of disk allocation methods using CGA. Additionally, bQIEAo was also applied to solve various problems, such as parameter selection for support vector machines (Luo et al. 2008), clustering gene expression data (Zhou et al. 2006b), neural network training (Ganesh and Singhal 2005) and so on (Lu et al. 2008; Liu et al. 2005, 2006; Huo and Stojkovic 2006, 2007; Akbarzadeh-T 2005; Xiao et al. 2006; Khorsand 2006; Feng et al. 2006; Vlachoglannis 2008; Zhao et al. 2006; Jang et al. 2003, 2004b, 2009; Lv and Liu 2007; John and John 2009; Gu et al. 2009b; Zhou et al. 2005, 2006a; Lau et al. 2009; Araujo et al. 2008; Jeong et al. 2009; Xing et al. 2009a, 2009b; Gu et al. 2009a). Various problems solved by bQIEAo are summarized in Tables 3 and 4. The above applications show that bQIEAo is a practical and efficient optimization algorithm.

Remarks As compared with binary, numeric and symbolic representations, the Q-bit representation can achieve a linear superposition of states given its probabilistic approach and is conductive to population diversity. Using a Q-gate as a variation operator, instead of crossover, recombination and mutation operators, bQIEAo can find the optimal or close-to-optimal solutions with a small number of individuals, even with a single individual, as verified in Han and Kim (2002), Zhou and Sun (2005). Furthermore, bQIEAo uses the current best solution to control different searching directions and only a small amount of information needs to be exchanged between multiple subpopulations; as a result, bQIEAo is suitable for parallel implementation and has the potential to greatly reduce the communication and synchronization costs. More importantly, the performance analysis and extensively convincing experiments in Han and Kim (2002, 2004, 2006) show further the soundness of bQIEAo, although the details would take us beyond the scope of this survey. As can shown by the experiments and analysis in the literature, bQIEAo can achieve good experimental results, and can also balance well between exploration and exploitation.

On the other hand, some issues concerning bQIEAo require further study. First, it is worth asking how best to present the parameters of a Q-gate. The Q-gate in bQIEAo has 8 parameters to be preset before its update process. This issue was investigated in

Han and Kim (2002, 2003a, 2003b, 2004), Zhang and Gao (2007a), Khorsand (2005) and an effective heuristic approach was derived empirically by considering the parameters as fixed values throughout the whole process of evolution. But ways of reducing the number of Q-gate parameters, and dynamically adjusting these parameters, is worth further discussion. For instance, the parameters could be set to relatively bigger values at the beginning of the evolution process so that the algorithm explores the whole solution space, which then decrease gradually to relatively small values so as to exploit the neighboring areas of the searched solutions. Also, as pointed out in Han and Kim (2002, 2004, 2006), the universality and effectiveness of the heuristic strategy in Han and Kim (2002) for Q-gate parameters need to be further verified in other problems and applications. Second, the searching direction is dominated by the current best solution, and consequently bQIEAo may get stuck in local minima when the probabilities of the current best solution become 0 or 1. In Han and Kim (2004), a H_ϵ gate was introduced to solve the problem to a certain degree, but an additional parameter ϵ was also brought into bQIEAo's definition. Finally, more comparisons are necessary not only between bQIEAo and the latest optimization methods such as particle swarm optimization and estimation of distribution algorithms, but also in solving other well-known optimization problems.

2.2.2 bQIEAcm

The bQIEAcm is a modified version of bQIEAo. It uses crossover and mutation operators to replace the bQIEAo migration operators. According to the bQIEAcm reported in Li et al. (2005a), Xu et al. (2005), Meshoul et al. (2005a, 2005b), Wang et al. (2005c), Yang et al. (2004a, 2004b, 2005); Talbi et al. (2004a, 2004b, 2004c), Li and Zhuang (2002), Abdesslem et al. (2006), Yang and Jiao (2003), Guo et al. (2007), Yang and Ding (2007), Shu (2007), Wei et al. (2008), Ding et al. (2008), Zhao et al. (2009), the pseudocode algorithm can be summarized as shown in Fig. 6. In bQIEAcm, the crossover and mutation operators are performed on Q-bit individuals, so they are called *quantum crossover and quantum mutation*, respectively, so as to differentiate them from those in CGA. In Fig. 6, the first three steps and steps (v)–(vii) are the same as those in Fig. 5. In steps (iv) and (viii), storing the best solution among $P(t)$ is sufficient. Quantum crossover in step (ix) and quantum mutation in step (x) are explained in Figs. 7 and 8, respectively, in which $\boldsymbol{q}_i$, $\boldsymbol{q}_j$ $(i, j = 1, 2, \ldots, n)$ are any two individuals of the population $Q(t)$ and $\boldsymbol{q}'_i$, $\boldsymbol{q}'_j$ are the resulting individuals. It is worth noting that Fig. 6 only shows one-point crossover and Fig. 8 shows uniform mutation, but obviously, other crossover and mutation operators in CGA may also be introduced into bQIEAcm.

In Abdesslem et al. (2006), Meshoul et al. (2005a), the bQIEAcm was successfully applied to solve a multiple sequence alignment problem, which is a well-known NP-hard combinatorial optimization problem in bioinformatics (Wang and Jiang 1994). Experiments were conducted on two benchmarks with 24 data sets (Thompson et al. 1999; Gardner et al. 2005). The results show that bQIEAcm is much better than several leading alignment techniques (Eddy 2009; Notredame et al. 1998) including CLUSTAL, DIALIGN, MATFFT, PROALIGN and COFFEE. In Meshoul et al. (2005b), the applicability of bQIEAcm to multi-objective knapsack problems was

Fig. 6 Pseudocode algorithm for bQIEAcm

```
              Begin
                t ← 0
(i) - (iii)     The first three steps of bQIEAo
(iv)            Store the best solution b among P(t)
                While (not termination condition) do
                  t ← t+1
(v) - (vii)       Same as bQIEAo
(viii)            Store the best solution b among P(t)
(ix)              Quantum crossover
(x)               Quantum mutation
                End
              End
```

$$\begin{cases} q_i & \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \ldots & \alpha_{ih} & \ldots & \alpha_{im} \\ \beta_{i1} & \beta_{i2} & \ldots & \beta_{ih} & \ldots & \beta_{im} \end{bmatrix} \\ q_j & \begin{bmatrix} \alpha_{j1} & \alpha_{j2} & \ldots & \alpha_{jh} & \ldots & \alpha_{jm} \\ \beta_{j1} & \beta_{j2} & \ldots & \beta_{jh} & \ldots & \beta_{jm} \end{bmatrix} \end{cases} \Rightarrow \begin{cases} q_i' & \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \ldots & \alpha_{jh} & \ldots & \alpha_{im} \\ \beta_{i1} & \beta_{i2} & \ldots & \beta_{jh} & \ldots & \beta_{im} \end{bmatrix} \\ q_j' & \begin{bmatrix} \alpha_{j1} & \alpha_{j2} & \ldots & \alpha_{ih} & \ldots & \alpha_{jm} \\ \beta_{j1} & \beta_{j2} & \ldots & \beta_{ih} & \ldots & \beta_{jm} \end{bmatrix} \end{cases}$$

Fig. 7 Quantum crossover operation (the h Q-bits have been swapped)

$$\left\{ q_i \quad \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \ldots & \alpha_{ih} & \ldots & \alpha_{im} \\ \beta_{i1} & \beta_{i2} & \ldots & \beta_{ih} & \ldots & \beta_{im} \end{bmatrix} \right. \Rightarrow \left\{ q_i' \quad \begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \ldots & \beta_{ih} & \ldots & \alpha_{im} \\ \beta_{i1} & \beta_{i2} & \ldots & \alpha_{ih} & \ldots & \beta_{im} \end{bmatrix} \right.$$

Fig. 8 Quantum mutation operation (the h Q-bits has been reversed)

discussed and experiments carried out on two benchmark data sets (Zitzler and Laumanns 1999; Ruiz 2009) show that a significant improvement over a state-of-the-art algorithm SPEA2 (Zitzler et al. 2001). Additionally, more applications of bQIEAcm and results regarding the performances of this method (Li et al. 2005a; Xu et al. 2005; Li and Zhuang 2002; Wang et al. 2005c; Yang et al. 2004a, 2004b, 2005; Yang and Jiao 2003; Talbi et al. 2004a, 2004b, 2004c; Guo et al. 2007; Yang and Ding 2007; Shu 2007) are listed in Table 4.

Remarks The crossover operator, a method for sharing information between chromosomes, and the mutation operator, a way of increasing the structural variability of a population, play a central role in improving CGA behavior because the former may produce additional diversity (divergence) or the refinement of the solutions (convergence) and the latter may restore lost or unexplored genetic materials to the population (Auger and Hansen 2005; Herrera et al. 1998). Quantum crossover and quantum mutation can be regarded as extensions of the crossover and mutation operators in CGA. It can be seen from Li et al. (2005a), Xu et al. (2005), Meshoul et al. (2005a), Yang and Ding (2007), Abdesslem et al. (2006), Meshoul et al. (2005b), Wang et al. (2005c), Yang et al. (2004a, 2004b, 2005), Talbi et al. (2004a, 2004b, 2004c), Li and Zhuang (2002), Shu (2007), Yang and Jiao (2003), Guo et al. (2007) that good results have been obtained in several applications. Like the crossover and mutation operators in CGA, quantum crossover and mutation operators are helpful to prevent

Fig. 9 Pseudocode algorithm for bQIEAn (Zhang et al. 2006)

Begin
$t \leftarrow 0$
(i) - (vi) The first six steps of bQIEAo
(vii) Update $Q(t)$ using Q-gates
(viii) Store the best solutions among $P(t)$ and $B(t\text{-}1)$ into $B(t)$
(ix) Migration
(x) Catastrophe
End

bQIEAcm from mature convergence to suboptimal solutions because they are beneficial to population diversity, especially in the latter stage of evolution. But to what degree they function and what is the contribution of each operator to the success of bQIEAcm is still worth investigating further because they are applied to Q-bit individuals, rather than the standard individuals for which results are normally quoted. In addition, more convincing experiments need to be conducted to compare bQIEAcm with bQIEAo and other good optimization algorithms.

2.2.3 bQIEAn

Zhang et al. (2006) presented a modified bQIEA called bQIEAn, in which a novel update method for Q-gates and a catastrophe operator were used. The pseudocode algorithm for bQIEAn is shown in Fig. 9. The eight steps (i)–(vi), (viii) and (ix) are the same as those in bQIEAo. In the step (vii), the Q-gate angle θ was defined as

$$\theta = k \cdot f(\alpha, \beta), \tag{10}$$

where $f(\alpha, \beta)$ and k are the sign and the value of θ, respectively. The k value has a direct effect on the convergence speed. In bQIEAn, k was defined as a variable related to evolutionary generations so as to dynamically adjust the search grid:

$$k = 0.5\pi \cdot e^{-5t/t_{max}}, \tag{11}$$

where t is the current evolutionary generation, and t_{max} is the maximum number of generations. $f(\alpha, \beta)$ is a searching direction function for guiding bQIEAn toward better solutions. The value of $f(\alpha, \beta)$ can be obtained from Table 2, in which $d_1 = \alpha_1\beta_1$ and $\xi_1 = \arctan(\beta_1/\alpha_1)$, where α_1, β_1 are the probabilities of the searched best solution, and $d_2 = \alpha_2\beta_2$, $\xi_2 = \arctan(\beta_2/\alpha_2)$, where α_2, β_2 are the probabilities of the current solution.

In the "catastrophe" step, if the best solution is maintained unchanged over a certain number of generations, the population catastrophe operation will be executed, causing the best individual in $b(t)$ to be replaced by the best individual of a new population.

In Zhang et al. (2004a, 2004b, 2006), bQIEAn was employed to select the most discriminatory feature subsets from a large number of features of radar emitter signals. The work shows that bQIEAn based feature selection algorithm can search for a good feature subset to identify different types of signals. Using a Markov chain method, the convergence of bQIEAn was analyzed mathematically in Zhang et al.

Table 2 Look-up table of function $f(\alpha, \beta)$ in bQIEAn

$d_1 \geq 0$	$d_2 > 0$	$f(\alpha, \beta)$	
		$\lvert\xi_1\rvert \geq \lvert\xi_2\rvert$	$\lvert\xi_1\rvert < \lvert\xi_2\rvert$
true	true	$+1$	-1
true	false	-1	$+1$
false	true	-1	$+1$
false	false	$+1$	-1

(2003b). The applicability of bQIEAn to physical distribution vehicle routing, FIR and IIR digital filter design and time-frequency atom decomposition was also discussed in Gao et al. (2006), Wang et al. (2007b), Zhang et al. (2003a, 2003c), Zhang and Rong (2006, 2007b). Table 4 summarizes the work done on bQIEAn.

Remarks The bQIEAn can be regarded as a modified type of bQIEAo. As compared with bQIEAo, bQIEAn has far fewer parameters in a Q-gate to preset, as there is only one parameter defined as a variable that changes dynamically with the evolutionary generation. Furthermore, the catastrophe operator helps the bQIEAn to avoid evolutionary stagnation and local minima by changing the search direction due to the replaced best individual on Q-gates. However, some aspects of the bQIEAn need further study. First, we need a systematic analysis of the bQIEAn so as to understand clearly its update method for Q-gates and the significance of the catastrophe operator. Extensive comparative experiments between dynamic adjustments and prescribed values should be carried out to find the best approach for adjusting the parameter used by the Q-gate operator. Next, more convincing experiments need to be conducted to compare bQIEAn with bQIEAo, as well as with other good optimization algorithms. Finally, the update method introduced in bQIEAn is a potentially promising scheme for numerical optimization problems, and this needs to be investigated further.

2.2.4 bQIEAh

To improve bQIEA performance, other optimization techniques have been introduced (Li et al. 2004a, 2004c, 2005b, 2006; Wang et al. 2005a, 2005b, 2005d, 2007a, 2007c; You et al. 2006a, 2006b, 2006c, 2007; Li and Jiao 2005, 2007, 2008; Li and Liu 2006; Jiao and Li 2005; Bi and Jin 2007; Huang et al. 2007; Malossini et al. 2008; Pan et al. 2007; Li and Wang 2006, 2007; Shu and He 2007; Qin et al. 2007; Yu et al. 2006; Su et al. 2010; Wu et al. 2009; Wang and Li 2010; Jiao et al. 2008; Niu et al. 2009; Zhang et al. 2008; Du et al. 2007). The class bQIEAh concentrates on the interactions between bQIEA and CGAs, immune algorithms and particle swarm optimization (PSO). These algorithms can essentially be divided into three groups: immune bQIEA (bQIEAi), PSO-based bQIEA (bQIEApso) and CGA-based bQIEA (bQIEAcga). We review these in turn.

Fig. 10 Pseudocode algorithm for bQIEAi

```
Begin
            t ← 0
(i)-(iv)    The first four steps of bQIEAcm
            While (not termination condition) do
                t ← t+1
(v)             Make P(t) by observing the states of Q(t -1)
(vi)            Perform immune operation on P(t)
(vii)           Evaluate P(t)
(viii)          Update Q(t) using Q-gates
(ix)            Store the best solutions b among P(t)
            End
End
```

(a) bQIEAi

The bQIEAi was studied in Li et al. (2004a, 2004c, 2005b, 2006), Li and Jiao (2005, 2007, 2008), You et al. (2006a, 2006b, 2006c, 2007), Du et al. (2007), Li and Liu (2006), Jiao and Li (2005), Bi and Jin (2007), Jiao et al. (2008), Wu et al. (2009), Niu et al. (2009), where immune concepts were introduced into bQIEA model. The pseudocode algorithm for bQIEAi can be summarized as the nine steps listed in Fig. 10, in which bQIEAi reuses the first five steps and the steps (vi)–(viii) of bQIEAcm as steps (i)–(v) and (vii)–(ix), respectively. So the following description focuses on step (vi). The immune operation consists of two steps: vaccination and immune selection (Li et al. 2004c). Based on prior knowledge about the problem, a vaccination is used to modify certain genes of some genotype individuals, and then the immune selection is implemented using the following two processes. One is the *immune test*, i.e., calculating the fitness of the vaccinated individuals. The other is *annealing selection*, in which an individual $\boldsymbol{q}_i$ $(i = 1, 2, \ldots, n)$ is chosen as offspring with probability $P(\boldsymbol{q}_i)$ given by

$$P(\boldsymbol{q}_i) = \frac{e^{(f(\boldsymbol{q}_i)/T_k)}}{\sum_{i=1}^{n} e^{(f(\boldsymbol{q}_i)/T_k)}}, \tag{12}$$

where $f(\boldsymbol{q}_i)$ is the corresponding fitness of $\boldsymbol{q}_i$ and T_k is called an *annealing temperature*. The value T_k is taken from a strictly decreasing sequence $\{T_k\}$ of values converging 0 (Zhang et al. 1997). In Li et al. (2004c), the sequence

$$T_k = \ln\left(\frac{T_0}{k} + 1\right), \quad T_0 = 100 \tag{13}$$

was used, where k is the evolutionary generation.

In Li et al. (2004c), comparisons were drawn between bQIEAi and immune GAs and bQIEAo to show the advantages of bQIEAi, using knapsack problems. Li et al. (2006) discussed the application of bQIEAi to multiuser detection, which is an important and difficult optimization problem in directed-sequence code-division multiple-access (DS-CDMA) communication systems. bQIEAi was compared with the optimal multiuser detector (OMUD), matched filters detector (MFD) and immune GAs. Experimental results show that bQIEAi obtained better performances for DS-CDMA systems than the other three techniques (Li et al. 2006). In addition to the above applications, the applicability of bQIEAi to other problems is summarized in Table 5.

Fig. 11 Pseudocode algorithm for bQIEApso

Begin
$t \leftarrow 0$
(i) - (vi) The first six steps of bQIEAo
(vii) Update $Q(t)$
(viii) Store the best solutions among $P(t)$ and $B(t\text{-}1)$ into $B(t)$ and the best solution **b** among $B(t)$
End

Remarks The main aim of introducing immune concepts into bQIEA is to make good use of prior knowledge in optimization problems so as to improve the bQIEA performance. Therefore, bQIEAi mainly suits a class of optimization problem with available prior information. Based on the principles of immune operators, it is more like a special kind of local search technique, which can improve the bQIEA performance to a considerable degree. However, parameter setting, more analysis and systematic comparisons between bQIEAi and bQIEAo and other algorithms are required.

(*b*) *QIEApso*
bQIEApso is the result of the interplay between bQIEA and PSO, studied in Pan et al. (2007), Wang et al. (2005d, 2007c), Yu et al. (2006), Huang et al. (2007). The pseudocode algorithm for bQIEApso is given in Fig. 11, in which only one step (vii) is different from bQIEAo in Fig. 5. In this step, bQIEApso uses one of two approaches to update the population $Q(t)$ as explored in Wang et al. (2007c) and Yu et al. (2006). They are shown in Eqs. 14 and 15, respectively.

$$\begin{cases} \theta_{ij}^{t} = c_1(p_{ij}^{t} - x_{ij}^{t}) + c_2(p_{gj} - x_{ij}^{t}) \\ \begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij}^{t} & -\sin\theta_{ij}^{t} \\ \sin\theta_{ij}^{t} & \cos\theta_{ij}^{t} \end{bmatrix} \begin{bmatrix} \alpha_{ij}^{t} \\ \beta_{ij}^{t} \end{bmatrix} \end{cases} \tag{14}$$

$$\begin{cases} \alpha_{ij}^{t+1} = \cos\theta_{ij}^{t+1}, \qquad \beta_{ij}^{t+1} = \sin\theta_{ij}^{t+1} \\ \theta_{ij}^{t+1} = \theta_{ij}^{t} + c_1(p_{ij}^{t} - x_{ij}^{t}) + c_2(p_{gj} - x_{ij}^{t}) \end{cases} \tag{15}$$

where c_1 and c_2 are two positive constants; x_{ij}, p_{ij} and p_{gj} $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, m)$ are the jth element of position vector of the ith particle, the jth element of the best position vector of the ith particle obtained based on its own experience and the jth element of the best position vector of the ith particle based on the overall swarm's experience, respectively.

Wang et al. (2007c) utilized two well-known combinatorial optimization problems, knapsack problems and traveling salesman problems, to test the advantages of bQIEApso over bQIEAo. In Yu et al. (2006), applications of bQIEApso to knapsack problems, function optimization and multiuser detection in DS-CDMA communication systems were discussed to draw comparisons between two versions of bQIEApso, versus PSO and bQIEAo. Experimental results show that bQIEApso performs better than bQIEAo and PSO. Additional work related to bQIEApso is listed in Table 5.

Remarks bQIEA and PSO are both heuristic search algorithms. bQIEA was developed based on concepts and principles of quantum computing, whereas PSO was derived from the simulation of social behavior. Both of them have their own features, so investigating their interactions is an attractive issue. In Pan et al. (2007), Wang et al. (2005d, 2007c), Yu et al. (2006), Huang et al. (2007), two paradigms were designed using the evolutionary strategy of PSO to produce offspring for bQIEA, instead of a lookup table for Q-gates, whence the algorithm structure is simplified. Nevertheless, a systematic analysis of this combined approach needs to be undertaken both theoretically and experimentally so that the characteristics of bQIEApso can be clearly understood. Furthermore, more convincing experiments could be carried out on various problems to compare bQIEApso with bQIEAo, improved PSO and other good optimization methods.

(b) QIEAcga
Combining bQIEAcm with CGA, a new framework (bQIEAcga) was presented in Wang et al. (2005a). In bQIEAcga, bQIEAcm and CGA were applied to search for solutions in micro-space and in macro-space, respectively. In Wang et al. (2005a), several numeric optimization problems, and an application for estimating parameters of a non-linear state-space model and a Hammerstein model, were taken as examples to show that bQIEAcga performs better than bQIEAo. Li and Wang (2007) extended bQIEAcga to multi-objective flow shop scheduling problems. Experiments conducted on nine testing problems and five random instances show that bQIEAcga is superior to permutation-based GAs in terms of several metrics including overall nondominated vector generation, distance metrics, Tan's spacing, maximum spread, average quality and running time. Further results concerning bQIEAcga are shown in Table 5, based on work in Li and Wang (2006, 2007).

Remarks Strictly speaking, bQIEAcga is a hierarchical EA because its bQIEA and CGA components are performed independently and cannot interact with each other. The transformation between the Q-bit and binary (or numeric) representation is unidirectional, i.e., from bQIEA to CGA. Due to the many genetic operators in bQIEAcga, it is really difficult to analyze the role and contribution of each operator to the overall performance, and there is reason to suspect that bQIEAcga may be a rather time-consuming optimization algorithm. So further work is needed to investigate the computational complexity aspects of this algorithm in order to prove its potential.

2.2.5 Summary of bQIEA

This section provides an overview on bQIEA and its variants. Initially, the comparisons between different types of bQIEA are drawn in Fig. 12, where similarities and differences, their performances and suggestions for further work are summarized. Subsequently, some of the main bQIEA flavors studied in the literature and the problems they have been applied to are summed up in Tables 3–5. Finally, a short remark concerning bQIEA is given.

Unlike numeric, binary or symbolic representations, bQIEA uses Q-bit representation to describe the individuals, and this representation is beneficial for population

Types		Algorithms	Similarities	Differences	Advantages	Suggestions
bQIEAo	/	Fig.5	● Q-bit representation for chromosomes ● Binary observation process for linking phenotypes with genotypes	● Q-gate in (6) ● Using (8) to adjust Q-gates ● Migration	● Q-bit representation ● Binary observation ● Q-gate update process ● Success in knapsack problems	● Dynamically adjust Q-gate parameters ● Other applications
bQIEAcm	/	Fig.6		● Q-gate in (6) ● Using (8) to adjust Q-gates ● Quantum crossover and quantum mutation	● Probably improving population diversity, especially in the latter stage of evolution ● Applicability to multiple sequence alignment problems	● Analysis of crossover and mutation operators ● More applications
bQIEAn	/	Fig.9		● Q-gate in (6) ● Using (10) to adjust Q-gates ● Migration ● Catastrophe	● Reducing the number of parameters of Q-gates ● Applicability to feature selection	● Systematic analysis on the modified Q-gates ● Other applications
bQIEAh	bQIEAi	Fig.10		● Q-gate in (6) ● Using (8) to adjust Q-gates ● Immune operator	● Introducing the prior knowledge of problems using immune operator ● Applicability to multiuser detection	● Systematic analysis on immune operator ● Applications
	bQIEApso	Fig.11		● Using (14) or (15) to generate offspring	● Using evolutionary strategy of PSO to update Q-gates instead of a lookup table ● Applicability to traveling salesman problems	● Systematic analysis of their combination mechanism ● More applications
	bQIEAcga	/		● Q-gate in (6) ● Using (8) to adjust Q-gates ● Selection operator ● Quantum crossover and quantum mutation ● CGA	● Searching solutions in two different spaces ● Success in flow shop scheduling problems	● Analysis on the role of each genetic operator ● Analysis of computational complexity ● Other applications

Fig. 12 Comparisons of various types of bQIEA

diversity. By introducing a binary observation process, a connection between a Q-bit and binary solutions was built into bQIEA. Instead of crossover, recombination and mutation operators in CGA, bQIEA adopted a Q-gate as its evolutionary operator to implement the evolutionary process. As compared with CGA, bQIEA can balance well between exploration and exploitation, and even with a small number of individuals, bQIEA can explore the search space. However, several research issues listed in Fig. 12 are worth discussing further with respect to bQIEA. When bQIEA is applied to solve numerical optimization problems, there are disadvantages: Hamming cliffs, discretization error and computational complexity. In the case of binary observation, we can define a Hamming distance between the binary codes of adjacent integers. Although gray codes can alleviate the problem, the Hamming distance does not increase monotonously with the difference in integer values. Thus, this phenomenon introduces Hamming cliffs at other levels (Srinivas and Patnaik 1994). In bQIEA, a real-valued variable corresponds to a string of Q-bits. When bQIEA is employed to solve high-dimensional numerical optimization problems, the binary observation and Q-gate update are time-consuming processes.

Hence, how to develop a QIEA for numerical optimization is an ongoing issue.

Table 3 Summarization of the bQIEA work. ‘ >′ means better than

Types	Problems	References	Contributions	Compared results
bQIEAo	Knapsack problem	Han and Kim (2002) Han and Kim (2004)	bQIEAo	bQIEAo>CGA
		Zhang and Gao (2007a) Han and Kim (2003a) Han and Kim (2003b) Platelt et al. (2007)	Parameter setting for Q-gates	bQIEAo>CGA
		Han et al. (2001)	Applicability to a parallel scheme	bQIEAo>CGA
		Kim et al. (2006)	Multiobjective	bQIEAo>NSGA2
		Zhou and Sun (2005)	Single-chromosome bQIEAo	bQIEAo>CGA
		Imabeppu et al. (2008)	Introducing pair swap into bQIEAo	/
		Zhao et al. (2006)	Application	/
		Li et al. (2009)	Convergence performance comparisons of 3 Q-gates	bQIEAo>MOEA, SPEA2, NSGA2
	Function optimization	Han and Kim (2004)	Stopping criteria, H_{ϵ} gate & setting of initial values	bQIEAo>CEP and FEP (Yao et al. 1999)
		Khorsand (2005)	Parameter setting	bQIEAo>CGA
		Chen et al. (2004)	Chaos update method for Q-gate	bQIEAo>CGA
		Han and Kim (2006)	Performance analysis	/
	Disk allocation	Kim et al. (2003)	Application	bQIEAo>CGA-based disk allocation
	Face verification or detection	Jang et al. (2004a) Jang et al. (2003) Jang et al. (2004b)	bQIEAo-based classifiers	bQIEAo>Distance-from-face space & maximum likelihood classifiers
	Clustering	Zhou et al. (2006b) Zhou et al. (2005) Zhou et al. (2006a)	Applications	bQIEAo>K-means, self-organizing maps
	SVM parameter selection	Luo et al. (2008)	Application	bQIEAo>Cross-validation approach
	Multiple sequence alignment	Huo and Stojkovic (2006) Huo and Stojkovic (2007)	Application	bQIEAo>CLUSTAL (Eddy 2009), Sequence alignment by CGA
	Image edge detection	Li et al. (2004b)	Applicability to a parallel scheme	/
	Blind source separation	Yang et al. (2003a) Yang et al. (2003b)	Applicability to a parallel scheme	bQIEAo>CGA
	Bandwidth adaptation	Xiao et al. (2006)	Application	/
	Image segmentation	Liu et al. (2005)	Application	bQIEAo>CGA

Table 3 (*Continued*)

Types	Problems	References	Contributions	Compared results
	Neural network training	Ganesh and Singhal (2005) Akbarzadeh-T (2005) Lu et al. (2008)	Application	bQIEAo>CGA and PSO
	Minimal reduct	Lv and Liu (2007)	Application	bQIEAo>CGA
	TSP	Feng et al. (2006)	Application	/
	Pattern design	Khorsand (2006)	Application	/
	Real & reactive power dispatch	Vlachoglannis (2008) John and John (2009)	Probability distribution of Q-bit individuals	bQIEAo>ACO, EGA, SA, HPSO, PSOPC, CLONEPAC, CGA
	State assignment for FSM	Araujo et al. (2008)	Application	bQIEAo>CGA, NOVA
	QoS multicast routing problem	Xing et al. (2009b) Xing et al. (2009a)	Multigranularity adaptive evolution methods for Q-gate	QIEAo>CGA, CQGA

2.3 Real observation QIEA

In Zhang and Rong (2007c) and Liu et al. (2008), a *real-observation QIEA* (rQIEA) was proposed to solve global numerical optimization problems with continuous variables. The pseudocode algorithm for rQIEA is illustrated in Fig. 13. The detailed explanation of the rQIEA algorithm is as follows.

(i). In the "initialize $Q(t)$" step, a population $Q(0)$ with n Q-bit individuals is produced, $Q(t) = \{\boldsymbol{q}_1^t, \boldsymbol{q}_2^t, \ldots, \boldsymbol{q}_n^t\}$, at the generation moment $t = 0$, where $\boldsymbol{q}_i^t$ $(i = 1, 2, \ldots, n)$ is an arbitrary individual in $Q(t)$, represented as

$$\boldsymbol{q}_i^t = \begin{bmatrix} \alpha_{i1}^t | \alpha_{i2}^t | \cdots | \alpha_{im}^t \\ \beta_{i1}^t | \beta_{i2}^t | \cdots | \beta_{im}^t \end{bmatrix}, \tag{16}$$

where m is the number of Q-bits, which corresponds to the number of variables to optimize. The value α_{ij}^t $(j = 1, 2, \ldots, m)$ is randomly chosen between -1 and 1, and the value β_{ij}^t $(|\beta_{ij}^t|^2 = 1 - |\alpha_{ij}^t|^2)$ is either positive or negative, which means that rQIEA starts a search process from a random point.

(ii). In this step, a real observation is used. By observing the states of $Q(t)$, this step makes real solutions in $P(t)$, where $P(t) = \{\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_n^t\}$ at generation t. One real solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \ldots, n)$ is a real number string of length m, i.e. $\boldsymbol{x}_i^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{im}^t\}$, where x_{ij}^t $(j = 1, 2, \ldots, m)$ is a real number in the range [0, 1] and is formed by selecting a real number between 0 and 1 for each Q-bit using the probability $|\alpha_{ij}^t|^2$ $(j = 1, 2, \ldots, m)$, of $\boldsymbol{q}_i^t$. For the probability amplitude $[\alpha_{ij}^t \; \beta_{ij}^t]^T$ of the ith Q-bit in $\boldsymbol{q}_i^t$, a random number r in the range [0, 1] is generated. If $r \le 0.5$, the corresponding observed value is set to $|\alpha_{ij}^t|^2$,

Table 4 Summarization of the bQIEA work. ‘ $>'$ means better than

Types	Problems	References	Contributions	Compared results
bQIEAo	Job shop scheduling problem	Gu et al. (2009a); Gu et al. (2009b)	Introducing strategies (competitive hunter, cooperative surviving & big fish eating small fish)	bQIEAo>CGA
	Unit commitment problems	Jang et al. (2009) Jeong et al. (2009) Lau et al. (2009)	Simplified update method for Q-gate	/
bQIEAn	Feature selection	Zhang et al. (2006) Zhang et al. (2004a) Zhang et al. (2004b)	Modified Q-gate	bQIEAn>bQIEAo, two conventional approaches
	Logistics distribution	Gao et al. (2006) Wang et al. (2007b)	Application	bQIEAn>bQIEAo, CGA
	FIR filter design	Zhang et al. (2003c)	Catastrophe operator	bQIEAn>bQIEAo, CGA
	IIR digital filter design	Zhang et al. (2003b) Zhang et al. (2003a)	Q-gate, convergence analysis, Applicability to parallel approach	bQIEAn>CGA
	Time-frequency analysis	Zhang and Rong (2007b) Zhang and Rong (2006)	Application	bQIEAn>Greedy algorithm
bQIEAcm	Knapsack problem	Yang et al. (2004a) Yang et al. (2004b) Meshoul et al. (2005b)	Quantum crossover mutation; Multiobjective bQIEAcm	bQIEAcm>CGA, Greedy algorithm bQIEAcm>SPEA
	Multiple sequence alignment	Abdesslem et al. (2006) Meshoul et al. (2005a)	Applications	bQIEAcm>CLUSTAL, DIALIGN, PROALIGN, MAFF (Eddy 2009; Notredame et al. 1998)
	Image registration	Talbi et al. (2004a) Talbi et al. (2004c)	Quantum crossover Quantum mutation	/
	Image detection	Li et al. (2005a)	Quantum crossover Quantum mutation	bQIEAcm>CGA
	TSP	Talbi et al. (2004b)	Application	/
	Flow shop scheduling	Wang et al. (2005c)	Application	bQIEAcm>NEH
	Function optimization	Li and Zhuang (2002) Yang and Jiao (2003) Yang and Ding (2007)	Quantum crossover and Quantum mutation	bQIEAcm>CGA, bQIEAo
	Blind source separation	Xu et al. (2005) Yang et al. (2005)	Applications	/ bQIEAcm>CGA
	Embedded system	Guo et al. (2007)	Applications	bQIEAcm>bQIEAo
	Grid resource allocation	Shu (2007)	Applications	/
	Hardware–software cosynthesis	Wei et al. (2008)	Applications	/
	Evolving quantum circuits	Ding et al. (2008)	Applications	/
	Fuzzy NN training		Quantum crossover Quantum mutation	/

Table 5 Summarization of the bQIEA work. '$>$' means better than

Types	Problems	References	Contributions	Compared results
bQIEAh	Knapsack problem	Li et al. (2004c) Li et al. (2005b) You et al. (2006b) Li and Jiao (2007) Jiao et al. (2008) Wu et al. (2009) Niu et al. (2009)	Immune algorithm + bQIEAo	bQIEAi>Intelligent EA, bQIEAo, SPEA, NSGA, VEGA, NPGA, CGA, FEP, OGA/Q, BGA, CMA-ES, AEA
		Wang et al. (2005d) Wang et al. (2007c)	PSO + bQIEAo	bQIEApso>bQIEAo
		Zhang et al. (2008)	P systems+bQIEAo	QEPS>bQIEAo
	Function optimization	You et al. (2006a) You et al. (2006c) You et al. (2007) Li and Jiao (2005) Li and Liu (2006)	Immune algorithm + bQIEAcm	bQIEAi>bQIEAcm, OGA/Q, breeder GA
		Li et al. (2004a)	Clonal algorithm + bQIEAo	/
		Wang et al. (2005a)	CGA + bQIEAcm	bQIEAcga>bQIEAcm
		Qin et al. (2007)	Multi-agent + bQIEAcm	bQIEAh>MAGA
		Wang et al. (2007a)	ACO + bQIEAcm	bQIEAh>bQIEAo,PSO
		Huang et al. (2007)	PSO + bQIEAo	bQIEApso>CGA
	Flow shop scheduling	Wang et al. (2005b)	CGA + bQIEAcm for single-objective problems	bQIEAcga>NEH, bQIEAcm, PGA
		Li and Wang (2007) Li and Wang (2006)	CGA + bQIEAcm for multi-objective problems	bQIEAcga>PGA
		Shu and He (2007)	Simulated annealing + bQIEAcm	bQIEAh>CGA, bQIEAcm
	Multiuser detection	Li and Jiao (2008) Li and Jiao (2005) Li et al. (2006) Jiao et al. (2008)	Immune + bQIEAo	bQIEAi>MFD, OMUD, IGA
		Yu et al. (2006)	PSO + bQIEAo	bQIEApso>bQIEAo, PSO
	SVM parameter selection	Pan et al. (2007)	PSO + bQIEAo	bQIEApso>PSO bQIEApso-based SVM>SVM
	Image segmentation	Bi and Jin (2007)	Immune + bQIEAcm	bQIEAi>CGA
	Tourism emergency event prediction	Du et al. (2007)	Immune + bQIEAcm	
	Parameter of estimation in chaotic systems	Wang and Li (2010) Su et al. (2010)	Differential evolution + bQIEAo	bQIEAh>Ant-Miner, CN2

Fig. 13 Pseudocode algorithm for rQIEA

Begin
$t \leftarrow 0$
(i) Initialize $Q(t)$
(ii) Make $P(t)$ by observing the states of $Q(t)$
(iii) Evaluate $P(t)$
(iv) Store the best solution among $P(t)$ into $b(t)$
While (not termination condition) **do**
$t \leftarrow t+1$
(v) Make $P(t)$ by observing the states of $Q(t$ -1)
(vi) Evaluate $P(t)$
(vii) Store the best solution among $P(t)$ and $b(t$-1) into $b(t)$
(viii) Update $Q(t)$ using Q-gates
(ix) Recombination
End
End

otherwise, the observed value is set to $|\beta_{ij}^t|^2$. Because $|\alpha_{ij}^t|^2$ and $|\beta_{ij}^t|^2$ are in the range [0, 1], a simple mapping need be employed to transform the range [0, 1] into any desired range of optimization variable.

(iii). Each real solution $\boldsymbol{x}_i^t$ $(i = 1, 2, \ldots, n)$ of $P(t)$ is evaluated thus obtaining its fitness.

(iv). The initial best solution is selected among $P(t)$ and stored into $b(t)$.

(v). Real solutions in $P(t)$ are generated by observing the states $Q(t-1)$ as step (ii).

(vi). Each real solution is evaluated for the fitness value as step (iii).

(vii). The best solution among $P(t)$ is selected and stored into $b(t)$.

(viii). In this step, the probabilities of all Q-bits in population $Q(t)$ are updated by using Q-gates, i.e., the jth Q-bit in the ith Q-bit individual $\boldsymbol{q}_i^t$, $j = 1, 2, \ldots, m$, $i = 1, 2, \ldots, n$, is modified by using the current Q-gate $G_{ij}^t(\theta)$

$$G_{ij}^t(\theta) = \begin{bmatrix} \cos\theta_{ij}^t & -\sin\theta_{ij}^t \\ \sin\theta_{ij}^t & \cos\theta_{ij}^t \end{bmatrix}, \tag{17}$$

where θ_{ij}^t is defined as

$$\theta_{ij}^t = k \cdot f(\alpha_{ij}^t, \beta_{ij}^t), \tag{18}$$

where k is a coefficient

$$k = \frac{\pi}{100 + mod(t, 100)}, \tag{19}$$

In Eq. 18, $f(\alpha_{ij}^t, \beta_{ij}^t)$ is a function for determining the search direction of rQIEA to a global optimum and can be obtained from Table 6, in which $\xi_b = \arctan(\beta_b/\alpha_b)$ and $\xi_{ij}^t = \arctan(\beta_{ij}^t/\alpha_{ij}^t)$, where α_b, β_b are the probabilities of the best solution stored in $b(t)$ and α_{ij}^t, β_{ij}^t are the probabilities of the current solution.

(ix). The recombination operation on Q-bits is shown in Fig. 14, in which $\boldsymbol{q}_i$, $\boldsymbol{q}_j$ $(i, j = 1, 2, \ldots, m)$, are any two arbitrary individuals of the population $Q(t)$,

Table 6 Look-up table for the function $f(\alpha, \beta)$, where $\xi_b = \arctan(\beta_b/\alpha_b)$, and $\xi_{ij}^t = \arctan(\beta_{ij}^t/\alpha_{ij}^t)$, and α_b, β_b are the probabilities of the best solution stored in $b(t)$ and α_{ij}^t, β_{ij}^t are the probabilities of the current solution

$\xi_b > 0$	$\xi_{ij}^t > 0$	$f(\alpha_{ij}^t, \beta_{ij}^t)$	
		$\xi_b \geq \xi_{ij}^t$	$\xi_b < \xi_{ij}^t$
True	True	$+1$	-1
True	False	$\text{sign}(\alpha_b \cdot \alpha_{ij}^t)$	$\text{sign}(\alpha_b \cdot \alpha_{ij}^t)$
False	True	$-\text{sign}(\alpha_b \cdot \alpha_{ij}^t)$	$-\text{sign}(\alpha_b \cdot \alpha_{ij}^t)$
False	False	$+1$	-1
$\xi_b, \xi_{ij}^t = 0$ or $\pm\pi/2$		± 1	± 1

$$\begin{cases} q_i & \begin{bmatrix} \alpha_{i1} \big| \ldots \big| \alpha_{ih} \big| \ldots \big| \alpha_{ih} \big| \ldots \big| \alpha_{im} \\ \beta_{i1} \big| \ldots \big| \beta_{ih} \big| \ldots \big| \beta_{ih} \big| \ldots \big| \beta_{im} \end{bmatrix} \\ q_j & \begin{bmatrix} \alpha_{j1} \big| \ldots \big| \alpha_{jh} \big| \ldots \big| \alpha_{jh} \big| \ldots \big| \alpha_{jm} \\ \beta_{j1} \big| \ldots \big| \beta_{jh} \big| \ldots \big| \beta_{jh} \big| \ldots \big| \beta_{jm} \end{bmatrix} \end{cases} \Rightarrow \begin{cases} q_i' & \begin{bmatrix} \alpha_{i1} \big| \ldots \big| \beta_{jh} \big| \ldots \big| \beta_{jh} \big| \ldots \big| \alpha_{im} \\ \beta_{i1} \big| \ldots \big| \alpha_{jh} \big| \ldots \big| \alpha_{jh} \big| \ldots \big| \beta_{im} \end{bmatrix} \\ q_j' & \begin{bmatrix} \alpha_{j1} \big| \ldots \big| \beta_{ih} \big| \ldots \big| \beta_{ih} \big| \ldots \big| \alpha_{jm} \\ \beta_{j1} \big| \ldots \big| \alpha_{ih} \big| \ldots \big| \alpha_{ih} \big| \ldots \big| \beta_{jm} \end{bmatrix} \end{cases}$$

Fig. 14 The recombination in rQIEA

respectively; $\boldsymbol{q'}_i$, $\boldsymbol{q'}_j$ are the recombined individuals, respectively; h and h' ($h, h' \in [1, m]$, and $h' \geq h$) are any two arbitrary positions in $\boldsymbol{q}_i$ and $\boldsymbol{q}_j$, respectively.

Remarks By extending two states '1' and '0' to an arbitrary pair of states between '1' and '0' in quantum system, rQIEA is characterized by the modified Q-bit representation for briefly representing a Q-bit individual, the use of real observation for generating real-valued solutions from Q-bit individuals, and the modified Q-gate for adaptively guiding the individuals toward better solutions. In rQIEA, there is only one parameter to adjust in the modified Q-gate. This issue was preliminarily discussed in Zhang and Rong (2007a). In contrast, bQIEA's Q-gate has eight angle parameters which remain unchanged throughout the evolutionary process and have to be prescribed. Relative to bQIEA, rQIEA is more suitable for a wide range of real-world numerical optimization problems, as shown by the experiments reported in the next section. rQIEA may be appropriate to some problems such as optimization design of digital filters, system identification, controller design and signal processing.

2.4 iQIEA

Several EAs for numerical optimization problems are called QIEAs in Abs da Cruz et al. (2004, 2006, 2007, 2005), Sailesh Babu et al. (2008), Al-Othman et al. (2007), Fan et al. (2007), Li and Li (2008), Alfares et al. (2004), Alfares and Esat (2006), Zhang and Gao (2007b), but they are a bit different from the above QIEA that are characterized by the Q-bit representation, observation process and Q-gates. In this paper, they are grouped under the heading iQIEA. In what follows two representative variants of iQIEA (Sailesh Babu et al. 2008; Abs da Cruz et al. 2007) are taken as representative of the iQIEA work done in the literature.

In Abs da Cruz et al. (2007), a pair of values, (ρ, σ), consisting of the mean ρ and the width σ of a square pulse, was proposed to represent a gene. A probability density function and then a cumulative distribution function of a train of square pulses were calculated to connect pulse representation with real-valued variables. The evolutionary rules are made up of three steps: crossover, translation and resize. The crossover operator exchanges some individuals in the current population and the searched best population. Translation and resize operators are applied to modify the mean ρ and the width σ of the square pulse, respectively. Experiments conducted on four benchmark functions show that the algorithm is competitive to stochastic GA, fast evolutionary programming (Yao et al. 1999) and PSO.

In addition to using the Q-bit representation, a Q-gate and migration operator as in bQIEAo, Sailesh Babu et al. (2008) presented two neighbourhood operators, NO1 and NO2, to produce neighbourhood solution strings and choose the best out of them. NO1 and NO2 play the same roles as the observation process of rQIEA, i.e., transforming encoded genotypes to real-valued candidate solutions. The algorithm was tested by using three load dispatch problems and experimental results verify its advantages over several load dispatch approaches reported in the literature, such as variants of simulated annealing, hybrid PSO.

The main points relating to iQIEAs and the problems they have been applied to are summarized in Table 7.

Remarks The evolutionary operators of iQIEA in Abs da Cruz et al. (2007) are performed on pulse parameters instead of real values, which is similar to the way in which evolutionary rules are carried out on QIEA Q-bits. However, the iQIEA algorithm in Abs da Cruz et al. (2007) differs from that used by QIEAs and seems to be more of an estimation of distribution algorithm (EDA) (Santana et al. 2008; Pelikan et al. 2000, 2002; Baluja 1994; Baluja and Davies 1997; De Bonet et al. 1997; Harik 1999; Harik et al. 1998; Larrañaga et al. 2000; Mühlenbein and Mahnig 1998, 1999; Pelikan and Mühlenbein 1999) due to the calculation of the probability density function and cumulative distribution function. It makes sense, therefore, that future studies should concentrate on comparing the iQIEA algorithm with QIEAs and EDAs to better estimate their relative performances. To build the connection between Q-bit representation and real-valued variables, two neighborhood operators in Sailesh Babu et al. (2008) were applied to replace the binary observation process of bQIEA, however, they are rather complicated and the probabilistic observation of QIEAs remains of little importance. So the practicability and usability of the iQIEA (Sailesh Babu et al. 2008) is questionable and future studies should test how powerful and useful this approach is.

2.5 Summary of QIEAs

In summary, a brief overview of the three types of QIEA discussed in Sects. 2.2–2.4 is given in Fig. 15, which illustrates their similarities and differences, together with relative advantages and some suggestions for further work.

Table 7 Summarization of the iQIEA work, ‘ >′ means better than

No.	Ref.	Main points	Problems	Compared results
1	Abs da Cruz et al. (2004) Abs da Cruz et al. (2006) Abs da Cruz et al. (2007) Abs da Cruz et al. (2005)	• Pulse representation; • Generation of real-valued candidate solutions using probability density functions and cumulative distribution functions; • Evolutionary rules: crossover, translation and resize operators	Function optimization	iQIEA>CGA, PSO, stochastic GA, FEP
	Fan et al. (2007)		Option pricing model calibration	/
2	Alfares et al. (2004) Alfares and Esat (2006)	• Triploid representation; • Producing candidate solutions using register process; • Hadamard gates	Gear train design; Pressure vessel design; Tension/Compression spring design; Welded bean design	iQIEA>Augmented Lagrange, Branch and bound, CGA, PSO, DE
	Al-Othman et al. (2007)		Economic dispatch in power system	iQIEA>CGA, Quadratic programming
3	Sailesh Babu et al. (2008)	• Q-bit representation; • Generating candidate solutions using two neighbourhood operators; • Q-gates; • Migration operator	Economic load dispatch in power systems	iQIEA>SA, Hybrid PSO, Hybrid stochastic search
4	Li and Li (2008)	• Spherical coordinate representation; • Coordinate values corresponding to candidate solutions; • Coordinate transformation matrix	Function optimization; Neural network training	iQIEA>bQIEAo, CGA
5	Zhang and Gao (2007b)	• Triploid representation; • Complementary double mutation operator; • Q-gate; • Discrete crossover; • Hill climbing selection	Function optimization	iQIEA>Improved evolution strategy

3 Experimental results

In order to better illustrate the performance of and differences among the various types of QIEA discussed in this paper, we have conducted a small experimental study. We have also conducted experiments that compare QIEAs with other state-of-the-art EAs presented in the recent literature. The QIEA algorithms were implemented by using Matlab. All the programs involved in this section were made by the author. Upon the readers’ request, the author can provide the source codes of the programs.

Types	Commons	Differences	Advantages	Suggestions
bQIEA	• Q-bit representation • Q-gate in (6)	• Binary observation • Mainly using (8) to adjust Q-gates • Genetic operators: migration, crossover, mutation, etc.	• More suitable for combinatorial optimization • Solving benchmark and application problems	• More comparisons with state-of-the-art combinatorial EAs • More applications
rQIEA		• Real observation • Using (18) to adjust Q-gates • Recombination operator	• Suitable for numeric optimization • Mainly focusing on benchmark problems	• Comparisons with state-of-the-art numeric EAs • Applications
iQIEA		• Various representations: pulse, triploid, Q-bit, spherical co-ordinate, etc. • Schemes linking phenotypes with genotypes: register process, neighborhood operators, etc. • Evolutionary operators: Q-gates, crossover, translation, resize, migration, selection, etc.	• Suitable for numeric optimization • Mainly focusing on engineering optimization problems	• Performance testing using more benchmark functions • Theoretical analysis

Fig. 15 Comparisons of three types of QIEAs

3.1 Comparisons of QIEAs

In this subsection, we focus on the knapsack problem, a well-known NP-hard combinatorial optimization problem (Han and Kim 2002), and use benchmark functions to test the performances of different variants of QIEAs.

3.1.1 Combinatorial optimization

The knapsack problem is the optimization problem that requires us to select a subset from a given set of items so that the profit $f(x)$ is maximum, where

$$f(x) = \sum_{i=1}^{m} p_i x_i \tag{20}$$

subject to

$$\sum_{i=1}^{m} w_i x_i \le C \tag{21}$$

where m is the number of the items given; C is the capacity of the given knapsack; p_i and w_i are the profit and weight of the ith item, respectively; and x_i is 0 or 1 (where $x_i = 1$ if and only if item i is one of those selected). In order to compare bQIEAo, bQIEAn, bQIEAcm and CGA, strongly correlated sets of data are considered. We

Table 8 Comparisons of bQIEAo, bQIEAcm, bQIEAn and CGA using the knapsack problems: the number of items 50, 200 and 400, the maximal number of generations 1000, the number of runs 30. BS, MBS, WS, STD and ET represent best solution, mean best solution, worst solution, standard deviation and elapsed time per run, respectively. The bold style highlights the best result for each criterion

Items	Criteria	CGA	bQIEAo	bQIEAcm	bQIEAn
50	BS	296.45	**312.17**	312.13	307.25
	MBS	287.29	**307.40**	306.86	304.24
	WS	282.00	**307.21**	302.24	299.23
	STD	3.06	**0.90**	1.80	2.77
	ET	**3.41**	36.91	30.37	39.03
200	BS	1047.98	**1178.22**	1173.18	1102.08
	MBS	1027.13	**1166.67**	1156.22	1090.64
	WS	1017.15	**1153.27**	1143.20	1077.45
	STD	7.34	7.11	6.79	**6.61**
	ET	**9.84**	142.25	129.29	149.76
400	BS	2120.54	**2341.36**	2336.41	2211.12
	MBS	2100.85	**2322.47**	2315.92	2190.67
	WS	2086.29	**2300.49**	2291.25	2165.62
	STD	**9.01**	11.52	10.46	11.41
	ET	**18.64**	284.93	270.86	301.51

take w_i to be uniformly random $[1, v]$ and $p_i = w_i + r$, where $v = 10, r = 5$. The average knapsack capacity $C = 0.5 \sum w_i$ is used. The data are unsorted and three knapsack problems with 50, 200, and 400 items are considered.

The pseudocode algorithms for the bQIEAo, bQIEAcm and bQIEAn have been shown in Figs. 5, 6 and 9, respectively. The CGA utilizes fitness proportional selection, two-point crossover and uniform mutation. As for bQIEAcm and CGA, the crossover probability has seven choices including 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9 and the mutation probability varies in the values of 0.001, 0.01, 0.05, 0.1. Thus, there are 28 combinations for bQIEAcm and CGA and the best results are selected. In all experiments, a random repair scheme is adopted. The termination criterion for all experiments is the maximal number of generations, 1000. The performances of the four algorithms are evaluated by using the criteria: the best solution and the worst solution searched within 1000 generations over 30 runs, the mean best solution, the standard deviation and the elapsed time per run. When the population size is set to 20, experimental results for the three cases of 50, 200, and 400 items are shown in Table 8, in which three versions of bQIEA produce much better results than CGA (bQIEAo and bQIEAn obtain the best and the worst results among the several versions of bQIEA, respectively). To further investigate the effects of population size on the four algorithms' performance, six additional cases, in which the population sizes are 10, 40, 60, 80, 100 and 200, respectively, are considered, in more experiments conducted on the knapsack problem with 200 items. The relationship between the population size and the mean best profits over 30 runs is illustrated in Fig. 16. The experimental results show that each of the four algorithms can achieve an increase of mean best

Fig. 16 Mean best profits as the population sizes

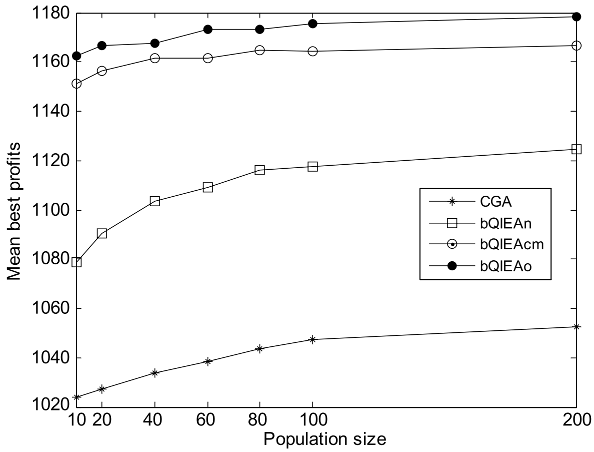


Table 9 Comparisons of bQIEAo-R, bQIEAo-H and rQIEA using six functions. The results of bQIEAo-R and bQIEAo-H are referred from Han and Kim (2004). The number of runs is 50. Mean, Std and NoFE represent the mean best, the standard deviation and the number of function evaluations, respectively. The bold style highlights the best result for each function

Problems		F_{Sph}	F_{Ack}	F_{Gri}	F_{Ras}	F_{Sch}	F_{Ros}
NoFE		1.5e+5	1.5e+5	2.0e+5	5.0e+5	9.0e+5	2.0e+6
bQIEAo-H	Mean	1.8e−4	2.5e−3	3.6e−2	3.9e−2	**3.8e−4**	1.2e+1
	Std	1.3e−4	8.1e−4	3.2e−2	1.9e−1	**3.0e−9**	1.8e+1
bQIEAo-R	Mean	4.3e−6	4.8e−4	5.8e−2	18.7	2.2e+2	7.2e+0
	Std	0.0e+0	0.0e+0	7.5e−2	7.4	1.6e+2	6.8e+0
rQIEA	Mean	**1.8e−30**	**2.3e−9**	**1.9e−15**	**1.6e−15**	3.7e+3	**1.3e−2**
	Std	**1.3e−30**	**1.0e−8**	**1.1e−15**	**7.5e−15**	5.3e+2	**3.8e−4**

profits when the population size varies from 10 to 200, and the four algorithms have similar climbing tendency.

3.1.2 Numeric optimization

Six benchmark numeric optimization problems with 30 dimensions, including *Sphere* (f_{Sph}), *Ackley* (f_{Ack}), *Griewank* (f_{Gri}), *Rastrigin* (f_{Ras}), *Schwefel* (f_{Sch}) and *Rosenbrock*, (f_{Ros}), were employed in Han and Kim (2004) to test the performance of two versions of bQIEAo, i.e., bQIEAo with a rotation Q-gate (bQIEAo-R) and bQIEAo with an H_ϵ Q-gate (bQIEAo-H). We use the same functions to conduct comparative experiments so as to draw a comparison between rQIEA and bQIEAo. The population size is set to 50 for rQIEA. The stopping criterion is the number of function evaluations. Each test function is performed with 50 independent runs. The mean best values and the standard deviations are recorded for each test function. Table 9 lists the statistical results, which show that rQIEA provides better results than two versions of bQIEAo in searching for optimal solutions and maintaining their robustness.

3.2 Comparisons with other EAs

In order to show the superiority of the QIEA over other techniques, this subsection provides experimental comparisons between QIEAs and other EAs. We firstly compare bQIEAo with polyploid GA (PGA) in terms of population diversity and then compare rQIEA with several state-of-the-art EAs.

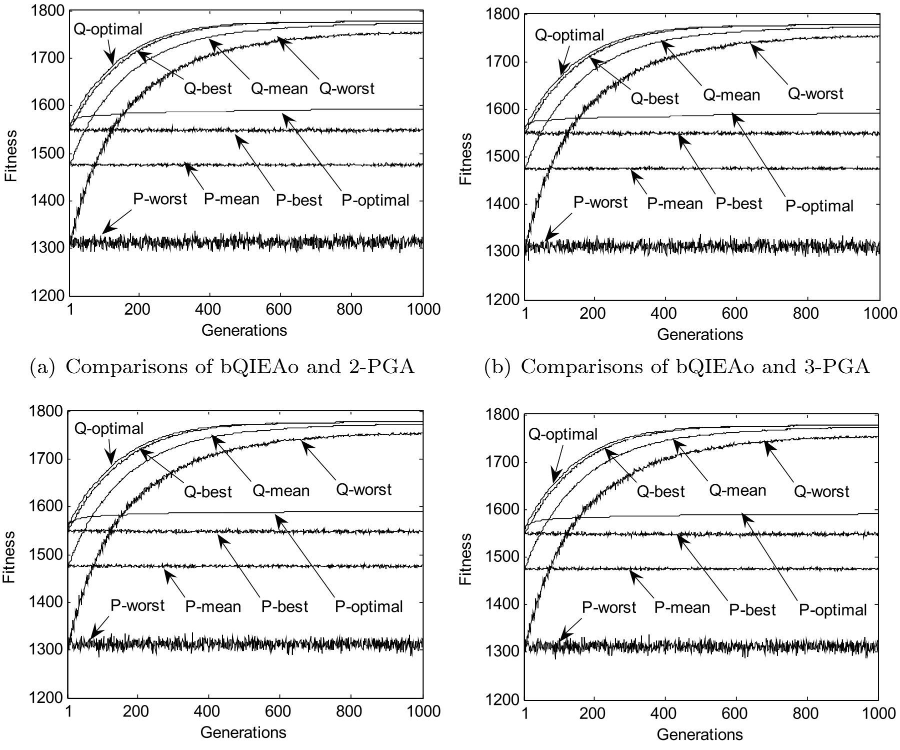
3.2.1 Comparisons between bQIEAo and PGA

Population diversity is very important for EAs to explore the search space (Goldberg 1989; Collingwood et al. 1996; Corne et al. 1996; Eshelman 1991; Chaiyaratana et al. 2007; Koumousis and Katsaras 2006; Lozano et al. 2005, 2008). PGA (Goldberg 1989) is regarded as an algorithm with good population diversity. So we compare bQIEAo with PGA with respect to population diversity in the process of finding an optimal solution. The experiments are carried out on the knapsack problem with 300 items described in Sect. 3.1. PGA uses rank-based selection, two-point crossover and uniform mutation operators. The crossover probability has 12 choices: 0.0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, and the mutation probability is assigned one of 14 values: 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. Thus, there are 168 combinations and the best result is selected to compare PGA with bQIEAo. In the experiments, diploid, triploid, tetraploid and pentaploid PGAs are employed and are labeled 2-PGA, 3-PGA, 4-PGA and 5-PGA, respectively. The stopping criterion is the maximal number of generations, 1000, and the population size is set to 60 for bQIEAo and all PGAs. So 2-PGA, 3-PGA, 4-PGA and 5-PGA have 60 masks and 120, 180, 240 and 300 chromosomes, respectively. The performances are evaluated using Hamming distance and the quality of solutions (Herrera and Lozano 1996). The former includes the mean Hamming distance d_{mh} of all phenotypic individuals and the Hamming distance d_{bwh} between the worst and best phenotypic individuals. The latter consists of the best fitness searched, the best fitness at each generation, the mean fitness at each generation and the worst fitness at each generation. The distances d_{mh} and d_{bwh} are defined as

$$d_{mh} = \frac{2}{n(n-1)} d(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad i, j = 1, 2, \ldots, n \text{ and } i \neq j \tag{22}$$

$$d_{bwh} = d(\boldsymbol{x}_b, \boldsymbol{x}_w) \tag{23}$$

where $d(\cdot)$ is the Hamming distance between two strings, i.e., the number of positions for which the corresponding bits are different; n is the number of individuals in the population, and $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are any two arbitrary selected phenotypic individuals. To be specific, in bQIEAo, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are any two binary individuals in the set $P(t)$ identified in step (ii) of the algorithm, and in PGA, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are any two phenotypic individuals in the population composed of single chromosomes (haploid) instead of polyploid genotypic individuals consisting of multiple sets of chromosomes and a mask. In Eq. 23, $\boldsymbol{x}_b$ and $\boldsymbol{x}_w$ are the best and the worst phenotypic individuals (e.g., in bQIEAo, $\boldsymbol{x}_b, \boldsymbol{x}_w \in P(t)$) in terms of profits, respectively. Figures 17 and 18 provide the statistically experimental results of 30 independent runs. In Fig. 17, Q-optimal, Q-best, Q-mean and Q-worst represent the best fitness found among the past



(a) Comparisons of bQIEAo and 2-PGA

(b) Comparisons of bQIEAo and 3-PGA

(c) Comparisons of bQIEAo and 4-PGA

(d) Comparisons of bQIEAo and 5-PGA

Fig. 17 Comparisons of bQIEAo and PGA using fitness. Q-optimal, Q-best, Q-mean and Q-worst represent the searched best fitness, the best fitness, the mean fitness and the worst fitness at each generation of bQIEAo, respectively. P-optimal, P-best, P-mean and P-worst represent the searched best fitness, the best fitness, the mean fitness and the worst fitness at each generation of PGA, respectively

evolutionary generations, the best fitness, the mean fitness and the worst fitness at each generation of bQIEAo, respectively. Likewise, P-optimal, P-best, P-mean and P-worst represent the best fitness searched among the past evolutionary generations, the best fitness, the mean fitness and the worst fitness at each generation of PGA, respectively. In Fig. 18, Q-mh and Q-bwh represent the mean Hamming distance d_{mh} of all phenotypic individuals and the Hamming distance d_{bwh} between the best and worst phenotypic individuals of bQIEAo, respectively. P-mh and P-bwh represent the mean Hamming distance d_{mh} of all phenotypic individuals and the Hamming distance d_{bwh} between the best and worst phenotypic individuals of PGA, respectively.

It can be seen from Fig. 17 that bQIEAo obtains much better profits than PGA. Figures 17 and 18 show that among 2-PGA, 3-PGA, 4-PGA and 5-PGA, with the same population size, there is little difference with respect to profits and Hamming distances. Similar conclusions were also drawn by Collingwood et al. (1996) who conducted experiments on the ONE MAX problem and showed that there is little difference when the ploidy values vary from 2 to 10, and that the results of PGA are worse than a normal GA. In Fig. 18, at the beginning of evolution, bQIEAo has equiva-

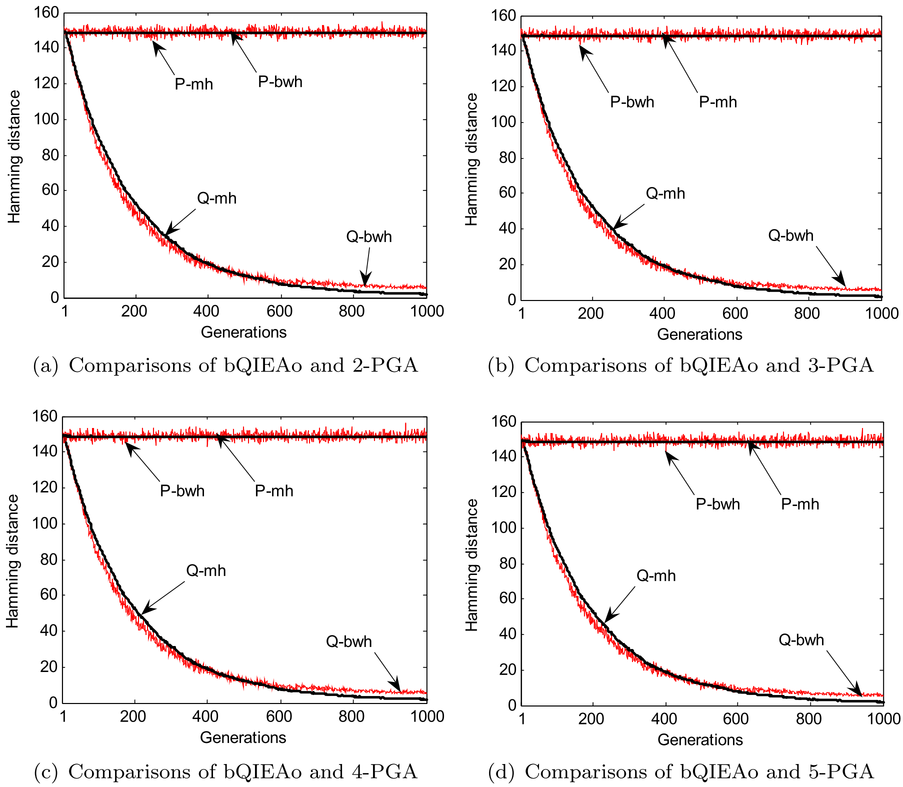


Fig. 18 Comparisons of bQIEAo and PGA using Hamming distance. Q-mh and Q-bwh represent the mean Hamming distance of all individuals and the Hamming distance between the best and worst individuals of bQIEAo, respectively. P-mh and P-bwh represent the mean Hamming distance of all individuals and the Hamming distance between the best and worst individuals of PGA, respectively

lent Hamming distances to PGA, which indicates the Q-bit representation has similar population diversity to PGA. As the generation increases, the Hamming distances of bQIEAo decrease gradually because it converges toward the optimal solution, while the Hamming distances of PGA stay at a steady level instead of converging. In the evolution process implemented by EAs, there is a conflict between population diversity and convergence, whereas bQIEAo attempts to compromise between them and PGA keeps only the former, which produces the various results shown in Fig. 17. It is worth pointing out that the studies in Collingwood et al. (1996) and Corne et al. (1996) shows that a PGA is sometimes better than a normal GA, and sometimes not, and a PGA seems particularly helpful in cases where a normal GA would be likely to irretrievably lose important genetic material.

3.2.2 *Comparisons between rQIEA and other EAs*

In this subsection, experiments are reported for a large number of benchmark numeric optimization problems to compare the performance of rQIEA with that of several

Table 10 Comparisons of rQIEA, SGA-ConDiv-NN, RCMA-XHC and CHC. The results of SGA-ConDiv-NN are referred from Lozano et al. (2008). The results of RCMA-XHC, CHC, CHC-SW-100, CHC-SW-1000 and CHC-SW-500 are referred from Lozano et al. (2004). The number of runs is 50. The bold style highlights the best result for each function

f_{Sph}	f_{Ros}	f_{Sch}	f_{Ras}	f_{Gri}	P_{sle}	P_{fms}	P_{pfp}	
CHC	5.8e−31	1.9e+1	2.0e−2	1.6e+1	6.5e−3	3.9e+1	**1.7e−18**	3.3e+2
CHC-SW-100	2.1e−14	1.8e+1	2.4e+2	4.5e+1	3.4e−3	1.4e+1	5.0e+0	1.5e+2
CHC-SW-1000	9.6e−25	1.5e+1	1.4e+1	6.2e+1	2.0e−2	1.5e+2	1.6e+1	6.6e+2
CHC-SW-5000	8.5e−63	1.5e+1	1.2e−1	9.4e+1	4.4e−2	3.6e+2	2.0e+1	1.4e+3
RCMA-XHC	6.5e−101	**2.2e+0**	3.8e−7	1.4e+0	1.3e−2	5.5e+1	7.7e+0	1.4e+2
SGA-ConDiv-NN	4.3e−50	2.0e+1	5.9e−2	4.4e−1	3.5e−4	3.1e+1	2.9e+2	**1.5e+0**
rQIEA	**1.7e−110**	1.2e+1	**5.0e−17**	**5.3e−15**	**4.0e−12**	**1.2e+1**	1.8e+1	1.58e+2

well-known EAs reported in Auger and Hansen (2005), Herrera et al. (1998, 2003), Eshelman (1991), Lozano et al. (2004, 2008), Deb et al. (2002), Noman and Iba (2008), Price et al. (2005).

Initially, we compare rQIEA with SGA-ConDiv-NN (Lozano et al. 2008), RCMA-XHC (Lozano et al. 2004) and CHC (Eshelman 1991). SGA-ConDiv-NN is a steady-state GA with a replacement strategy and the experiments in Lozano et al. (2008) show that it can maintain high levels of population diversity and obtains higher quality of solutions than nine other algorithms. The CHC algorithm in Eshelman (1991) has become a reference point in the GA literature (Chaiyaratana et al. 2007; Koumousis and Katsaras 2006; Lozano et al. 2008; Noman and Iba 2008; Whitley et al. 1996). According to the experimental comparisons (Lozano et al. 2004), RCMA-XHC and CHC outperforms another 20 real-coded memetic algorithms. We employ the benchmark problems used in Lozano et al. (2005) and Lozano et al. (2004) including five frequently used test functions: the *Sphere* model (f_{Sph}), *Generalized Rosenbrock's* function (f_{Ros}), *Schwefel's problem 1.2* (f_{Sch}), *Generalized Rastringin's* function (f_{Ras}), *Griewangk's* function (f_{Gri}) and three additional real-world problems including *Systems of Linear Equations* (P_{sle}), *Frequency Modulation Sounds Parameter Identification Problem* (P_{fms}) and *Polynomial Fitting Problem* (P_{pfp}). The dimension of the search space is 25 for the first five optimization problems and 10, 6, 9 for P_{sle}, P_{fms} and P_{pfp}, respectively. The optimal solution for each problem is 0. A detailed description of these problems can be found in Lozano et al. (2008) or Lozano et al. (2004). Three combined CHCs utilized in Lozano et al. (2004), i.e., CHC-SW-100, CHC-SW-1000 and CHC-SW-5000, are also compared. In our experiments, rQIEA uses the same number of function evaluations 100000 as the stopping criterion for each problem and the population size is set to 20. Each test problem is performed with 50 independent runs. The mean best values of 50 runs are recorded. Experimental results of the seven algorithms are given in Table 10.

According to the study in Garcia et al. (2009), non-parametric statistical tests are more appropriate than parametric statistical tests in the analysis of EAs' behavior over multiple numeric optimization problems. In this paper, two non-parametric tests, *Wilcoxon's* and *Friedman's* tests, are employed to check whether there are significant

Table 11 The results of *Wilcoxon's* and *Friedman's* tests for the algorithms in Table 10. + and − represent significant difference and no significant difference, respectively

rQIEA vs.	CHC	CHC-SW-100	CHC-SW-1000	CHC-SW-5000	RCMA-XHC	SGA-ConDiv-NN
Wilcoxon (p-value)	0.1094(−)	0.3828(−)	0.0391(+)	0.0078(+)	1.0000(−)	0.1484(−)
Friedman (p-value)	0.0339(+)	0.1573(−)	0.0339(+)	0.0047(+)	0.4795(−)	0.0339(+)

differences for the control algorithm rQIEA. The level of significance considered is 0.05. Table 11 lists the results of *Wilcoxon's* and *Friedman's* tests.

In Table 10, rQIEA achieves better solutions for five (f_{Sph}, f_{Sch}, f_{Ras}, f_{Gri} and P_{sle}) out of eight problems than the other six algorithms. The better results for the other three functions (f_{Ros}, P_{fms} and P_{pfp}) are obtained by RCMA-XHC, CHC and SGA-ConDiv-NN, respectively. According to the *Friedman's* test in Table 11, rQIEA is superior to four other algorithms (CHC, CHC-SW-1000, CHC-SW-5000 and SGA-ConDiv-NN). *Wilcoxon's* tests show that there is a significant difference between the control algorithm rQIEA and CHC-SW-1000 and CHC-SW-5000, but there is not significant difference between the control algorithm rQIEA and the other algorithms. So we can conclude that rQIEA is at least as competitive as the other six algorithms.

Secondly, rQIEA is compared with the multiple GAs with best crossover operators in Herrera et al. (1998, 2003). Herrera et al. (2003) experimentally analyzed real-coded GAs with 18 crossovers and showed the nine crossovers, including BLX-0, BLX-0.3, BLX-0.5, SBX-2, SBX-5, FR, DHX, MMAX and LX, surpass the other ones. We apply the test suite described in Herrera et al. (2003) to conduct our own experiments. The suite consists of 12 optimization problems: *Sphere* model (f_{Sph}), *Schwefel's problem 1.2* (f_{Sch}), *Generalized Rastringin's* function (f_{Ras}), *Griewangk's* function (f_{Gri}), *Expansion of F10* (ef_{10}), *Generalized Rosenbrock's* function (f_{Ros}), *Systems of Linear Equations* (P_{sle}), *Frequency Modulation Sounds Parameter Identification Problem* (P_{fms}), *Polynomial Fitting Problem* (P_{pfp}), *Ackley's* function (f_{Ack}), *Bohachevsky's* function (f_{Boh}) and *Watson's* function (f_{Wat}). The dimension of the search space is 25 for the first four optimization problems, f_{Ros} and f_{Ack}. The other six test functions ef_{10}, P_{sle}, P_{fms}, P_{pfp}, f_{Boh} and f_{Wat} have 10, 10, 6, 9, 2 and 6 dimensions, respectively. In our experiments, the number of function evaluations (100000) in Herrera et al. (2003) is used as the stopping criterion of rQIEA for each problem. The population size of rQIEA is 20 and the independent runs are 30. Experimental results and the results of *Wilcoxon's* and *Friedman's* tests are shown in Tables 12, 13 and 14, respectively. The level of significance considered is 0.05.

The experimental results of 12 optimization problems in Tables 12 and 13 show that rQIEA obtains better performances for half of the functions (f_{Sph}, f_{Sch}, f_{Ras}, f_{Gri}, f_{Ack} and f_{Wat}) than another 9 algorithms, and the optimal result for one function f_{Boh}, which is also obtained by DHX. FR achieves two better results (ef_{10} and P_{fms}) and LX also gets two better results (P_{sle} and P_{pfp}) than the other algorithms. The best result of f_{Ros} is attained by DHX. According to *Friedman's* statistical analysis in Table 14, rQIEA has the advantages over 7 other algorithms (BLX-0, BLX-0.3,

Table 12 Comparisons of rQIEA and the GAs in Herrera et al. (2003). The results of these algorithms are referred from Herrera et al. (2003). Mean and Std represent the mean value of best results of 30 runs and their standard deviation, respectively. The bold style highlights the best result for each function. (To be continued)

Problems		f_{Sph}	f_{Sch}	f_{Ras}	f_{Gri}	ef_{10}	P_{sle}
BLX-0	Mean	1.28e−8	4.00e+1	4.47e+0	1.55e−2	2.15e+0	2.74e+1
	Std	1.09e−8	1.84e+1	2.01e+1	1.82e−2	8.62e−1	1.67e+1
BLX-0.3	Mean	7.51e−11	3.37e+1	7.86e+0	1.54e−2	3.18e−1	2.03e+1
	Std	5.35e−11	1.56e+1	1.80e+0	1.56e−2	1.21e−1	2.16e+1
BLX-0.5	Mean	6.31e−6	1.36e+3	8.72e+1	9.29e+1	1.47e+1	2.62e+1
	Std	8.11e−6	2.60e+2	1.25e+1	2.16e−1	4.54e+0	2.69e+1
SBX-2	Mean	1.97e−9	7.56e+0	1.36e+1	1.91e−2	1.35e+1	3.54e+1
	Std	1.17e−9	4.28e+0	4.56e+0	2.28e−2	8.26e+0	3.82e+1
SBX-5	Mean	2.76e−10	9.54e+1	7.13e+0	2.32e+2	1.99e+1	1.14e+2
	Std	2.08e−10	7.97e+1	2.15e+0	2.51e−2	1.46e+1	8.52e+1
FR	Mean	1.30e−11	8.97e+0	1.96e+1	7.71e−3	**2.45e−1**	2.66e+1
	Std	6.52e−12	7.08e+0	4.84e+0	9.60e−3	**7.29e−2**	1.72e+1
DHX	Mean	1.37e−14	6.04e+1	1.13e−11	9.67e−3	1.31e+0	1.27e+2
	Std	9.63e−15	2.99e+1	1.09e−11	1.32e−2	8.92e−1	5.19e+1
MMAX	Mean	3.17e−11	1.77e+2	9.28e−1	1.31e−2	3.15e+0	1.12e+2
	Std	3.75e−11	8.32e+1	9.23e−1	1.60e−2	2.20e+0	5.95e+1
LX	Mean	3.19e−10	3.86e−1	3.06e+1	2.30e−3	8.89e−1	**2.69e+0**
	Std	1.70e−10	2.66e−1	2.97e+1	5.04e−3	3.14e−1	1.90e+0
rQIEA	Mean	**1.70e−110**	**5.10e−17**	**7.58e−15**	**6.20e−12**	4.82e+0	1.16e+1
	Std	**7.76e−111**	**2.11e−17**	**2.41e−14**	**3.10e−11**	2.08e+0	**1.80e+0**

BLX-0.5, SBX-2, SBX-5, FR and MMAX). *Wilcoxon's* tests show that rQIEA is better than BLX-0, BLX-0.3, BLX-0.5, SBX-2 and SBX-5. So these results indicate that rQIEA is a comparable algorithm with the other real-coded GAs with nine crossovers.

Thirdly, we compare rQIEAs with differential evolution (DE) (Price et al. 2005), a generalized generation gap GA model with a parent-centric crossover (G3 + PCX) proposed by Deb et al. (2002), a DE with a crossover-based adaptive local search strategy (DEahcSPX) (Noman and Iba 2008) and a DE with a crossover hill-climbing strategy (DExhcSPX) introduced by Lozano et al. (2004). A DE is a population-based, stochastic globaloptimizer capable of working reliably in nonlinear and multi-modal environments (Noman and Iba 2008; Price et al. 2005). G3 + PCX was found to perform consistently and reliably perform better than all the other approaches involved in the study in Deb et al. (2002). DEahcSPX was found to perform well for a wide range of benchmark functions (Noman and Iba 2008). The crossover hill-climbing strategy is a self-adaptive crossover local search method with good performances in real-coded memetic algorithms (Lozano et al. 2004). The experiments were performed on the test suite consisting of the first ten functions from the newly defined test suite at CEC 2005 Special Session on real-parameter optimization (Suganthan et al. 2005), $F_1 - F_{10}$, and ten functions commonly used in the literature, *Sphere* function (f_{Sph}), *Rosenbrock's* function (f_{Ros}), *Ackley's* function (f_{Ack}),

Table 13 Comparisons of rQIEA and the GAs in Herrera et al. (2003). The results of these algorithms are referred from Herrera et al. (2003). Mean and Std represent the mean value of best results of 30 runs and their standard deviation, respectively. The bold style highlights the best result for each function. (Continued)

Problems		f_{Ros}	P_{pfp}	P_{fms}	f_{Ack}	f_{Wat}	f_{Boh}
BLX-0	Mean	2.22e+1	2.89e+2	2.05e+1	5.40e−4	1.11e+0	2.29e−11
	Std	5.88e−1	2.16e+2	4.05e+0	2.45e−4	6.04e−3	3.60e−11
BLX-0.3	Mean	2.18e+1	2.19e+2	1.39e+1	3.92e−5	1.11e+0	7.52e−14
	Std	7.35e−1	1.55e+2	6.75e+0	1.52e−5	2.79e−3	1.23e−13
BLX-0.5	Mean	2.61e+1	3.16e+2	1.50e+1	1.02e−2	1.16e+0	7.84e−12
	Std	1.42e+1	2.58e+2	4.56e+0	6.38e−3	3.50e−2	7.70e−12
SBX-2	Mean	2.99e+1	4.18e+2	1.79e+1	2.27e−4	1.37e+0	1.74e−12
	Std	1.98e+1	2.85e+2	4.05e+0	8.51e−5	2.74e−1	2.08e−12
SBX-5	Mean	3.90e+1	8.03e+2	1.08e+1	9.26e−5	1.13e+0	1.91e−13
	Std	2.71e+1	8.99e+2	4.98e+0	4.54e−5	4.73e−2	4.30e−13
FR	Mean	2.54e+1	4.51e+2	**7.30e+0**	1.81e−5	1.11e+0	7.33e−14
	Std	1.53e+1	3.38e+2	6.67e+0	6.44e−6	1.09e−2	6.92e−14
DHX	Mean	**2.17e+0**	7.40e+2	1.64e+1	3.81e−7	1.11e+0	**0.00e+0**
	Std	5.70e−1	4.65e+2	7.91e+0	1.67e−7	3.69e−3	**0.00e+0**
MMAX	Mean	2.67e+1	1.28e+3	1.57e+1	2.87e−5	1.10e+0	4.07e−16
	Std	1.53e+1	9.59e+2	7.62e+0	1.41e−5	**4.62e−5**	1.03e−15
LX	Mean	2.20e+1	**6.35e−1**	2.06e+1	8.29e−5	1.16e+0	4.39e−14
	Std	3.03e−1	**9.03e−1**	**3.31e+0**	2.75e−5	2.65e−2	5.51e−14
rQIEA	Mean	1.22e+1	1.50e+2	1.75e+1	**1.67e−12**	**1.04e−2**	**0.00e+0**
	Std	**2.39e−1**	6.47e+1	1.06e+1	**5.09e−12**	2.68e−3	**0.00e+0**

Table 14 The results of *Wilcoxon's* and *Friedman's* tests for the algorithms in Tables 12 and 13. + and − represent significant difference and no significant difference, respectively

rQIEA vs.	*Wilcoxon* (*p*-value)	*Friedman* (*p*-value)
BLX−0	0.0068 (+)	0.0039 (+)
BLX−0.3	0.0425 (+)	0.0209 (+)
BLX−0.5	0.0094 (+)	0.0039 (+)
SBX−2	0.0005 (+)	0.0005 (+)
SBX−5	0.0049 (+)	0.0039 (+)
FR	0.0522 (−)	0.0209 (+)
DHX	0.3203 (−)	0.1317 (−)
MMAX	0.0640 (−)	0.0209 (+)
LX	0.4697 (−)	0.0833 (−)

Griewangk's function (f_{Gri}), *Rastringin's* function (f_{Ras}), *Schwefel's problem 2.26* (f_{Sch}), *Salomon's* function (f_{Sal}), *Whitely's* function (f_{Wht}), *Generalized Penalized function 1* (f_{pn1}) and *Generalized Penalized function 2* (f_{pn2}), which were described in detail in Noman and Iba (2008). The dimension of the search space is 30 for the 20 optimization problems. rQIEA uses 50 individuals and the number of function

Table 15 Comparisons of rQIEA, DE, DExhcSPX, G3 + PCX and DEahcSPX. The results of the last four algorithms are referred from Noman and Iba (2008). Mean and Std represent the mean value of best results of 50 runs and their standard deviation, respectively. The bold style highlights the best result for each function. (To be continued)

	DE		DExhcSPX		G3 + PCX	
	Mean	Std	Mean	Std	Mean	Std
f_{Sph}	5.73e−17	2.03e−16	7.66e−29	1.97e−28	3.58e−81	1.36e−81
f_{Ros}	5.20e+1	8.56e+1	5.81e+0	4.73e+0	**4.18e+0**	9.68e+1
f_{Ack}	1.37e−9	1.32e−9	5.22e−15	2.62e−15	1.48e+1	4.17e+0
f_{Grw}	2.66e−3	5.73e−3	3.45e−3	7.52e−3	1.07e−2	1.30e−2
f_{Ras}	2.55e+1	8.14e+0	1.86e+1	7.05e+0	1.75e+2	3.37e+1
f_{Sch}	4.90e+2	**2.34e+2**	4.91e+2	4.06e+2	4.04e+3	1.09e+3
f_{Sal}	2.52e−1	4.78e−2	1.92e−1	4.93e−2	4.64e+0	4.74e+0
f_{Wht}	3.10e+2	**1.07e+2**	**2.84e+2**	1.10e+2	7.90e+2	1.27e+2
f_{pn1}	4.56e−2	1.31e−1	2.49e−2	8.61e−2	4.35e+0	6.94e+0
f_{pn2}	1.44e−1	7.19e−1	4.39e−4	2.20e−3	1.50e+1	1.58e+1
F_1	3.87e−14	2.71e−14	**0.00e+0**	**0.00e+0**	3.52e−13	1.22e−13
F_2	8.50e−2	7.94e−2	9.40e−4	1.80e−3	4.14e−12	1.21e−12
F_3	3.63e+6	2.06e+6	1.54e+6	1.15e+6	**1.07e+3**	**1.29e+3**
F_4	5.54e+1	6.37e+1	6.69e+0	1.06e+1	9.35e+4	2.66e+4
F_5	1.08e+3	5.31e+2	1.01e+3	**4.31e+2**	8.13e+3	2.65e+3
F_6	6.67e+1	1.51e+2	1.41e+1	1.86e+1	1.34e+2	2.48e+2
F_7	7.59e−3	8.96e−3	7.98e−3	9.48e−3	2.01e−2	1.85e−2
F_8	2.09e+1	1.33e−1	2.09e+1	7.41e−2	2.11e+1	6.67e−12
F_9	2.43e+1	**6.23e+0**	2.80e+1	7.75e+0	2.44e+2	3.98e+1
F_{10}	7.33e+1	6.62e+1	6.79e+1	4.80e+1	3.89e+2	9.96e+1

evaluations 300000 employed in DE, G3 + PCX, DEahcSPX and DExhcSPX as the stopping criterion. The statistical results for 50 independent runs are shown in Tables 15 and 16. The results of *Wilcoxon's* and *Friedman's* tests for these algorithms are listed in Table 17. The level of significance considered is 0.05.

In Table 17, the *Friedman's* and *Wilcoxon's* tests show that rQIEA surpasses only one algorithm G3 + PCX among four algorithms, but Table 15 and Table 16 shows that rQIEA achieves a little bit better results than the other four algorithms because rQIEA, DExhcSPX, G3 + PCX and DEahcSPX obtain the best results for 9, as opposed to 2, 2 and 8, functions, respectively, in terms of the mean best values.

Finally, we draw an experimental comparison between rQIEA and the algorithm G-CMA-ES in Auger and Hansen (2005). G-CMA-ES is a restart covariance matrix adaptation evolution strategy with increasing population size. It was the winner of the real-parameter optimization competition, organized in the 2005 IEEE congress on evolutionary computation (Garcia et al. 2009). The test suite is composed of 25 numeric optimization problems with 10 dimensions, $F_1 - F_{25}$, defined for the CEC 2005 Special Session on real-parameter optimization (Suganthan et al. 2005). In the experiments of rQIEA, the population size is set to 20 and the stopping criterion

Table 16 Comparisons of rQIEA, DE, DExhcSPX, G3 + PCX and DEahcSPX. The results of the last four algorithms are referred from Noman and Iba (2008). Mean and Std represent the mean value of best results of 50 runs and their standard deviation, respectively. The bold style highlights the best result for each function. (Continued)

	DEahcSPX		rQIEA	
	Mean	Std	Mean	Std
f_{Sph}	1.75e−31	4.99e−31	**1.06e−85**	**1.00e−85**
f_{Ros}	4.52e+0	1.55e+1	1.15e+1	**1.56e+0**
f_{Ack}	**2.66e−15**	**0.00e+0**	1.96e−14	4.35e−14
f_{Grw}	2.07e−3	5.89e−3	**1.89e−15**	**9.17e−16**
f_{Ras}	2.14e+1	1.23e+1	**8.14e−15**	**1.95e−14**
f_{Sch}	**4.70e+2**	2.96e+2	3.95e+3	9.69e+2
f_{Sal}	**1.80e−1**	**4.08e−2**	3.32e−1	7.12e−2
f_{Wht}	3.06e+2	1.10e+2	6.39e+2	1.58e+2
f_{pn1}	2.07e−2	8.46e−2	**2.60e−32**	**5.06e−24**
f_{pn2}	1.71e−31	5.35e−31	**1.35e−32**	**3.30e−33**
F_1	**0.00e+0**	**0.00e+0**	8.19e−14	2.85e−14
F_2	6.52e−5	4.84e−5	**6.22e−13**	**1.60e−13**
F_3	1.29e+6	9.22e+5	3.92e+5	1.00e+5
F_4	4.62e+0	8.78e+0	**1.57e+0**	**3.59e+0**
F_5	**9.00e+2**	4.79e+2	3.60e+3	9.42e+2
F_6	**3.84e+0**	**3.75e+0**	9.13e+1	8.19e+1
F_7	7.39e−3	6.32e−3	**6.61e−3**	**4.44e−3**
F_8	2.09e+1	1.12e−1	**2.01e+1**	**6.31e−2**
F_9	**2.04e+1**	8.19e+0	44.6e+1	1.11e+1
F_{10}	**5.27e+1**	4.84e+1	2.42e+2	**3.93e+1**

Table 17 The results of *Wilcoxon's* and *Friedman's* tests for the algorithms in Tables 15 and 16. + and − represent significant difference and no significant difference, respectively

rQIEA vs.	DE	DExhcSPX	G3 + PCX	DEahcSPX
Wilcoxon (p-value)	1.0000 (−)	0.4552 (−)	0.0276 (+)	0.4115 (−)
Friedman (p-value)	0.3711 (−)	1.0000 (−)	0.0017 (+)	1.0000 (−)

applies the number of function evaluations (100000) in Auger and Hansen (2005). The statistical results of 25 runs are given in Table 18. The results of *Wilcoxon's* and *Friedman's* tests for rQIEA and G-CMA-ES are 0.5605 and 0.8348, respectively. Compared with G-CMA-ES, Table 18 shows that rQIEA achieves better results for 12 functions (F_1, F_2, F_4, F_5, F_9, $F_{12} - F_{15}$, F_{21}, F_{22} and F_{25}) and the same results of two functions (F_8 and F_{24}), but is outperformed by G-CMA-ES for the other 11 functions (F_3, F_6, F_7, F_{10}, F_{11}, $F_{16} - F_{20}$ and F_{23}), in terms of mean best values. Examined relative to the level of significance 0.05, there is no significant difference between them. Therefore, what we can say is that rQIEA is not worse than G-CMA-ES.

Table 18 Comparisons of rQIEA and G-CMA-ES. The results of G-CMA-ES are referred from Garcia et al. (2009). Mean and Std represent the mean value of best results of 25 runs and their standard deviation, respectively. The bold style highlights the best result for each function with 10 dimensions

Problems	G-CMS-ES		rQIEA	
	Mean	Std	Mean	Std
F_1	5.20e−9	1.94e−9	**1.71e−9**	**1.67e−9**
F_2	4.70e−9	1.56e−9	**3.82e−9**	**1.34e−9**
F_3	**5.60e−9**	**1.93e−9**	3.92e+4	1.78e+4
F_4	5.02e−9	1.71e−9	**4.62e−9**	**1.53e−9**
F_5	6.58e−9	2.17e−9	**1.62e−9**	**2.02e−9**
F_6	**4.87e−9**	**1.66e−9**	2.88e+0	1.80e+0
F_7	**3.31e−9**	**2.02e−9**	1.90e−1	6.67e−2
F_8	**2.00e+1**	**3.89e−3**	**2.00e+1**	4.48e−2
F_9	2.39e−1	4.34e−1	**2.14e−1**	**1.02e−1**
F_{10}	**7.96e−2**	**2.75e−1**	1.74e+1	7.41e+0
F_{11}	**9.34e−1**	**9.00e−1**	6.17e+0	1.26e+0
F_{12}	2.93e+1	1.42e+2	**1.54e+1**	**4.82e+0**
F_{13}	6.96e−1	**1.50e−1**	**6.81e−1**	2.29e−1
F_{14}	3.01e+0	3.49e−1	**2.91e+0**	**2.05e−1**
F_{15}	2.28e+2	6.80e+1	**8.92e+1**	**6.81e+0**
F_{16}	**9.13e+1**	**3.49e+0**	1.32e+2	3.49e+1
F_{17}	**1.23e+2**	**2.09e+1**	1.79e+2	5.04e+1
F_{18}	**3.32e+2**	1.12e+2	4.51e+2	**5.22e+1**
F_{19}	**3.26e+2**	9.93e+1	4.40e+2	**5.83e+1**
F_{20}	**3.00e+2**	**0.00e+0**	4.38e+2	5.97e+1
F_{21}	5.00e+2	**3.48e−13**	**4.28e+2**	9.80e+1
F_{22}	7.29e+2	**6.86e+0**	**4.42e+2**	2.45e+2
F_{23}	**5.59e+2**	**3.24e−11**	7.44e+2	6.70e+1
F_{24}	**2.00e+2**	2.29e−6	**2.00e+2**	**1.14e−7**
F_{25}	3.74e+2	**3.22e+0**	**3.62e+2**	8.59e+0

4 Conclusions and future research paths

The interaction of QIEAs and EAs generates three branches: EDQA, QEA and QIEA. In this paper we have presented a systematic review of recent efforts to develop a theory of QIEAs. After giving a brief introduction to the algorithms and problems considered in this overview, we discussed the Q-bit representation and the basic structure of QIEAs, and reviewed binary observation QIEA, real observation QIEA and QIEA-like algorithms. Regarding bQIEA, we have summarized the algorithms used and results obtained with respect to combinatorial optimization problems and numerical optimization problems; some hybrid algorithms of bQIEA and specific optimization methods were also presented. Finally, we conducted a small number of experiments to compare the performances of several QIEA variants and have drawn comparisons between QIEAs and some state-of-the-art EAs using frequently used benchmark problems. Currently there is intensive research in this area, but there are some aspects that need to be addressed. For example, the following issues deserve special attention:

– *Theoretical research*. Despite many developments in the current literature concerning experiments and applications, very few studies regarding theoretical aspects of QIEAs have been presented. For instance, how does a QIEA work when searching for the optimal solution? To what extent and how can a QIEA escape minima? Although there are discussions regarding the convergence of QIEAs, there is no systematic analysis of the advantages and disadvantages of the approach. Moreover, further work is needed on the application of other concepts and principles from quantum computing, such as quantum registers, entanglement, and interference, to EAs to solve more complex optimization problems including those with dependent variables. For instance, controlled quantum-inspired gates, such as controlled NOT gates and controlled rotation gates, could be used to solve the problems which depend on two or more Q-bits. A controlled NOT gate suitable for dealing with interactions between two Q-bits can be defined as in Eq. 24 or Eq. 25 (DiVincenzo 1998; Barenco et al. 1995).

$$G_{NOT1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{24}$$

$$G_{NOT2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

In the controlled NOT gates, one of the two Q-bits is taken to be a control bit. If the control bit is 1, the NOT operation is applied to the other Q-bit; otherwise the second Q-bit is left unchanged. Thus, it can be used to process dependent variables in optimization problems. Additionally, a controlled rotation gate might be defined (DiVincenzo 1998; Barenco et al. 1995) by

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{26}$$

It is worth noting that control bits may be multiple and that controlled gates can be applied to deal with dependencies between multiple Q-bits.

– *Engineering applications*. The research conducted so far presents the QIEA as an effective EA with a lot of promising features and many potential applications. QIEAs have successfully been used to test some important combinatorial optimization problems, such as the knapsack problem and some benchmark optimization functions; they have also been employed to solve some engineering optimizations such as digital filter design and image processing. However, the potential of QIEAs has hardly been explored for engineering applications, compared with other optimization methods such as particle swarm optimization. In particular, applications research for the rQIEA is still at a preliminary stage, although QIEAs may feasibly be modified to satisfy specific engineering application requirements.

– *Comparative experiments*. Most of the experiments that have been published were conducted to compare QIEAs with CGAs. There are few or no convincing comparisons between QIEAs and other optimization methods such as particle swarm optimization, ant colony optimization, evolutionary programming, evolutionary strategy and immune algorithm. The advantages and disadvantages of QIEAs over other optimization methods are still pending issues.
– *Extensions of QIEAs*. Except for solving single-objective optimization problems and unconstrained problems, QIEAs can be extended to other fields such as multi-objective optimization and constraint-handling techniques. There are many such problems in real-world applications and QIEAs may be relevant to solving these problems.
– *Hybrid algorithms*. Some research approaches have concentrated on combining QIEAs with CGAs, immune algorithms, clonal algorithms and particle swarm optimization, but further theoretical and experimental analysis is needed to provide an easy and clear description of the combination mechanism.

Acknowledgements The author is very grateful to the anonymous Associate Editor, the anonymous reviewers, Dr. M. Gheorghe, Dr. N. Krasnogor and Dr. M. Stannett for their insightful recommendations, suggestions and comments to improve the quality of this paper.

References

Abdesslem, L., Soham, M., Mohamed, B.: Multiple sequence alignment by quantum genetic algorithm. In: Proc. IPDPS, pp. 360–367 (2006)

Abs da Cruz, A., Hall Barbosa, C., Pacheco, M., Vellasco, M.: Quantum-inspired evolutionary algorithms and its application to numerical optimization problems. Lect. Not. Comput. Sci. **3316**, 212–217 (2004)

Abs da Cruz, A., Pacheco, M., Vellasco, M., Barbosa, C.: Cultural operators for a quantum-inspired evolutionary algorithm applied to numerical optimization problems. Lect. Not. Comput. Sci. **3562**, 1–10 (2005)

Abs da Cruz, A., Vellasco, M., Pacheco, M.: Quantum-inspired evolutionary algorithm for numerical optimization. In: Proc. CEC, pp. 2630–2637 (2006)

Abs da Cruz, A., Vellasco, M., Pacheco, M.: Quantum-inspired evolutionary algorithm for numerical optimization. In: Studies in Computational Intelligence, vol. 75, pp. 19–37 (2007)

Akbarzadeh-T, M.: Evolutionary quantum algorithms for structural design. In: Proc. IEEE SMC vol. 4, pp. 3077–3082 (2005)

Al-Othman, A., Al-Fares, F., EL-Nagger, K.: Power system security constrained economic dispatch using real coded quantum inspired evolution algorithm. Int. J. Electr. Comput. Syst. Eng **1**(4), 199–206 (2007)

Alfares, F., Esat, I.: Real-coded quantum inspired evolution algorithm applied to engineering optimization problems. In: Proc. ISoLA, pp. 169–176 (2006)

Alfares, F., Alfares, M., Esat, I.: Quantum-inspired evolution algorithm: experimental analysis. In: Proc. ACDM, pp. 377–389 (2004)

Araujo, M., Nedjah, N., Mourelle, L.: Quantum-inspired evolutionary state assignment for synchronous finite state machines. J. Univers. Comput. Sci. **14**(15), 2532–2548 (2008)

Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: Proc. CEC, pp. 1769–1776 (2005)

Bäck, T., Hammel, U., Schwefel, H.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evol. Comput. **1**(2), 3–17 (1997)

Baluja, S.: Population based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report No. CMU-CS-94-163. Carnegie Mellon University, Pittsburgh, Pennsylvania (1994)
Baluja, S., Davies, S.: Using optimal dependency trees for combinatorial optimization: Learning the structure of search space. Technical Report CMU-CS-97-107. Carnegie Mellon University, Pittsburgh, Pennsylvania (1997)
Barenco, A., Bennett, C., Cleve, R., Divincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**(5), 3457–3467 (1995)
Bennett, C., DiVincenzo, D.: Quantum information and computation. Nature **404**, 247–255 (2000)
Bi, X., Jin, G.: Image segmentation algorithm based on quantum immune programming. In: Proc IEEE ICIT, pp. 403–407 (2007)
Box, G.: Evolutionary operation: a method for increasing industrial productivity. Appl. Stat. **6**, 81–101 (1957)
Bremermann, H.: Optimization through evolution and recombination. In: Yovits MC (ed) Self-Organizing Systems, Spartan, Washington DC (1962)
Burian, R.: Underappreciated pathways toward molecular genetics as illustrated by Jean Brachet's cytochemical embryology. In: Sarkar, S. (ed.) The Philosophy and History of Molecular Biology: New Perspectives, pp. 67–85. Kluwer, Dordrecht (1996)
Chaiyaratana, N., Piroonratana, T., Sangkawelert, N.: Effects of diversity control in single objective and multi-objective genetic algorithms. J. Heuristics **13**(1), 1–34 (2007)
Chen, H., Zhang, J., Zhang, C.: Chaos updating rotated gates quantum-inspired genetic algorithm. In: Proc. ICCCAS, pp. 1108–1112 (2004)
Collingwood, E., Corne, D., Ross, P.: Useful diversity via multiploidy. In: Proc. CEC, pp. 810–813 (1996)
Corne, D., Collingwood, E., Ross, P.: Investigating multiploidy's niche. Lect. Not. Comput. Sci. **1143**, 189–198 (1996)
Darwin, C.: On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. Murray, London (1859)
De Bonet, J., Isbell, C., Viola, P.: Mimic: Finding optima by estimating probability densities. In: NIPS. MIT Press, Cambridge (1997)
Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evol. Comput. **10**, 371–395 (2002)
Ding, S., Jin, Z., Yang, Q.: Evolving quantum circuits at the gate level with a hybrid quantum-inspired evolutionary algorithm. Soft Comput. **12**(11), 1059–1072 (2008)
DiVincenzo, D.: Quantum gates and circuits. Proc. R. Soc. A, Math. Phys. Eng. Sci. **454**, 261–276 (1998)
Du, J., Tian, Y., Zuo, M., Zhou, Y.: Using quantum immune clone algorithm in the prediction of tourism emergency events. In: Proc. ICCAS, pp. 2519–2522 (2007)
Eddy, S.: Infernal: inference of RNA alignments. http://www.fli-leibniz.de/RNA.html, the RNA World Website (2009)
Eshelman, L.: The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Rawlin, S.M.G.J.E. (ed.) Foundations of Genetic Algorithms, pp. 265–283. Morgan Kaufmann, San Mateo (1991)
Fan, K., Brabazon, A., O'Sullivan, C., O'Neill, M.: Option pricing model calibration using a real-valued quantum-inspired evolutionary algorithm. In: Proc. GECCO, pp. 1983–1990 (2007)
Feng, X., Wang, Y., Ge, H., Zhou, C., Liang, Y.: Quantum-inspired evolutionary algorithm for travelling salesman problem. Comput. Methods, 1363–1367 (2006)
Fogel, L., Owens, A., Walsh, M.: Artificial Intelligence Through Simulated Evolution. Wiley, New York (1966)
Fraser, A.: Simulation of genetic systems by automatic digital computers. Aust. J. Biol. Sci. **10**, 484–491 (1957)
Friedberg, R.: A learning machine: Part i. IBM J. Res. Dev. **2**, 2–13 (1958)
Friedberg, R., Dunham, B., North, J.: A learning machine: Part ii. IBM J. Res. Dev. **3**, 282–287 (1959)
Ganesh, V., Singhal, G.: Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training MLP and SRN neural networks. J. Comput. Theor. Nanosci. **2**(4), 561–568 (2005)
Gao, H., Xu, G., Wang, Z.: A novel quantum evolutionary algorithm and its application. In: Proc. WCICA, pp. 3638–3642 (2006)
Garcia, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J. Heuristics **15**(6), 617–644 (2009)

Gardner, P., Wilm, A., Washietl, S.: A benchmark of multiple sequence alignment programs upon structural RNAs. Nucleic Acids Res. **33**, 2433–2439 (2005)
Glassner, A.: Quantum computing, part 2. IEEE Comput. Graph. Appl. 86–95 (2001a)
Glassner, A.: Quantum computing, part 3. IEEE Comput. Graph. Appl. 73–82 (2001b)
Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley/Longman, Boston (1989)
Grigorenko, I., Garcia, M.: Ground-state wave functions of two-particle systems determined using quantum genetic algorithms. Physica A, Stat. Mech. Its Appl. **291**(1–4), 439–448 (2001)
Grigorenko, I., Garcia, M.: Calculation of the partition function using quantum genetic algorithms. Physica A, Stat. Mech. Its Appl. **313**(3–4), 463–470 (2002)
Grover, L.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**(2), 325–328 (1997)
Grover, L.: Quantum computation. In: Proc. VLSI Design, pp. 548–553 (1999)
Gu, J., Gu, M., Cao, C., Gu, X.: A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. Comput. Oper. Res. **37**(5), 927–937 (2009a)
Gu, J., Gu, X., Gu, M.: A novel parallel quantum genetic algorithm for stochastic job shop scheduling. J. Math. Anal. Appl. **355**(1), 63–81 (2009b)
Guo, R., Li, B., Zou, Y., Zhuang, Z.: Hybrid quantum probabilistic coding genetic algorithm for large scale hardware-software co-synthesis of embedded systems. In: Proc. CEC, pp. 3454–3458 (2007)
Han, K., Kim, J.: Genetic quantum algorithm and its application to combinatorial optimization problem. In: Proc. CEC, vol. 2, pp. 1354–1360 (2000)
Han, K., Kim, J.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. **6**(6), 580–593 (2002)
Han, K., Kim, J.: On setting the parameters of QEA for practical applications: Some guidelines based on empirical evidence. Lect. Not. Comput. Sci. **2723**, 427–428 (2003a)
Han, K., Kim, J.: On setting the parameters of quantum-inspired evolutionary algorithm for practical application. In: Proc. CEC, pp. 178–184 (2003b)
Han, K., Kim, J.: Quantum-inspired evolutionary algorithms with a new termination criterion, h-epsilon gate, and two-phase scheme. IEEE Trans. Evol. Comput. **8**(2), 156–169 (2004)
Han, K., Kim, J.: On the analysis of the quantum-inspired evolutionary algorithm with a single individual. In: Proc. CEC, pp. 2622–2629 (2006)
Han, K., Park, K., Lee, C., Kim, J.: Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In: Proc. CEC, vol. 2, pp. 1422–1429 (2001)
Harik, G.: Linkage learning via probabilistic modeling in the ECGA. Tech. Rep., Illinois Genetic Algorithm Laboratory, University of Illinois, Urbana, Illinois (1999)
Harik, G.R., Lobo, F.G., Goldberg, DE: The compact genetic algorithm. In: Proc. EC, pp. 523–528 (1998)
Herrera, F., Lozano, M.: Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In: Genetic Algorithms and Soft Comput., pp. 95–125. Physica-Verlag, Heidelberg (1996)
Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis. Artif. Intell. Rev. **12**(4), 265–319 (1998)
Herrera, F., Lozano, M., Sanchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. Int. J. Intell. Syst. **18**(3), 309–338 (2003)
Hey, T.: Quantum computing: an introduction. Comput. Control Eng. J. **10**(3), 105–112 (1999)
Hinterding, R.: Representation, constraint satisfaction and the knapsack problem. In: Proc. CEC, pp. 1286–1292 (1999)
Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
Huang, Y., Tang, C., Wang, S.: Quantum-inspired swarm evolution algorithm. In: Proc. CISW, pp. 208–211 (2007)
Huo, H., Stojkovic, V.: Two-phase quantum based evolutionary algorithm for multiple sequence alignments. In: Proc. ICCIAS, pp. 374–379 (2006)
Huo, H., Stojkovic, V.: Two-phase quantum based evolutionary algorithm for multiple sequence alignment. In: Lecture Notes in Artificial Intelligence, vol. 4456, pp. 11–21 (2007)
Imabeppu, T., Nakayama, S., Ono, S.: A study on a quantum-inspired evolutionary algorithm based on pair swap. Artif. Life Robot. **12**(1), 148–152 (2008)
Jang, J.S., Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm-based face verification. In: Lecture Notes in Computer Science, vol. 2724, pp. 2147–2156 (2003)
Jang, J.S., Han, K.H., Kim, J.H.: Evolutionary algorithm-based face verification. Pattern Recognit. Lett. **25**(16), 1857–1865 (2004a)

Jang, J.S., Han, K.H., Kim, J.H.: Face detection using quantum-inspired evolutionary algorithm. In: Proc. CEC, pp. 2100–2106 (2004b)
Jang, S.H., Jung, Y.W., Kim, W., Shin, J.R., Park, J.B.: A thermal unit commitment approach based on a bounded quantum evolutionary algorithm. Trans. Korean Inst. Electr. Eng. **58**(6), 1057–1064 (2009)
Jeong, Y.W., Park, J.B., Shin, J.R., Lee, K.Y.: A thermal unit commitment approach using an improved quantum evolutionary algorithm. Electr. Power Compon. Syst. **37**(7), 770–786 (2009)
Jiao, L., Li, Y.: Quantum-inspired immune clonal optimization. In: Proc. ICNN&B, pp. 461–468 (2005)
Jiao, L., Li, Y., Gong, M., Zhang, X.: Quantum-inspired immune clonal algorithm for global optimization. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **38**(5), 1234–1253 (2008)
John, V., John, P.: Reactive power and voltage control based on general quantum genetic algorithms. Expert Syst. Appl. **36**(3), 6118–6126 (2009)
Kent, A., Williams, J.G.: Encyclopedia of Computer Science and Technology. CRC Press, Boca Raton (1999)
Khorsand, A.: Quantum gate optimization in a meta-level genetic quantum algorithm. In: Proc. IEEE SMC, pp. 3055–3062 (2005)
Khorsand, A.: Genetic quantum algorithm for voltage and pattern design of piezoelectric actuator. In: Proc. CEC, pp. 2593–2600 (2006)
Kim, K.H., Hwang, J.Y., Han, K.H., Kim, J.H., Park, K.H.: A quantum-inspired evolutionary computing algorithm for disk allocation method. IEICE Trans. Inf. Syst. **E86D**(3), 645–649 (2003)
Kim, Y., Kim, J.H., Han, K.H.: Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems. In: Proc. CEC, pp. 2601–2606 (2006)
Koumousis, V.K., Katsaras, C.P.: A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. IEEE Trans. Evol. Comput. **10**(1), 19–27 (2006)
Koza, J.R., Al-Sakran, S.H., Jones, L.W.: Cross-domain features of runs of genetic programming used to evolve designs for analog circuits, optical lens systems, controllers, antennas, mechanical systems, and quantum computing circuits. In: Proc NASA/DoD EH, pp. 205–212 (2005)
Larrañaga, P., Etxeberria, R., Lozano, J.A., Peña, J.M.: Combinatorial optimization by learning and simulation of bayesian networks. In: Proc. UAI, pp. 343–352 (2000)
Lau, T.W., Chung, C.Y., Wong, K.P., Chung, T.S., Ho, S.L.: Quantum-inspired evolutionary algorithm approach for unit commitment. IEEE Trans. Power Syst. **24**(3), 1503–1512 (2009)
Li, B., Zhuang, Z.: Genetic algorithm based-on the quantum probability representation. In: Lecture Notes in Computer Science, vol. 2412, pp. 79–95 (2002)
Li, B.B., Wang, L.: A hybrid quantum-inspired genetic algorithm for multi-objective scheduling. In: Lecture Notes in Computer Science, vol. 4113, pp. 511–522 (2006)
Li, B.B., Wang, L.: A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **37**(3), 576–591 (2007)
Li, N., Du, P., Zhao, H.J.: Independent component analysis based on improved quantum genetic algorithm: Application in hyperspectral images. In: Proc. IGARSS, pp. 4323–4326 (2005a)
Li, P., Li, S.: Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits. Neurocomputing **72**(1–3), 581–591 (2008)
Li, Y., Jiao, L.: Quantum-inspired immune clonal algorithm. In: Lecture Notes in Computer Science, vol. 3627, pp. 304–317 (2005)
Li, Y., Jiao, L.: Quantum-inspired immune clonal multiobjective optimization algorithm. In: Lecture Notes in Artificial Intelligence, vol. 4426, pp. 672–679 (2007)
Li, Y., Liu, F.: A novel immune clonal algorithm. In: Lecture Notes in Computer Science, vol. 4222, pp. 31–40 (2006)
Li, Y., Jiao, L., Liu, F.: Self-adaptive chaos quantum clonal evolutionary programming. In: Proc. ICSP, vol. 2, pp. 1550–1553 (2004a)
Li, Y., Zhang, Y.N., Zhao, R.C., Jiao, L.C.: An edge detector based on parallel quantum-inspired evolutionary algorithm. In: Proc. ICMLC, pp. 4062–4066
Li, Y., Zhang, Y.N., Zhao, R.C., Jiao, L.C.: The immune quantum-inspired evolutionary algorithm. In: Proc. IEEE ICSMC, pp. 3301–3305 (2004c)
Li, Y., Zhang, Y., Cheng, Y., Jiang, X., Zhao, R.: A novel immune quantum-inspired genetic algorithm. In: Lecture Notes in Computer Science, vol. 3612, pp. 215–218 (2005b)
Li, Y., Jiao, L., Gou, S.: Quantum-inspired immune clonal algorithm for multiuser detection in DS-CDMA systems. In: Lecture Notes in Computer Science, vol. 4247, pp. 80–87 (2006)
Li, Y.Y., Jiao, L.C.: Quantum-inspired immune clonal algorithm and its application. In: Proc. ISPACS, pp. 670–673 (2008)

Li, Z., Rudolph, G., Li, K.: Convergence performance comparison of quantum-inspired multi-objective evolutionary algorithms. Comput. Math. Appl. **57**(11–12), 1843–1854 (2009)
Liu, F., Li, S.Q., Liang, M., Hu, L.Z.: Wideband signal DOA estimation based on modified quantum genetic algorithm. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E89A**(3), 648–653 (2006)
Liu, H., Zhang, D., Yan, J.Q., Li, Z.S.: Fast and robust portrait segmentation using QEA and histogram peak distribution methods. In: Lecture Notes in Computer Science, vol. 3645, pp. 920–928 (2005)
Liu, H., Zhang, G., Liu, C., Fang, C.: A novel memetic algorithm based on real-observation quantum-inspired evolutionary algorithms. In: Proc. ISKE, pp. 486–490 (2008)
Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-coded memetic algorithms with crossover hill-climbing. Evol. Comput. **12**(3), 273–302 (2004)
Lozano, M., Herrera, F., Cano, J.R.: Replacement strategies to maintain useful diversity in steady-state genetic algorithms. In: Soft Computing: Methodology and Applications. Springer, Berlin (2005)
Lozano, M., Herrera, F., Cano, J.R.: Replacement strategies to preserve useful diversity in steady-state genetic algorithms. Inf. Sci. **178**(23), 4421–4433 (2008)
Lu, T.C., Juang, J.C., Yu, G.R.: On-line outliers detection by neural network with quantum evolutionary algorithm. In: Proc ICICIC, pp. 254–257 (2008)
Luo, Z., Wang, P., Li, Y., Zhang, W., Tang, W., Xiang, M.: Quantum-inspired evolutionary tuning of SVM parameters. Progr. Nat. Sci. **18**(4), 475–480 (2008)
Lv, Y.J., Liu, N.X.: Application of quantum genetic algorithm on finding minimal reduct. In: Proc. GrC, pp. 728–733 (2007)
Malossini, A., Blanzieri, E., Calarco, T.: QGA: a quantum genetic algorithm. Technical Report No. DIT-04-105, Informatica e Telecommunicazioni, University of Trento (2004)
Malossini, A., Blanzieri, E., Calarco, T.: Quantum genetic optimization. IEEE Trans. Evol. Comput. **12**(2), 231–241 (2008)
Martinez, A., Benavente, R.: The AR face database. http://rvl1.ecn.purdue.edu/~aleix/aleixfaceDB.html (1998)
Meshoul, S., Layeb, A., Batouche, M.: A quantum evolutionary algorithm for effective multiple sequence alignment. In: Lecture Notes in Artificial Intelligence, vol. 3808, pp. 260–271 (2005a)
Meshoul, S., Mahdi, K., Batouche, M.: A quantum inspired evolutionary framework for multi-objective optimization. In: Lecture Notes in Artificial Intelligence, vol. 3808, pp. 190–201 (2005b)
Moore, M., Narayanan, A.: Quantum-inspired computing. Technical Report, Department of Computer Science, University Exeter, Exeter, UK (1995)
Mühlenbein, H., Mahnig, T.: The equation for response to selection and its use for prediction. Evol. Comput. **5**(3), 303–346 (1998)
Mühlenbein, H., Mahnig, T.: The factorized distribution algorithm for additively decomposed functions. In: Proc. CEC, pp. 752–759 (1999)
Narayanan, A.: Quantum computing for beginners. In: Proc. CEC, pp. 2231–2238 (1999)
Narayanan, A., Moore, M.: Quantum-inspired genetic algorithms. In: Proc. CEC, pp. 61–66 (1996)
Nielsen, A.M., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
Niu, Q., Zhou, T., Ma, S.: A quantum-inspired immune algorithm for hybrid flow shop with makespan criterion. J. Univers. Comput. Sci. **15**(4), 765–785 (2009)
Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. IEEE Trans. Evol. Comput. **12**(1), 107–125 (2008)
Notredame, C., Holm, L., Higgins, D.: Coffee: an objective functions for multiple sequence alignments. Bioinformatics **14**, 407–422 (1998)
Pan, G.F., Xia, K.W., Dong, Y., Shi, J.: An improved LS-SVM based on quantum PSO algorithm and its application. In: Proc. ICNC, pp. 606–610 (2007)
Pelikan, M., Mühlenbein, H.: The bivariate marginal distribution algorithm. In: Advances in Soft Computing—Engineering Design and Manufacturing, pp. 521–535 (1999)
Pelikan, M., Goldberg, D., Cantú-paz, E.: Linkage problem, distribution estimation and bayesian networks. Evol. Comput. **8**(3), 311–340 (2000)
Pelikan, M., Goldberg, D., Lobo, F.G.: A survey of optimization by building and using probabilistic models. Comput. Optim. Appl. **21**, 5–20 (2002)
Platel, M., Schliebs, S., Kasabov, N.: Quantum-inspired evolutionary algorithm: A multimodel EDA. IEEE Trans. Evol. Comput. **13**(6), 1218–1232 (2009)
Platelt, M.D., Schliebs, S., Kasabov, N.: A versatile quantum-inspired evolutionary algorithm. In: Proc. CEC, pp. 423–430 (2007)

Pötz, W., Fabian, J. (eds.) Quantum Coherence: from Quarks to Solids. Springer, Berlin (2006)
Price, K., Storn, R.M., Lampinen, JA: Differential Evolution: A Practical Approach to Global Optimization. Springer, Berlin (2005)
Qin, C., Zheng, J., Lai, J.: A multiagent quantum evolutionary algorithm for global numerical optimization. In: LNBI, vol. 4689, pp. 380–389 (2007)
Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systemenach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart (1973)
Ruiz, F.: MCDM numerical instances library. http://www.univ-valencienne.fr/ROAD/MCDM, international Society on Multiple Criteria Decision Making (2009)
Rylander, B., Soule, T., Foster, J., Alves-Foss, J.: Quantum genetic algorithms. In: Proc. GECCO, pp. 373–377 (2000)
Sahin, M., Tomak, M.: The self-consistent calculation of a spherical quantum dot: a quantum genetic algorithm study. Physica E, Low-Dimens. Syst. Nanostruct. **28**(3), 247–256 (2005)
Sahin, M., Atav, U., Tomak, M.: Quantum genetic algorithm method in self-consistent electronic structure calculations of a quantum dot with many electrons. Int. J. Mod. Phys. C **16**(9), 1379–1393 (2005)
Sailesh Babu, G.S., Bhagwan Das, D., Patvardhan, C.: Real-parameter quantum evolutionary algorithm for economic load dispatch. IET Gener. Transm. Distrib. **2**(1), 22–31 (2008)
Santana, R., Lozano, J., Larrañaga, P.: Protein folding in simplified models with estimation of distribution algorithms. IEEE Trans. Evol. Comput. **12**(4), 418–438 (2008)
Schwefel, H.P.: Evolutionsstrategie und numerische optimierung. PhD dissertation, Technische Berlin, Germany (1975)
Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proc. SFCS, pp. 124–134 (1994)
Shu, W.N.: Optimal resource allocation on grid computing using a quantum chromosomes genetic algorithm. In: Proc. DMAMH, pp. 254–257 (2007)
Shu, W.N., He, B.J.: A quantum genetic simulated annealing algorithm for task scheduling. In: Lecture Notes in Computer Science, vol. 4683, pp. 169–176 (2007)
Sofge, D.A.: Toward a framework for quantum evolutionary computation. In: Proc. CIS, pp. 789–794 (2006)
Spector, L., Barnum, H., Bernstein, H.: Genetic programming for quantum computers. In: Proc. GP, pp. 365–373 (1998)
Spector, L., Barnum, H., Bernstein, H., Swamy, J.N.: Finding a better-than-classical quantum and/or algorithm using genetic programming. In: Proc. CEC, pp. 2239–2246 (1999)
Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**, 17–26 (1994)
Su, H., Yang, Y., Zhao, L.: Classification rule discovery with DE QDE algorithm. Expert Syst. Appl. **37**(2), 1216–1222 (2010)
Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University (2005)
Talbi, H., Batouche, M., Draa, A.: A quantum-inspired genetic algorithm for multi-source affine image registration. In: Lecture Notes in Computer Science, vol. 3211, pp. 147–154 (2004a)
Talbi, H., Draa, A., Batouche, M.: A new quantum-inspired genetic algorithm for solving the travelling salesman problem. In: Proc ICIT, pp. 1192–1197 (2004b)
Talbi, H., Draa, A., Batouche, M.C.: A genetic quantum algorithm for image registration. In: Proc. ICTTA, pp. 395–396 (2004c)
Thompson, J.D., Plewniak, F., Poch, O.: Balibase: A benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics **15**, 87–88 (1999)
Udrescu, M., Prodan, L., Vladutiu, M.: Implementing quantum genetic algorithms: a solution based on Grover's algorithm. In: Proc. CF, pp. 14–16 (2006)
Vlachoglannis, J.G.: Quantum-inspired evolutionary algorithm for real and reactive power dispatch. IEEE Trans. Power Syst. **23**(4), 1627–1636 (2008)
Wang, L., Jiang, T.: On the complexity of multiple sequence alignment. J. Comput. Biol. **1**, 337–348 (1994)
Wang, L., Li, L.P.: An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems. Expert Syst. Appl. **37**(2), 1279–1285 (2010)
Wang, L., Tang, F., Wu, H.: Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. Appl. Math. Comput. **171**(2), 1141–1156 (2005a)
Wang, L., Wu, H., Tang, F., Zheng, D.Z.: A hybrid quantum-inspired genetic algorithm for flow shop scheduling. In: Lecture Notes in Computer Science, vol. 3645, pp. 636–644 (2005b)

Wang, L., Wu, H., Zheng, D.Z.: A quantum-inspired genetic algorithm for scheduling problems. In: Lecture Notes in Computer Science, vol. 3612, pp. 417–423 (2005c)

Wang, L., Niu, Q., Fei, M.R.: A novel quantum ant colony optimization algorithm. In: Lecture Notes in Computer Science, vol. 4688, pp. 277–286 (2007a)

Wang, X.H., Ying, Y., Xiao, J.M.: Application of quantum genetic algorithm in logistics distribution planning. In: Proc. CCC, pp. 759–762 (2007b)

Wang, Y., Feng, X.Y., Huang, Y.X., Zhou, W.G., Liang, Y.C., Zhou, C.G.: A novel quantum swarm evolutionary algorithm for solving 0-1 knapsack problem. In: Lecture Notes in Computer Science, vol. 3611, pp. 698–704 (2005d)

Wang, Y., Feng, X.Y., Huang, Y.X., Pu, D.B., Zhou, W.G., Liang, Y.C., Zhou, C.G.: A novel quantum swarm evolutionary algorithm and its applications. Neurocomputing **70**(4–6), 633–640 (2007c)

Wei, W., Li, B., Zou, Y., Zhang, W., Zhuang, Z.: A multi-objective HW-SW co-synthesis algorithm based on quantum-inspired evolutionary algorithm. Int. J. Comput. Intell. Appl. **7**(2), 129–148 (2008)

Whitley, D., Rana, S., Dzubera, J., Mathias, E.: Evaluating evolutionary algorithms. Artif. Intell. Rev. **85**, 245–276 (1996)

Wu, Q., Jiao, L., Li, Y., Deng, X.: A novel quantum-inspired immune clonal algorithm with the evolutionary game approach. Progr. Nat. Sci. **19**(10), 1341–1347 (2009)

Xiao, W.X., Zang, X., Yan, X.P.: QGA based bandwidth adaptation scheme for wireless/mobile networks. In: Proc. ITST, pp. 1323–1326 (2006)

Xing, H., Ji, Y., Bai, L., Liu, X., Qu, Z., Wang, X.: An adaptive-evolution-based quantum-inspired evolutionary algorithm for QOS multicasting in IP/DWDM networks. Comput. Commun. **32**(6), 1086–1094 (2009a)

Xing, H., Liu, X., Jin, X., Bai, L., Ji, Y.: A multi-granularity evolution based quantum genetic algorithm for QOS multicast routing problem in WDM networks. Comput. Commun. **32**(2), 386–393 (2009b)

Xu, J.J., Chen, H.J., Cheng, Y.H., Luo, R.: Blind signal separation based on quantum genetic algorithm. J. Commun. Comput. **2**(9), 62–66 (2005)

Yang, J.A., Li, Z.Q., Zhuang, Z.Q.: Multi-universe parallel quantum genetic algorithm and its application to blind source separation. In: Proc. ICNNS, pp. 393–398 (2003a)

Yang, J.A., Peng, H., Zhuang, Z.Q.: Research of nonlinear blind source separation algorithm based on quantum evolutionary neural network. In: Proc. ICMLC, pp. 835–840 (2003b)

Yang, J.A., Zhao, B., Ye, Z.F.: Research of blind deconvolution algorithm based on high-order statistics and quantum inspired GA. In: Lecture Notes in Computer Science, vol. 3611, pp. 461–467 (2005)

Yang, Q., Ding, S.C.: Methodology and case study of hybrid quantum-inspired evolutionary algorithm for numerical optimization. In: Proc. ICNC, pp. 634–638 (2007)

Yang, S.Y., Jiao, L.C.: The quantum evolutionary programming. In: Proc. ICCIMA, pp. 362–367 (2003)

Yang, S.Y., Wang, M., Jiao, L.C.: A genetic algorithm based on quantum chromosome. In: Proc. ICSP, pp. 1622–1625 (2004a)

Yang, S.Y., Wang, M., Jiao, L.C.: A novel quantum evolutionary algorithm and its application. In: Proc CEC, pp. 820–826 (2004b)

Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**(2), 82–102 (1999)

You, X., Liu, Y., Shuai, D.: On parallel immune quantum evolutionary algorithm based on learning mechanism and its convergence. In: Lecture Notes in Computer Science, vol. 4221, pp. 903–912 (2006a)

You, X., Shuai, D., Liu, S.: Research and implementation of quantum evolution algorithm based on immune theory. In: Proc. WCICA, pp. 3410–3414 (2006b)

You, X., Sheng, L., Dianxun, S.: Studying the performance of quantum evolutionary algorithm based on immune theory. In: Lecture Notes in Computer Science, vol. 4490, pp. 1068–1075 (2007)

You, X.M., Liu, S., Shuai, D.X.: On improved parallel immune quantum evolutionary algorithm based on learning mechanism. In: Proc. ISDA, pp. 908–913 (2006c)

Yu, Y., Tian, Y.F., Yin, Z.F.: Hybrid quantum evolutionary algorithms based on particle swarm theory. In: Proc. IEA, pp. 309–315 (2006)

Zhang, G., Rong, H.: Parameter setting of quantum-inspired genetic algorithm based on real observation. In: Lecture Notes in Artificial Intelligence, vol. 4481, pp. 492–499 (2007a)

Zhang, G.X., Rong, H.N.: Improved quantum-inspired genetic algorithm based time-frequency analysis of radar emitter signals. In: Lecture Notes in Artificial Intelligence, vol. 4481, pp. 484–491 (2006)

Zhang, G.X., Rong, H.N.: Quantum-inspired genetic algorithm based time-frequency atom decomposition. In: Lecture Notes in Computer Science, vol. 4490, pp. 243–250 (2007b)

Zhang, G.X., Rong, H.N.: Real-observation quantum-inspired evolutionary algorithm for a class of numerical optimization problems. In: Lecture Notes in Computer Science, vol. 4490, pp. 989–996 (2007c)

Zhang, G.X., Jin, W.D., Hu, L.H.: A novel parallel quantum genetic algorithm. In: Proc. PDCAT, pp. 693–697 (2003a)
Zhang, G.X., Jin, W.D., Hu, L.Z.: Quantum evolutionary algorithm for multi-objective optimization problems. In: Proc. ISIC, pp. 703–708 (2003b)
Zhang, G.X., Jin, W.D., Li, N.: An improved quantum genetic algorithm and its application. In: Lecture Notes in Artificial Intelligence, vol. 2639, pp. 449–452 (2003c)
Zhang, G.X., Hu, L.Z., Jin, W.D.: Quantum computing based machine learning method and its application in radar emitter signal recognition. In: Lecture Notes in Artificial Intelligence, vol. 3131, pp. 92–103 (2004a)
Zhang, G.X., Hu, L.Z., Jin, W.D.: Resemblance coefficient and a quantum genetic algorithm for feature selection. In: Lecture Notes in Artificial Intelligence, vol. 3245, pp. 155–168 (2004b)
Zhang, G.X., Li, N., Jin, W.D., Hu, L.Z.: Novel quantum genetic algorithm and its applications. Front. Electr. Electron. Eng. China **1**(1), 31–36 (2006)
Zhang, G.X., Gheorghe, M., Wu, C.Z.: A quantum-inspired evolutionary algorithm based on p systems for knapsack problem. Fund. Inf. **87**(1), 93–116 (2008)
Zhang, J.S., Xu, Z.B., Liang, Y.: The whole annealing genetic algorithms and their sufficient and necessary conditions of convergence. Sci. China Ser. E, Technol. Sci. **27**(2), 154–164 (1997)
Zhang, R., Gao, H.: Improved quantum evolutionary algorithm for combinatorial optimization problem. In: Proc. ICMLC, pp. 3501–3505 (2007a)
Zhang, R., Gao, H.: Real-coded quantum evolutionary algorithm for complex functions with high-dimension. In: Proc. ICMA, pp. 2974–2979 (2007b)
Zhao, S., Xu, G., Tao, T., Liang, L.: Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks. Comput. Math. Appl. **57**(11–12), 2009–2015 (2009)
Zhao, Z., Peng, X., Peng, Y., Yu, E.: An effective constraint handling method in quantum-inspired evolutionary algorithm for knapsack problems. WSEAS Trans. Comput. **5**(6), 1194–1199 (2006)
Zhou, S., Sun, Z.: A new approach belonging to EDAS: Quantum-inspired genetic algorithm with only one chromosome. In: Lecture Notes in Computer Science, vol. 3612, pp. 141–150 (2005)
Zhou, W., Zhou, C., Huang, Y., Wang, Y.: Analysis of gene expression data: Application of quantum-inspired evolutionary algorithm to minimum sum-of-squares clustering. In: Lecture Notes in Artificial Intelligence, vol. 3642, pp. 383–391 (2005)
Zhou, W.G., Zhou, C.G., Huang, Y.X., Wang, Y.: Analysis of gene expression data: application of quantum-inspired evolutionary algorithm to minimum sum-of-squares clustering. In: Proc. FSLCT, SPIE, vol. 6105, pp. 383–391 (2006a)
Zhou, W.G., Zhou, C.G., Liu, G.X., Lv, H.Y., Liang, Y.C.: An improved quantum-inspired evolutionary algorithm for clustering gene expression data. Comput. Methods, pp. 1351–1356 (2006b)
Zitzler, E., Laumanns, M.: Problems and test data for multi-objective optimizers. http://www.tik.ee.ethz.ch/zitzler/testdata.html (1999)
Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the performance of the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Communication Networks lab (TIK), Swiss Federal Institute of Technology (ETH) (2001)