A Real-Coded Quantum-Inspired Evolutionary Algorithm for Global Numerical Optimization

Chaoyong Qin
Department of Mathematics and Information Science
Guangxi University
Nanning, China
Email:qcy@mail.dhu.edu.cn

Yongjuan Liu
Department of Computer science
Guangxi University of Technology
Liuzhou, China

Jianguo Zheng
School of Business and Management
Donghua University
Shanghai, China

***Abstract*—In this paper, a novel kind of algorithm, Real-coded quantum-inspired evolutionary algorithm (RQEA), is proposed based on evolutionary programming and quantum computation. In RQEA, several real numbers are directly encoded in a chromosome which is usually represented by quantum bits in quantum evolutionary programming. Quantum computation mechanics is employed to accelerate evolution process. The result of experiments shows that RQEA has a strong ability of global optimization and high convergence speed.**

***Keywords*—quantum computation; evolutionary algorithm; quantum-bit chromosome; real-coded**

I. INTRODUCTION

Recent developments in quantum technology have shown that quantum computers can provide dramatic advantages over classical computers for some problems [1],[2]. Quantum algorithms rely upon the inherent parallel qualities of quantum mechanics to achieve their improvement. In recent years, quantum theory has been widely used in intelligence computation. Quantum-inspired genetic algorithm (QGA), which were firstly developed by Narayanan [3], is a novel algorithm. QGA achieves much better performance than classical genetic algorithm. Reference [4] combined evolutionary theory and quantum theory to propose a new kind of evolutionary programming, the Quantum Evolutionary Programming (QEP), achieved rapid convergence and good global search capacity. The application of quantum parallelism to classical intelligence programming has been promising.

The existing QGA and QEA (QEP) rely on the concepts of quantum bits, and on superposition of states from quantum mechanics. Only binary strings can be obtained by observing the probability amplitudes of quantum bits. The quantum-inspired evolutionary algorithm with binary representation [5] works properly in problems where this kind of representation is more suited. But, as the situations of numerical optimization, representation by real numbers is more efficient. Reference [6] proposed a novel representation for quantum-inspired evolutionary algorithm and obtained good performance in optimization problems.

Enlightened by them, this paper presents a real-coded quantum-inspired evolutionary algorithm (RQEA). Instead of binary or quantum bits representation, RQEA adopts a novel chromosome as a representation. Each gene in chromosome is defined by a rectangle which is located by its center and radius. Due to this representation of states, one chromosome can takes the information of several individuals. Interference process forms the superposition states which bring many benefits to RQEA such as inherent parallel and high convergence speed. Hereinafter we will explain it in details.

II. REAL-CODED QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

A global numerical optimization can be formulated as solving the following objective function

$$\min f(x), x = (x_1, x_2, ..., x_n) \in S \qquad (1)$$

Where $S = [\underline{x}, \overline{x}]^n$ and $S \subseteq R^n$ defines the search space which is a n-dimensional space bounded by the parametric constraints $\underline{x_i} \le x_i \le \overline{x_i}, i = 1,2,...,n$ .Here, $\underline{x}$ presents the low bound of search space, $\overline{x}$ presents the upper bound of search space.

A. Chromosome base on real- coded

The core of quantum computation lies on the superposition states of quantum bits. The superposition brings many benefits to quantum computation such as inherent parallel, which make the most different between classical computation and quantum computation. To take the advantage of the superposition states of quantum bits, this paper proposes a new chromosome with real coded.

978-1-4244-1674-5/08 /$25.00 ©2008 IEEE

Definition 1: A chromosome represents a candidate solution to the optimization problem in hand. A gene is defined with a pair of numbers (c, r). An m-gene chromosome is defined as:

$$
\begin{aligned}
q_i &= [v_1, v_2, \ldots, v_n] \\
&= [(c_1, r_2), (c_2, r_2), \ldots, (c_n, r_n)], (i = 1, \ldots, m)
\end{aligned} \tag{2}
$$

Where n defines the dimensions of the search space, m is the size of population, c_j and r_j represents respectively the center and the radius of the j th variance ($j = 1, 2, \ldots, n$). An example of chromosome gene is shown in figure 1. Gene $v_j = (c_j, r_j)$ includes all the possible value in the rectangle area specified by the center c_j and the radius r_j . One chromosome takes several individuals information. Exact value of an individual could be obtained by observing process.

Inspired by the quantum bit state of 0 and 1, the low bound and the upper bound of search space can be considered respectively as "0" and "1" state. All candidate solution states are located between "0" and "1" states. And the main target of optimization process is to increase the probability of suitable solution.

Definition 2: The energy of q_i is h_i , it represents the relatively fitness of an individual among the population.

$$
h_i = \frac{E_i}{\sum_{k=1}^{l} E_k} \tag{3}
$$

Where $E(x_i) = M - f(x_i)$ .Here M is a constant to ensure $E(x_i) > 0$. In RQEA, h_i denotes the height of gene rectangle as shown in figure.1. Equation (3) guarantees that all individuals will have a total height equal to 1.

Equation (4) implements the interference process among individuals. This process basically consists in summing up the individuals that form the quantum population. In other words, the radiuses of gene in the same position of all individuals are taken into consideration. If the domains of gene constraint by the radius are overlapping, then interference process happen and superposition states are formed. ϕ presents the result of interference process,

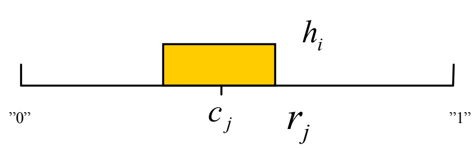

Fig.1 Gene v_j

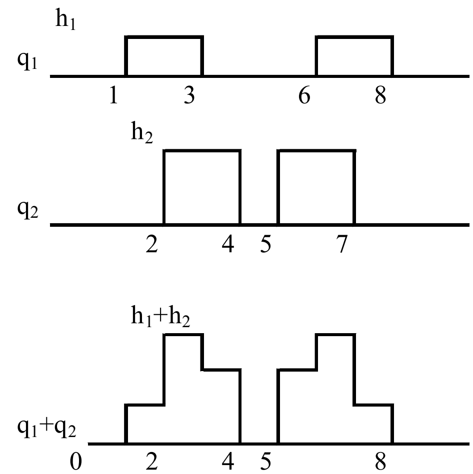

Fig.2 Interference result

$$
\phi_i = \sum_{j=1}^{m} (c_j, r_j), (i = 1, \ldots, n) \tag{4}
$$

For example, considering two individuals $q_1 = [(2,1), (7,1)]$ and $q_2 = [(3,1), (6,1)]$, which energy is h_1 and h_2 respectively. The domain of the first gene of two individuals is (1,3) and(6,8), the domain of the second gene is (2,4) and (5,7). Each gene and the result of interference are shown in figure 2.

B. *Quantum-inspired evolutionary operators*

Crossover operator: In classical genetic algorithm and evolutionary programming, crossover operator was only taken between two individuals and the information was shared in a very small scale. In RQEA, crossover operator is performed among all individuals of population, each individual make full use of the whole population information. This mechanism increases the probability of generating better individual in evolution program. Supposed the size of population is 6, the length of chromosome is 7. Table I lists the results after perform crossover operator.

TABLE I. CHROMOSOME AFTER PERFORM CROSSOVER OPERATOR

Individual	**Chromosome**						
q_1	v_{11}	v_{62}	v_{53}	v_{44}	v_{35}	v_{26}	v_{17}
q_2	v_{21}	v_{12}	v_{63}	v_{54}	v_{45}	v_{36}	v_{27}
q_3	v_{31}	v_{22}	v_{13}	v_{64}	v_{55}	v_{46}	v_{37}
q_4	v_{41}	v_{32}	v_{23}	v_{14}	v_{65}	v_{56}	v_{47}
q_5	v_{51}	v_{42}	v_{33}	v_{24}	v_{15}	v_{66}	v_{57}
q_6	v_{61}	v_{52}	v_{13}	v_{34}	v_{25}	v_{16}	v_{67}

Genes in each row in table I form a new chromosome. The whole system information is shared by each new chromosome. This mechanism brings new characteristic to the population and improves the probability of generating better individuals.

Mutation operator: Mutation operator is applied to increase the diversity of the population so as to enhance the chance for RQEA to escape from local optima. Mutation happens on each chromosome gene with a probability of pm , and this probability should be kept small; otherwise, the favorable building blocks discovered so far by RQEA will be exhaustively destroyed, which implies failure. Suppose $q_{\max}^{t} = (v_{m1}, v_{m2,} \ldots, v_{mj,} \ldots, v_{mn})$ is the individual with maximum energy in previous t generation. Given probability pm and pt , mutation process can be described as follow:

if $U_1(0,1) > pm$,

$$v_j' = v_j;$$

if $U_1(0,1) \le pm$,

$$\begin{cases} v_j' = v_{new}, & U_2(0,1) \ge pt \\ v_j' = v_{mj} + \sigma r_j U(-1,1), & U_2(0,1) < pt \end{cases} \quad (5)$$

Where $U_1(0,1)$ and $U_2(0,1)$ are uniform random number generator. $U(-1,1)$ generates uniform random number from rage (-1,1).When $U_2(0,1) \ge pt$,new gene v_j' is generated randomly from search space, while when $U_2(0,1) < pt$, we use the best individual to steer the direction of mutation. Variance σ controls the extent of mutation process.

Self-learning operator: For numerical optimization problems, integrating local searches with evolution algorithm can improve the performance. There are several ways to realize the local searches. In RQEA, we propose the self-learning operator which uses a small scale RQEA to realize the behavior of local search. Also , $q_{\max}^{t}$ is supposed as the individual with maximum energy till t-th generation. The population size of this small scale RQEA is assigned as sm . For more clarity, ALGORITHM 1 describes the details of this operator.

ALGORITHM 1 Self-learning operator

step 1 : Generate population $F(t)$ in the space which constraints by the low bound and upper bound of $q_{\max}^{t}$;

step 2 : Generate classical population $G(t)$ according $F(t)$;

step 3 : Compute fitness of each individual in $G(t)$;

step 4 : Perform interference process, generate new individuals from space with high energy after overlapping , form population $F(t)'$;

step 5 : Perform crossover operator on $F(t)'$, obtaining $F(t)''$;

step 6 : Perform mutation operator on each individual of $F(t)''$,obtaining $F(t)'''$;

step 7 : Generate classical population $G'(t)$ from $F(t)'''$;

step 8 : Find the individual with maximum energy in $G'(t)$: max($G'(t)$),if $E(q_{\max}^{t}) < E(\max(G'(T))$ then $q_{\max}^{t} \leftarrow \max(G'(t))$.

ALGORITHM 1 is performed on $q_{\max}^{t}$, if an individual with higher energy is found after local search, then $q_{\max}^{t}$ is replaced.

C. Quantum-inspired evolutionary algorithm

ALGORITHM 2 Real-coded quantum-inspired evolutionary algorithm (RQEA)

step 1: Generate population $Q(t)$, $t \leftarrow 0$;

step 2: Generate classical population $P(t)$ according $Q(t)$;

step 3: Compute energy of each individual in population $P(t)$;

step 4: Perform interference process;

step 5: Generate new individuals from space with high energy after overlapping, select best m individuals from $Q(t)$ and new individuals ,obtaining $Q(t)'$;

step 6: Perform crossover operator on $Q(t)'$,obtaining $Q(t)''$;

step 7: Perform mutation operator on each individual of $Q(t)''$, obtaining $Q(t)'''$;

step 8: Generate classical population from $Q(t)'''$, select m individuals with high energy form $Q(t)'$ and $Q(t)'''$,population $Q(t)$ is replaced, find $q_{\max}^{t}$;

step 9: Perform self-learning operator on $q_{\max}^{t}$;

step 10: If termination criteria are reached, output q_{max}^t and stop; otherwise, go to Step 2.

There are two ways to generate classical population. We can take the center of each rectangle or generate one point randomly in each rectangle. In RQEA, the later is adopted.

III. EXPERIMENTS

Numerical experiments are conducted to test the effectiveness and efficiency of RQEA. Some parameters must be assigned to before RQEA is used to solve problems. m is population size and can be chosen from 5~20. The value of r depends on search space and should be set less than $\frac{(\overline{x}-\underline{x})}{2m}$, usually $r=(0.1\sim 0.8)*\frac{(\overline{x}-\underline{x})}{2m}$. Mutation probability pm can be chosen from 0.01~0.3. When $pt<0.5$, RQEA puts emphasis on searching in the new space, while when $pt>0.5$, on making use of information of the best individual so far. On account of the computational cost, it is better to let sm be small than 5 and choose from 5 to 10.

The benchmark functions used here is a set of 8 different functions selected from [7]. These functions are widely used as benchmark in numerical optimization and the cited paper allows the comparison between results using the following methods:

1) Stochastic Genetic Algorithms (StGA) [7];

2) Fast Evolutionary Programming (FEP) [8].

The test functions are list as follow:

$$f_1(x)=\sum_{i=1}^{n} x_i^2, S=[-100,100]^n, f_{\min}=0;$$

$$f_2(x)=\sum_{i=1}^{n}|x_i|+$$
$$\prod_{i=1}^{n}|x_i|, S=[-10,10]^n, f_{\min}=0;$$

$$f_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2, S=[-100,100]^n,$$
$$f_{\min}=0;$$

$$f_4(x)=\max_i(|x_i|, 1\le i\le n), S=[-100,100]^n,$$
$$f_{\min}=0$$

$$f_5(x)=\sum_{i=1}^{n}\left[100(x_i-x_{i+1})^2+(x_i-1)^2\right],$$
$$S=[-30,30]^n, f_{\min}=0;$$

$$f_6(x)=\sum_{i=1}^{n}(-x_i\sin\sqrt{|x_i|}), S=[-500,500]^n,$$
$$f_{\min}=-418.983n;$$

$$f_7(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-$$
$$\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)), S=[-32,32]^n, f_{\min}=0;$$

$$f_8(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10],$$
$$S=[-5.12,5.12]^N, f_{\min}=0;$$

$f_1\sim f_5$ are unimodal functions, which are relatively easy to optimize, but the difficulty increases as the problem dimension goes high. $f_6\sim f_8$ are multimodal functions with many local optima, and they represent the most difficult class of problems for many optimization algorithms. As an example, Fig. 3 shows the surface landscapes of f_8 when the dimension is set to 2.

The parameter configuration for the RQEA for all the problems above is given in table Ⅱ.

For each function 50 experiments were made. The number of generations was chosen in order to make the total number of evaluations equal to the number of evaluations for the Stochastic Genetic Algorithm in [7]. The mean function value for each experiment and standard deviation are shown in table Ⅲ.

The results show that RQEA was able to reach better results with much less evaluations than FEP. For f_6, the solutions of RQEA are as good as StGA with little more number of function evaluations. For the others functions, both the mean function value and the standard deviation of RQEA are much better than those of StGA with the approximately same number of function evaluations. To summarize, the results show that RQEA outperforms FEP and StGA, and is competent for the numerical optimization problems.

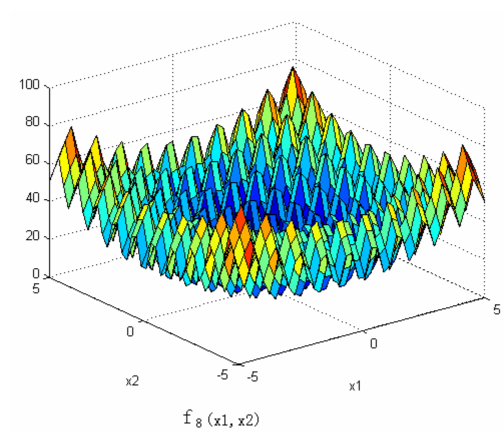

Fig.3 Graphs of f_8 with dimension of 2

TABLE II. PARAMETERS FOR THE RQEA

Test Function	m	Number of generations	sm	Mean number of function evaluations
f_1	10	500	5	29500
f_2	10	300	5	18000
f_3	10	340	5	21000
f_4	10	580	5	32000
f_5	10	720	5	43000
f_6	10	40	5	1900
f_7	10	180	5	10000
f_8	10	450	5	28300

TABLE III. COMPARISON BETWEEN FEP, StGA, AND RQEA ON FUNCTIONS WITH 30 DIMENSIONS

TF	Mean number of function evaluations(×100)			Mean function value (Standard deviation)		
	RQEA	***StGA***	***FEP***	***RQEA***	***StGA***	***FEP***
f_1	295	300	1500	6.42×10^{-19} 2.12×10^{-18}	2.45×10^{-15} 5.25×10^{-16}	5.7×10^{-4} 1.3×10^{-4}
f_2	180	176	2000	4.73×10^{-12} 5.15×10^{-12}	2.03×10^{-7} 2.95×10^{-8}	8.1×10^{-3} 7.7×10^{-4}
f_3	210	230	5000	7.78×10^{-18} 4.91×10^{-16}	9.98×10^{-29} 6.9×10^{-29}	1.6×10^{-2} 1.4×10^{-2}
f_4	320	320	5000	8.2×10^{-18} 3.42×10^{-15}	2.01×10^{-8} 3.42×10^{-9}	0.30 0.50
f_5	430	450	20000	3.62×10^{-10} 2.73×10^{-8}	0.04435 0	5.06 5.87
f_6	19	15	9000	-12569.5 0.0	-12569.5 0.0	-12554.5 52.6
f_7	100	100	1500	3.98×10^{-10} 2.35×10^{-11}	3.52×10^{-8} 3.51×10^{-9}	1.8×10^{-2} 2.1×10^{-3}
f_8	283	285	5000	4.51×10^{-15} 3.28×10^{-14}	4.42×10^{-13} 1.14×10^{-13}	4.6×10^{-2} 1.2×10^{-2}

TF: Test Function

IV. CONCLUSION

In this paper, a real-coded quantum-inspired evolutionary algorithm (RQEA) is presented to deal with global optimization problems. A rectangle area is adopted to take the place of one point coding mechanism, one chromosome take several individuals information. The rectangle height is represented by individual energy. The solutions are guided toward the region with high energy after interference process. An effective crossover scheme is proposed such that each individual can make full use of the whole population information. The information of best individual is used to steer evolution process. Self-learning operator searches better solution in local environments.

The algorithm is tested on 8 functions of moderate dimensions. Results obtained from 50 trials for each function show that the RQEA is able to find the near-global solution for all these test functions. Comparison demonstrates that RQEA outperforms StGA and FEP in terms of effectiveness, as well as efficiency.

[1] P. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring, Proceedings 35th annual symposium on foundations of computer science, pp. 124-134,1994.

[2] L. Grover, A Fast Quantum Mechanical Algorithm for Database Search, Proceedings of the 28th annual ACM symposium on the theory of computing, pp, 212-219,1996.

[3] A. Narayanan and M. Moore, Quantum-inspired Genetic Algorithm. Proceedings of IEEE international conference on evolutionary computation. Piscatawa: IEEE Press, 1996.

[4] Shuyuan Yang and Licheng Jiao, The quantum evolutionary programming, Proceedings of the fifth international conference on computational intelligence and multimedia applications (ICCIMA'03), IEEE, 2003.

[5] K.H. Han and J.H. Kim, Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Transactions on Evolutionary Computation 6 (2002) 580–593.

[6] Cruz A V A, Vellasco M B R and Pacheco M A C, Quantum-inspired Evolutionary Algorithm for Numerical Optimization. In: IEEE Congress on Evolutionary Computation, Canada, 2006, 2630-2637.

[7] Z. Tu and Y. Lu, A robust stochastic genetic algorithm (stga) for global numerical optimization. IEEE Trans. Evolutionary Computation,vol. 8, no. 5, pp. 456–470, 2004.

[8] X. Yao, Y. Liu and G. Lin, Evolutionary programming made faster, IEEE Trans Evolutionary Computation, vol.3, pp. 82–102, July 1999.