

A Novel Self-adaptive Behavior Quantum Evolutionary Algorithm

Hassan K. Khalafi

School of Mathematics and Computer Science, Damghan University,
Damghan, Semnan, P.O. Box 36715-364 Iran
h_kadkhodaei@du.ac.ir

Abstract: In accordance with tradition quantum evolutionary algorithms can obtain excellent results in the optimization of Multi-peak functions. In any case, they are easy to be trapped to hurriedness. In this article, A Novel Self-adaptive Behavior Quantum Evolutionary Algorithm is recommending on the basis of the concepts and tenet of quantum evolutionary algorithms in order to enhance the efficiency. Firstly, Self-adaptive Behavior triploid chromosome is constructed to keep the population variety; Secondly, double mutation is used to make sure the variety of the swarm, then individual chromosome cross will be imported into this new algorithm in order to achieve the information communication between the chromosomes and enlarge the search scope in the available space. Experiments on test functions of varied intricacies are implemented and compared with other EAs. The result indicates that the new algorithm in this article can search and get the global most efficient solution in a shorter time. [Hassan K. Khalafi. **A Novel Self-adaptive Behavior Quantum Evolutionary Algorithm**. Journal of American Science 2010;6(12):1483-1486]. (ISSN: 1545-1003). <http://www.americanscience.org>.

Keywords: Self-adaptive Behavior; Evolutionary Algorithm; QEA; Double Mutation; Discrete Cross

1. Introduction

Evolutionary algorithms (EAs) such as Genetic Algorithm [1], Evolutionary Programming [2], Evolutionary Strategies [3], which can find out the best solution when they are used to deal with the complexity problems. So they have widely applied into parameters estimate, mode recognition, machine learning, neural network, industry control and so on. However, when they are used to deal with the optimization of non-linear system, both of them often exist in the problem of prematurity and slow convergence speed.

The quantum evolutionary algorithm [4] is proposed in order to solve above problems. QEA is a new probability optimization method based on quantum computing theory. In recent years, much attention is paid to QEA because it has the characteristics of strong parallelism, rapid convergence, good search capability, short computing time, and small population size. Some QEAs and its improvement have been proposed for some combinatorial optimization problems. Although the QEA show better performance in solving combinatorial optimization problems than the traditional evolutionary algorithm, it is not suitable for solving complex optimization problems. So this paper presents a real-coded quantum evolutionary algorithm (RCQEA). RCQEA uses the variable component of the solving complex functions and qubit to construct a real-coded triploid chromosome in order to increase the diversity of the swarm, then all of the individuals (chromosomes) will evolve based on the double mutation and random discrete cross in order to achieve a balance between the local

search and the global search. Standard simulation on the testified function shows that RCQEA which is used for solving complex optimization function has a very good performance.

2. Overview of QEA

As a new research field, QEA combines the quantum computing with genetic evolutionary algorithm. QC deals with investigations on quantum mechanical computers and quantum mechanics like qubits representation and superposition of states. QC can process huge numbers of quantum states simultaneously in parallel.

In QEA, the smallest unit of information stored in two-state quantum computer is called qubit, which maybe in the "1" state, or in the "0" state, or in any superposition of the two. The state of a qubit can be described as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where α, β are complex numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ gives the probability that the qubit will be found in the 0 state and $|\beta|^2$ gives the probability that the qubit will be found in the 1 state. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1$$

The state of a qubit can be changed by the operation with a quantum gate. Inspired by the concept of quantum computing, QEA is designed with a novel Q-bit representation, a Q-gate as a variation operator, and an observation process.

A Q-bit individual as a string of n Q-bits is defined as

$$q_j^t = \begin{bmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \dots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \dots & \beta_{jm}^t \end{bmatrix}$$

Where m is the number of qubit, i.e., the string length of the qubit individual, and $j = 1, 2, \dots, n$. The following rotation gate is used as a Q-gate in QEA, such

$$U(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Where θ is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign.

The structure of QEA implies that most operations of QEA are based on the probabilistic, and QEA can not use any prior knowledge in the whole computing process.

The basic structure of QEA is described in the following [5]:

Procedure QEA

Begin

$t \leftarrow 0$

Initialize $Q(t)$

Make $P(t)$ by observing the states of $Q(t)$

Evaluate $P(t)$

Store the best solutions among $P(t)$ into $B(t)$

While (not termination condition) do

Begin

$t \leftarrow t + 1$

Make $P(t)$ by observing the states of $Q(t-1)$

Evaluate $P(t)$

Update $Q(t)$ using Q-gates

Store the best solutions among $B(t-1)$ and

$P(t)$ into $B(t)$

Store the best solution b among $B(t)$

If (global migration condition)

Then migrate b to $B(t)$ globally

Else if (local migration condition)

Then migrate b to $B(t)$ locally

End

End

Where $Q(t)$ is a population of qubit chromosomes at generation t , and $P(t)$ is a set of binary

solutions at generation f .

3. A New Real-Coded Quantum Evolutionary Algorithm

In this paper, the basic idea of RCQEA is as follows: Firstly, to reject the traditional encoding method in order to form a structure of real-coded triploid chromosome; then the chromosomes will be evolved based on the specific design of the real-coded triploid chromosome; Thirdly, double mutation is used to make sure the diversity of the swarm, then discrete chromosome cross will be imported into this new algorithm in order to achieve the information communication between the chromosomes and enlarge the search scope in the available space. At last, the new algorithm create the new chromosome through improved quantum gate.

3.1. Real-coded structure

As the multi-peak function optimization problem, it generally consists of two parts: The objective function:

$$\text{Min}(f(X)) = f(x_1, x_2, \dots, x_n)$$

Subject to: $L \leq X \leq U$

$$X = \{(x_1, x_2, \dots, x_i, \dots, x_n) \mid l_i \leq x_i \leq u_i\} \in R^n$$

$$i = 1, 2, \dots, n$$

Where $L = (l_1, l_2, \dots, l_n)$, $U = (u_1, u_2, \dots, u_n)$, L and U are the solution space, as for the since the variables x_i , its lower bound is l_i , and its upper bound is u_i . Real-coded triploid chromosome is composed of a variable component of the function and qubit. That is to say, the new chromosome is described as follows:

$$\begin{bmatrix} \alpha_1 & \dots & \alpha_i & \dots & \alpha_n \\ \beta_1 & \dots & \beta_i & \dots & \beta_n \\ x_1 & \dots & x_i & \dots & x_n \end{bmatrix}$$

Where $|\alpha|^2 + |\beta|^2 = 1$

3.2. Double mutation and random discretecross

In this paper, every new chromosome in the swarm will be mutated in one gene bit, that is, only one gene bit in the chromosome will be mutated every time, and other gene bits in this chromosome will keep invariability in order to build a new chromosome. The experiment has show that single gene mutation has higher search efficiency than the entire gene mutation [6].

Suppose the swarm in the t generation is:

$$P(t) = \{p_1^t, p_2^t, \dots, p_j^t, \dots, p_N^t\}$$

As a chromosome p_j^t , the new algorithm will select one gene bit random:

$$[\alpha_{ji}^t, \beta_{ji}^t, x_{ji}^t]^T, i = 1, 2, \dots, n$$

variable x_{ji}^t can be mutated as follows:

$$x_{ji}^{t+1} = x_{ji}^t + \sigma_{ji}^t N(0, 1)$$

$$\sigma_{ji}^t = (u_j - l_j) \exp\left(\frac{-\mu k + \sqrt{\mu k}}{2}\right)$$

Where $0 \leq \mu \leq 0.3$, μ is a step, which will be changed in a smaller range. The value of μ is related to the expectations precision of the objective function, k is the number of generation. In order to control the variation does not exceed the value of the domain, the parameters is adjusted as follows:

$$\text{If } x_{ji}^{t+1} + 1 > u_j$$

$$\text{Then } x_{ji}^{t+1} = 2 \times u_j - x_{ji}^{t+1}$$

$$\text{If } x_{ji}^{t+1} < l$$

$$\text{Then } x_{ji}^{t+1} = 2 \times l_j - x_{ji}^{t+1}$$

Repeat this process until x_{ji}^{t+1} can meet the conditions of a viable solution. If the new chromosome after mutation is better than the original chromosome, then this process is considered as a beneficial mutation for the evolution; otherwise as a invalid mutation for the evolution. As to the $\alpha_{ji}^t, \beta_{ji}^t$ in the original chromosome, the new algorithm will make mutation in its low bit (multi-point mutation), which can represent a larger search space, increase the diversity of the swarm. In this paper, when the solving function is very complex and the variables have strong correlation, so the algorithm will carry out random separated-cross in order to avoid the algorithm trapped into the local optimization, the process is as follows: Select an appointed chromosome p_u^t , and select another chromosome p_v^t random from the swarm, ($u \neq v$). Then both of them will carry out random separated-cross as follows:

$$i = \text{fin}(\text{random} \times N)$$

Where i is the cross location in the chromosome, $\text{fin}(x)$ is a function, its result will be the maximum integer which is less small than x . random will

create a new random number between $[0, 1]$; N is the length of chromosome.

The new two chromosomes after the random separated-cross are through the gene bit exchange in the location of i .

3.3 The procedure of RCQE

Begin

Step 1. $t = 0$ initialize $Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$

according to the formula 6, define a empty memory storeroom;

Step 2. Make $R(t)$ by observing the states of $Q(t)$;

Step 3. Evaluate the whole swarm, and select the best chromosome;

Step 4. While the termination is true, the output of RCQE is p_{best}^t , and the algorithm

ends, otherwise, go to the next step;

Step 5. The chromosomes are evolved according to the 3.2;

Step 6. $t = t + 1$ go to step 4. End

4. Experimental results

0/1 knapsack problem [7] is referred to n articles with various value and weight, as well as partial articles are selected. To each article, there are two ways: select or not. The total weight of selected articles can't overrun that of knapsack appointed boundary and should reach the maximum of total value. If the total weight of all articles is less than that of the knapsack, then the problem will be extremely simple. And the benefit is equal to the total value of the whole articles. But actually, the knapsack's weight is always less than the total weight of the articles.

0/1 knapsack problem is an effective criterion to verify the performance of all kinds of algorithms. 0/1 knapsack problem is a typical combinatorial optimization problem, it belongs to a NP-complete, we can describe it as follows:

Suppose something related to travel, its number is n , the quality and value of each one is $w_i (w_i > 0), c_i (c_i > 0), (i = 1, 2, \dots, n)$. The capacity of this knapsack is $V (V > 0)$, then to solve an answer $x = (x_1, x_2, \dots, x_n)$ to make sure the total value of this knapsack which is loaded with a lot of things is the most largest. We can describe this question as follows:

$$\max f(x_1, x_2, \dots, x_n) = \sum_{i=1}^n c_i x_i$$

Subject to

$$\sum_{i=1}^n w_i x_i \leq V, \quad x_i \in \{0,1\}, (i=1,2,\dots,n), x_i \text{ is}$$

a decision variable, if $x_i = 1$, which means res i

has loaded in this knapsack; if $x_i = 0$, which means

res i has not loaded in this knapsack.

In order to get a result, all the test data in this experiment have the strong correlation between weight and value.

$$w_i = \text{random}([1,10]) \text{ (Equably)}$$

$$c_i = w_i + 5$$

And the average capacity of the knapsack:

$$V = 0.5 \sum_{i=1}^m w_i$$

Before we solve knapsack problem, we have some prior knowledge, which are:

- When people begin to install a knapsack, they will choose bigger value of “profit/weight” firstly;
- When people begin to eliminate the capacity of knapsack, they will choose smaller value of “profit/weight” to discard firstly.

So this is character information, and we will use them in our new RCQEA. In order to compare the new RCQEA with others in this paper, we adopt traditional GA, QEA, and RCQEA to validate the knapsack problem. The matlab 7.0 software was adopted for the implementation of the approach described previously. The number in knapsack problem is 250, and 500 respectively, and the size of population is 100 in GA and QEA. The size of population is 50 in RCQEA. The three evolutionary algorithms run 50 times and get their statistical results, respectively. The statistical results are in Table 1. We can find that the RCQEA's performance improve obviously compared with GA and QEA.

Table 1. statistical results

Number	Value	GA	QAE	RCQEA
300	Best	1653.4	1681.3	1732.1
	Average	1507.6	1595	1671.2
	Worst	1302.7	1421.4	1620.4
50	Best	2722.3	2907.2	3301.5
	Average	2599.4	2767.1	3113.4
	Worst	2278.2	2635.9	2997.1

5. Conclusions

This paper proposed a new algorithm RCQEA, inspired by the concept of Real-Coded. Compared with the traditional quantum evolutionary algorithm, its encoding method is convenient, running time is short, the speed of convergence is fast, and its global search ability is strong. So it is suitable for optimizing the complex functions such as 0/1 knapsack problems.

Corresponding Author:

Hassan K. Khalafi
School of Mathematics and Computer Sciences
Damghan University
Damghan, Semnan, P.O. Box 36715-364, Iran
E-mail: h_kadkhodaei@du.ac.ir

References:

1. Holland J H. Genetic algorithms and classifier systems: foundations and their applications[C], Proceedings of the Second International Conference on Genetic Algorithms DHillsdale, NJ : F Lawrence Erlbaum Associates, 1987, pp82-89.
2. Fogel L J, Owens A J, Walsh M J, Artificial Intelligence Through Simulated Evolution[M], Chichester: John Wiley,1966
3. Hans Paul Schwefel, Evolution and Optimum Seeking: A Wiley Interscience Publication, John Wiley & Sons, Inc., New York, NY, 1994.
4. Ren Ailian, Zhou Liang, Qu Hongjian, An Improved Quantum Evolutionary Algorithm and Its Application [C], Proceedings of the 3rd International Conference on Product Innovation Management, 2008, 10, pp1428-1431
5. Kuk-Hyun Han, Jong-Hwan Kim, On the analysis of the quantum-inspired evolutionary algorithm with a single individual, in: Proceedings of 2006 IEEE Congress on Evolutionary Computation, Vancouver. 2006, pp2622-2629
6. Wang Xiangzhong, Yu Shouyi, Improved evolution strategies for high-dimensional optimization[J], Control Theory & Applications, Vol.23, No.1, 2006, pp148-151
7. Hanxiao SHI, Solution on 0/1 Knapsack Problem Based on Improved Ant Colony Algorithm, Proceedings of the 2006 IEEE International Conference on Information Acquisition, 2006, 8, pp1062-1066