Journal of Engineering Science and Technology Review 6 (5) (2013) 25-32

JOURNAL OF
Engineering Science
and Technology Review

www.jestr.org

Research Article

Real Coded Quantum Genetic Algorithm and its Application

Ying Sun* and Hegen Xiong

[1] *The Key Laboratory for Metallurgical Equipment and Control of the Ministry of Education,*
Wuhan University of Science and Technology, Wuhan

Received 15 August 2013; Accepted 23 December 2013

Abstract

Optimization design, as a new branch of science in recent years, is the result of optimization technique and computer technology applied in design. Optimization design provides an important scientific design method for engineering designer, which seeks the most perfect and optimum design proposal and greatly improves design efficiency and quality. Genetic Algorithm (GA) is an overall random searching method based on the Darwin's evolution and Mendelian. Genetic algorithm has better adaptability and wide accomodation. But it also has the disadvantages of poor local search ability, slow convergence and other defects. Quantum Genetic Algorithm (Quantum Genetic QGA) is produced by combining Quantum computing and Genetic Algorithm. And it is got enormous attention because of its small population size, higher ability to find the global optimal value, fast convergence rate and time-saving. A new algorithm named Real Coded Quantum Genetic Algorithm (RQGA) is proposed. After systematically introduce algorithm flow chart, key techniques and characteristics of SA and QGA, based on the characteristics of mechanical optimization problems, this paper proposes a new real number coded quantum genetic algorithm, and introduces coding method, operators, algorithm flow chart and etc. Furthermore, this paper also describes the algorithm implementation based on a Java language. Take reducer that is widely used in mechanical for example, and study the application of RQGA in the mechanical structure optimization design. Establish the optimization model is established and is resolved by using RQGA. Then the solution of SA and QGA are analyzed and compared. And the comparison shows the validity and feasibility of the RQGA.

Keywords: Real Number Coding, Genetic Algorithm, Quantum Genetic Algorithm, Optimal Design Application

1. Introduction

Optimization design is an effective design method to ensure the products have excellent performance, reduce the weight and size of products and reduce the cost. Mechanical optimization design is the transfer and application of optimization design in mechanical field, its basic idea is: first, translate the engineering design problem into optimization problem, then select the appropriate optimization method and use computer to automatically find the optimal design scheme to achieve the desired objectives from the feasible design scheme which meets the requirements. The optimal design method in machinery design, one hand it adapts the requirements of modern production, and on the other hand, it meets the need of the reality of energy-saving society construction, so more and more scientific workers and engineering pay attention to mechanical optimization design, and it has obtained considerable development. So far, as the foundation and core of optimization design, optimization algorithm has hundreds of species.

Overall, the optimization algorithm can be divided into traditional optimization algorithms and modern optimization algorithm, with the popularization and application of computer technology, modern optimization algorithm has been developed greatly. So far, the modern optimization algorithms include simulated annealing algorithm (SA), genetic algorithm (GA) and ant colony algorithm (ACA). Genetic algorithm is an important branch of evolutionary algorithm, it is widely used because better robustness and wide adaptability, at the same time, genetic algorithm has many deficiencies like having more iterative times, slow convergence speed and easily getting into local extremum. With the development of quantum computing, the concept of quantum genetic algorithm is proposed based on the concept of qubit and quantum superposition state. The quantum genetic algorithm (QGA) is a probability search optimization method combined quantum computation and genetic algorithm, quantum genetic algorithm has more diversities of the population, better stability of solution and solving speed [1,3,12].

2. Real Coded Quantum Genetic Algorithm

2.1 Code Element

In this algorithm, a quantum bit is as follows:

* E-mail address: wustsunying@126.com

ISSN: 1791-2377

$$|\phi\rangle = \alpha \cdot |x^l\rangle + \beta \cdot |x^u\rangle \tag{1}$$

Among them, α , β are complex constants, respectively represents the probability amplitudes when quantum is Lower limit state and upper limit state, and all of them satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$.

If there is an N dimension function optimization problem, the mathematical programming model is as follows:

$$\begin{aligned} &\min \quad f(X) = f(x_1, x_2, \text{L} \ , x_i, \text{L} \ , x_N) \\ &s.t. \quad x_i^l \le x_i \le x \quad i = 1,2,\text{L} \ ,N \end{aligned} \tag{2}$$

Among them x_i^l and x_i^u were the lower and upper bounds of variable x_i .

The quantum bit code corresponds the variable x_i in the k chromosome of the x_i populations can be expressed as:

$$q_{k,i}^t = \begin{Bmatrix} \alpha_{k,i}^t \\ \beta_{k,i}^t \end{Bmatrix} \tag{3}$$

So a complete quantum bit chromosome coding of the problem is:

$$q_k^t = \left\{ \begin{array}{c|c|c|c|c|c|c} \alpha_{k,1}^t & \alpha_{k,2}^t & \text{L} \quad \text{L} & \alpha_{k,i}^t & \text{L} \quad \text{L} & \alpha_{k,N}^t \\ \beta_{k,1}^t & \beta_{k,2}^t & \text{L} \quad \text{L} & \beta_{k,i}^t & \text{L} \quad \text{L} & \beta_{k,N}^t \end{array} \right\} \tag{4}$$

Obviously, the chromosome length is $L = N$. Set the population size is s , the generation qubit chromosome population of this problem is:

$$Q^t = \{q_k^t \mid k = 1,2,\text{L} \ ,s\} \tag{5}$$

2.2 Quantum Collapse

Observe the quantum bits of a design variable x_i of formula (4), it can get a code represented by the variable boundary. The specific method is: generate random numbers between 0~1 wrote $r_{k,i}^t$, if $r_{k,i}^t \le \left|\alpha_{k,i}^t\right|^2$, it is x_i^l state, or it is x_i^u state. Obviously, the collapse results of the qubit of any variable x_i are:

$$vb_{k,i}^t \in \{x_i^l, x_i^u\} \tag{6}$$

Therefore, a complete variable boundary code representation problem may be the following form:

$$vb_k^t = \left\{ x_1^l \middle| x_2^u \middle| \text{L} \quad \text{L} \ \middle| x_i^l \middle| \text{L} \quad \text{L} \ \middle| x_N^u \right\} \tag{7}$$

The T generation of variable boundary coding chromosome population can be expressed as:

$$VB^t = \{vb_k^t \mid k = 1,2,\text{L} \ ,s\} \tag{8}$$

2.3 Code Rule

From the formula (6), boundary representation of variable x_i is one of the two cases, two ranges corresponding to the variable, as shown in figure 1. If $\Delta x_i = x_i^u - x_i^l$, the size of each area is $\Delta x_i / 2$, the decoding rule in these two cases of variable x_i is:

(1) If $vb_{k,i}^t = x_i^l$, x_i is the value of lower limit, it is shown in Figure 1 (a).

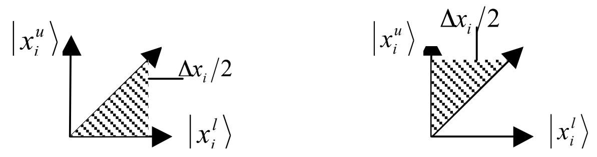
$$x_i = x_i^l + (1 - r) \cdot \Delta x_i / 2 \tag{9}$$

In the formula, r is a random number between 0~1, hereinafter the same.

(2) If $vb_{k,i}^t = x_i^u$, x_i is the value of upper limit, it is shown in Figure 1 (b).

$$x_i = x_i^u - (1 - r) \cdot \Delta x_i / 2 \tag{10}$$

Your goal is to simulate the usual appearance of papers in a Journal of the Academy Publisher. We are requesting that you follow these guidelines as closely as possible.



(a) Lower limit code

(b) Upper limit code

Fig. 1. Schematic diagram of decoding rule

2.4 Quantum Gate Adjustment Strategy

Table 1 is the dynamic selection strategy of the rotation angle of quantum gates [4]. In the table x and b are the current solution and the optimal solution by previous generation, x_i and b_i are the i variables boundary coding of homographic solution, e is the average of upper and lower limit value of the i variables, $f(x)$ and $f(b)$ are the corresponding fitness value. Similarly, when $f(x) \ge f(b)$, the solution x is superior to solution b . In order to make the quantum rotation gate guide solution to approach better direction, according to selection strategy of Table 1, the rotation angle value is $\theta = s(\alpha_i, \beta_i) \cdot \Delta\theta$.

3. Quantum Genetic Algorithm Programming

Computer technology applied to quantum genetic algorithm in this paper is Java language, programming environment is JBuilder 2005. JBuilder 2005 is a powerful Java enterprise platform, inherites almost all the Java technology, covers all the process of software development life cycle. Java is the most commonly used computer programming language, it is also one of the main development language network. The Java has the advanced computing language characteristics of object oriented, distributed and multi thread, but also it is becoming an industry standard because of portability, high safety and mobility. JBuilder is a Java technology followed the new technology standards. Because the JBuilder and several separate sources code processing products are integrated together, so it can support multiple Java development kit versions [11]. The system structure diagram of visualization operating system developed in this paper is shown below in Figure 3.

After the program has run, it appears a visual interface; select optimization and optimization algorithm in visual interface, and enter the necessary control parameters, and then optimize it, finally output optimal solution in the visual interface.

In the program, mainly designs following 7 categories, the main function and class relations are as follows:

(1) Main Class: Main program class, it is the entrance of the whole process, including 1 main method: public static void main(String[] args)

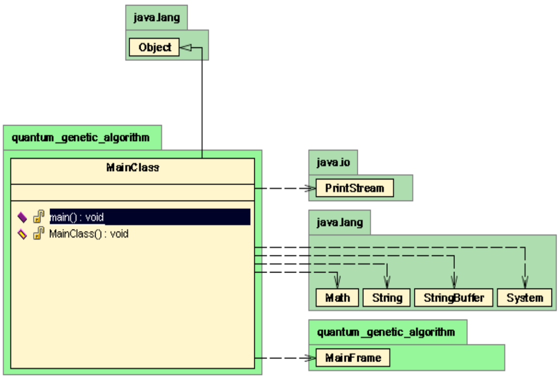


Fig 2. UML model diagram of MainClass

Table 1. Selection strategy of rotation angle

x_i	b_i	$f(x) \geq f(b)$	$x_i > e$	$b_i > e$	$\Delta\theta$	$s(\alpha_i, \beta_i)$			
						$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
x_i^l	x_i^l	false			0				
x_i^l	x_i^l	true			0				
x_i^l	x_i^u	false	true		0.05 π	+1	-1	0	±1
			false	true	0.25 π				
				false	0.05 π				
x_i^l	x_i^u	true	true		0.05 π	-1	+1	±1	0
			false	true	0.25 π				
				false	0.05 π				
x_i^u	x_i^l	false		true	0.05 π	-1	+1	±1	0
			true	false	0.25 π				
			false		0.05 π				
x_i^u	x_i^l	true		true	0.05 π	+1	-1	0	±1
			true	false	0.25 π				
			false		0.05 π				
x_i^u	x_i^u	false			0				
x_i^u	x_i^u	true			0				

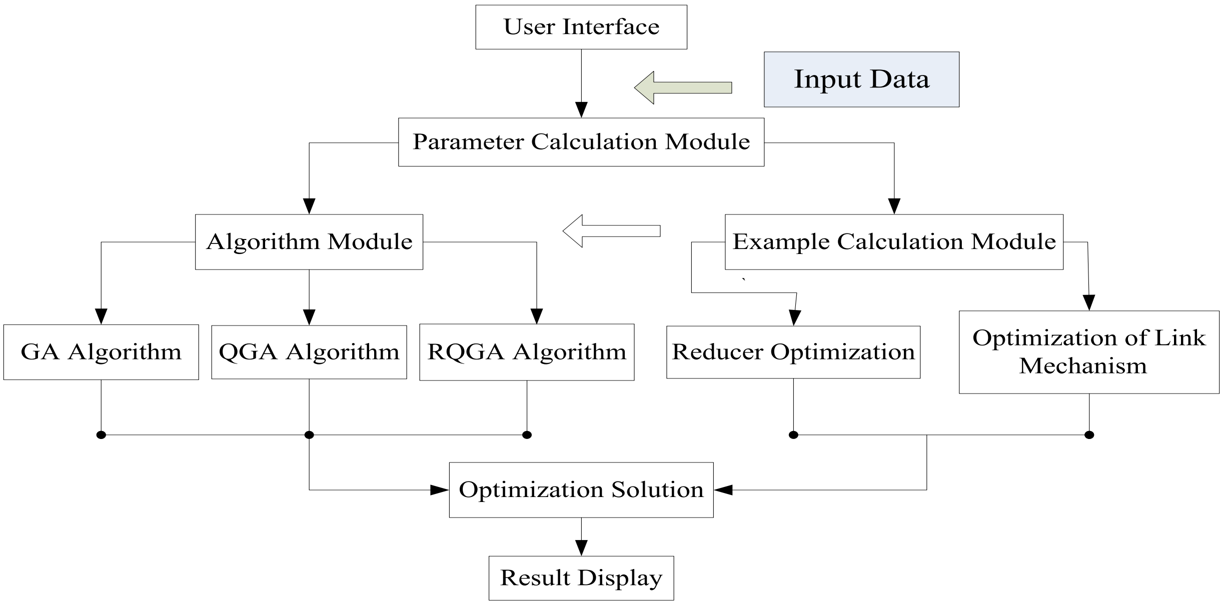
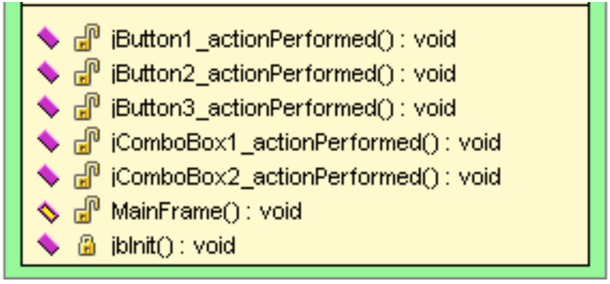


Fig. 3. The system structure diagram

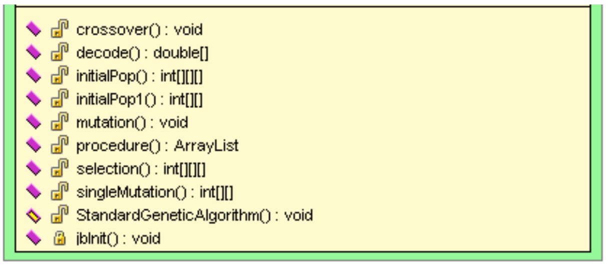
(2) MainFrame Class: Its main functions include: implementation of visual interactive interface and generation of optimization results, the class of various optimization algorithms can be called directly by this class, and output the optimal values of simulation and the simulation time. In this class, there are 6 main methods:



```
jButton1_actionPerformed() : void
jButton2_actionPerformed() : void
jButton3_actionPerformed() : void
jComboBox1_actionPerformed() : void
jComboBox2_actionPerformed() : void
MainFrame() : void
jbInit() : void
```

Fig 4. UML model diagram of MainFrame class

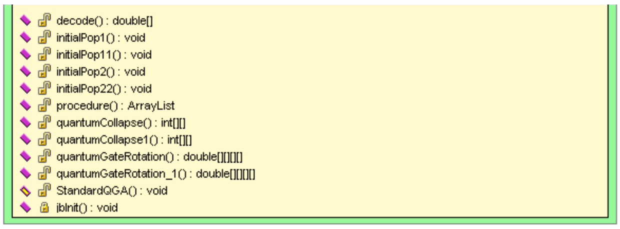
(3) StandardGeneticAlgorithm Class: It contains 9 methods, to achieve the population initialization, decoding operation and the selection, crossover and mutation of three kinds of genetic operation;



```
crossover() : void
decode() : double[]
initialPop() : int[][][]
initialPop1() : int[][]
mutation() : void
procedure() : ArrayList
selection() : int[][][]
singleMutation() : int[][]
StandardGeneticAlgorithm() : void
jbInit() : void
```

Fig 5. UML model diagram of StandardGeneticAlgorithm class

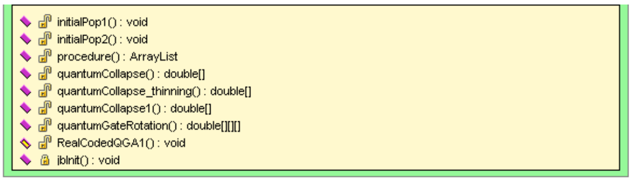
(4) StandardQGA Class: It contains 11 methods, to achieve the population initialization, decoding operation, the quantum collapse and spin quantum gate operation ;



```
decode() : double[]
initialPop1() : void
initialPop11() : void
initialPop2() : void
initialPop22() : void
procedure() : ArrayList
quantumCollapse() : int[][]
quantumCollapse1() : int[][]
quantumGateRotation() : double[][][][]
quantumGateRotation_1() : double[][][][]
StandardQGA() : void
jbInit() : void
```

Fig 6. UML model diagram of StandardQGA class

(5) RealcodedQGA Class: It contains 8 methods, to achieve the population initialization, the quantum collapse and spin quantum gate operation;



```
initialPop1() : void
initialPop2() : void
procedure() : ArrayList
quantumCollapse() : double[]
quantumCollapse_thinning() : double[]
quantumCollapse1() : double[]
quantumGateRotation() : double[][][]
RealCodedQGA1() : void
jbInit() : void
```

Fig 7. UML model diagram of StandardQGA class

(6) HelicalCylinderGearReducer Class: It contains 5 methods, to store upper and lower limit of variables of the gear reducer, to calculate constraint conditions, the initial value of penalty factor, the reduction coefficient of penalty factor and the fitness function;

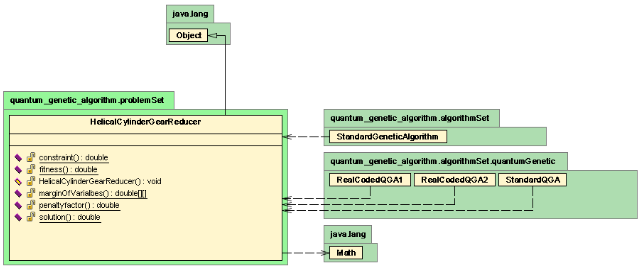

java.lang
Object
quantum_genetic_algorithm.problemSet
HelicalCylinderGearReducer
constraint() : double
fitness() : double
HelicalCylinderGearReducer() : void
marginOfVarialbes() : double[][]
penaltyfactor() : double
solution() : double
quantum_genetic_algorithm.algorithmSet
StandardGeneticAlgorithm
quantum_genetic_algorithm.algorithmSet.quantumGenetic
RealCodedQGA1
RealCodedQGA2
StandardQGA
java.lang
Math

Fig 8. UML model diagram of HelicalCylinderGearReducer class

(7) FourBarLinkage Class: It contains 6 methods, to store upper and lower limit of variables of four bar mechanism, to calculate constraint conditions, the initial value of penalty factor, the reduction coefficient of penalty factor and the fitness function;

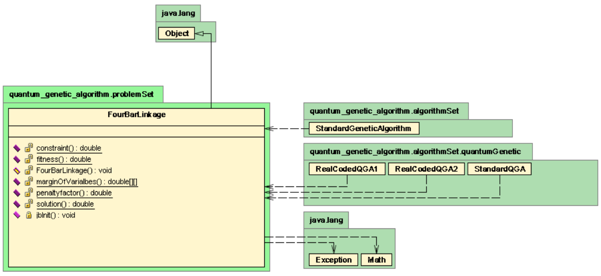


Fig 9. UML model diagram of FourBarLinkage class

After the program has run, it appears a visual interface Figure 10, in the window, it can set up the optimization, optimization algorithm, the population size, generation of initial population, crossover probability, mutation probability and whether the elitist strategy is adopted, then show the results. Taking the single pole six peak humpback function solving process as an example:

$$F = 10 + \frac{\sin(1/x)}{0.1 + (x - 0.16)^2}, \quad 0 \le x \le 1 \tag{11}$$

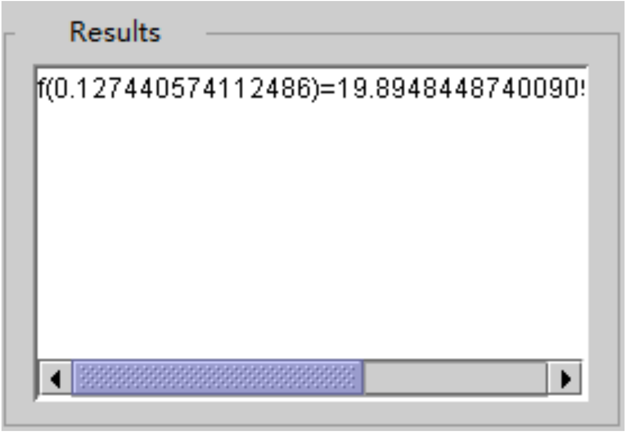
The specific operation process is as follows:

(1) Select the single pole six peak humpback function selection problem in a drop-down menu;

(2) Select the basic quantum genetic algorithm in algorithm selection drop-down menu;

(3) Do not use optimal preservation strategy, the others maintain the initial setup;

(4) Click to optimize, when operation is completed, click to save results;



```
f(0.127440574112486)=19.8948448740090
```

Fig 10. Show the results

(5) Click to draw trend, the curve of each generation convergence is shown in Figure 11.

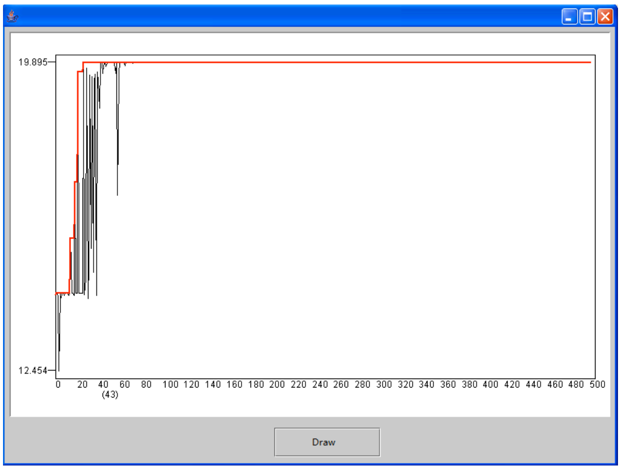


Fig 11. Draw the results

Based on the quantum computing, it introduces the basic principle of quantum genetic algorithm, and combines the single quantum bit code and real coding, gives the basic steps of real coded quantum genetic algorithm, lays algorithm foundations for optimal design of mechanical optimization examples.

4. Optimization Example Of Gear Reducer

Gear reducer has a very wide range of applications in all walks of life, it is an indispensable mechanical transmission device. The reducer has many problems like large volume, large weight, or the transmission ratio and low mechanical efficiency. In order to meet the requirements of saving energy and reducing consumption, reducer provided today must develop to high power, large transmission ratio, small volume, high mechanical efficiency and long service life. Therefore, in addition to the continuous improvement of material quality, raise the level of technology, using advanced optimization methods to optimize the design of gear reducer is also very necessary [5].

As shown in Figure 12, two grade helical cylindrical gear reducers, high speed shaft input power is 6.2*kw*, high speed shaft speed is 1450*r/min*, transmission ratio is 31.5, tooth width coefficient is 0.4, gear materials and heat treatment of large gear 45 steel normalizing is HB=187-207, small gear 45 steel tempered is HB=228-255. Total worked time is not less than 10 years. It is required to determine the main parameters of the overall scheme according to the minimum center distance.

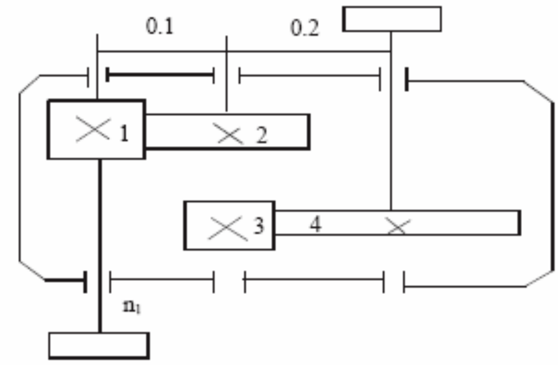


Fig 12. Two grade helical cylindrical gear reducer

5. Establish The Optimization Model

5.1 Optimization Objective

Select center distance of the reducer as the optimization objective, specific calculation formula is as follows:

$$f(x)=\sum a=a_1+a_2\frac{m_{n1}z_1(1+i_1)+m_{n2}z_3(1+i_2)}{2\cos\beta} \tag{12}$$

In the formula, m_{n1} -High-speed level gear surface normal modulus

m_{n2} -Low-speed level gear surface normal modulus

i_1 -High-speed level transmission ratio

i_2 -Low-speed level transmission ratio

z_1 -High-speed level small gear tooth number

z_3 -Low-speed level small gear tooth number

β -Helical gear

5.2 Design Variable

In this case, the independent parameters to calculate the total center distance are m_{n1} , m_{n2} , z_1 , z_3 , i_1 (obviously $i_2=31.5/i_1$), and because $i_1=z_2/z_1$, in order to facilitate the processing, the design variables are:

$$X=[m_{n1},m_{n2},z_1,z_3,z_2,\beta]^T=[x_1,x_2,x_3,x_4,x_5,x_6]^T$$

From the transfer power and speed it can be estimated: $2\le m_{n1}\le 5$, standard values are (2,2.5,3,4,5); $2\le m_{n2}\le 6$, standard values are (3,4,5,6). Consider comprehensively various factors, such as steady transmission, the axial force cannot be too large, meeting the short-term overload, immersion depth of high level and low level big gear is approximately the same, gear pitch circle size is not too small, take them as $14\le z_1\le 22$, $16\le z_3\le 22, 82\le z_2\le 154$, $8^0\le\beta\le 15^0$.

5.3 Constraint Condition

In this case, consider the conditions comprehensively like tooth surface contact strength, bending strength and high speed large gear and a low speed shaft cannot generate interference, set the relevant parameters, we can get the following constraints [8]:

$$\begin{cases}
g_1=3.098\times10^{-6}x_1^3x_3^2x_5-\cos^3x_6\ge 0\\
g_2=1.017\times10^{-4}x_2^3x_4^3-x_5^2\cdot\cos^3x_6/x_3^2\ge 0\\
g_3=9.939\times10^{-5}\cdot(1+x_5/x_3)\cdot x_1^3x_3^2-\cos^3x_6\ge 0\\
g_4=1.076\times10^{-4}\cdot(31.5+x_5/x_3)\cdot x_2^3x_4^2-x_5^2\cdot\cos^2x_6/x_3^2\ge 0\\
g_5=x_2x_4\cdot(31.5+x_5/x_3)-x_5/x_3\cdot[2(x_1+50)\cdot\cos x_6+x_1x_5]\ge 0\\
g_6=1.116\times10^{-4}\cdot(1+x_5/x_3)\cdot x_1^3x_3^2-\cos^2x_6\ge 0\\
g_7=1.171\times10^{-4}\cdot(31.5+x_5/x_3)\cdot x_2^3x_4^2-x_5^2\cdot\cos^2x_6/x_3^2\ge 0
\end{cases} \tag{13}$$

Because g_6 and g_7 is passive constraint obviously compared with g_3 and g_4 , so they can be omitted. Take five constraint conditions.

5.4 Optimization Model

From the above content, this optimization examples is constrained optimization problem, but genetic algorithm, quantum genetic algorithm and real coded quantum genetic algorithms only solve the unconstrained optimization problems, so we must adopt some certain methods to change the example of the constrained optimization problem into unconstrained optimization problem, and then optimize the calculation.

Internal penalty function method is referred to the interior point method[6], this method defines a new objective function into the feasible region, the sequence iterative points in the feasible region is approaching gradually to the optimum solution on the constrained boundary. The internal point method can only be used to solve the optimization problem with inequality constraints. That is only inequality constrained optimization problem.

$$\begin{cases}\min f(x)\\ s\cdot t\qquad g_j(x)\le 0,j=1,2,\cdots m\end{cases} \tag{14}$$

Translate it into

$$\Phi\ (x,r)=f(x)-r\sum_{j=1}^{m}\frac{1}{g_j(x)} \tag{15}$$

Or

$$\Phi\ (x,r)=f(x)-r\sum_{j=1}^{m}\ln[-g_j(x)] \tag{16}$$

In the formula r is the penalty factor, it is a sequence from big to small and close to 0, that is $r^0>r^1>r^2>\cdots>r^n\to 0$.

The effect of penalty factor is: because internal point method only iterations in the feasible region, but the optimal solution is likely to be near the border of the feasible region or on its boundary, now although the value of functional is very large, because the penalty factor is a positive value continuing decline, so after many iterations and close to the optimal solution, the penalty term is a very small positive, so the transformed unconstrained optimization problem comes down to the same optimal the solution of the original constrained problem.

Using penalty function method to establish the optimization model is shown in a formula (17) [2].

$$\begin{cases}
\min:\quad \Phi(x,r)=f(x)+r\sum_{j=1}^{5}\frac{1}{g_j(x)}\\
\qquad g_1=3.098\times10^{-6}x_1^3x_3^2x_5-\cos^3x_6\ge 0\\
\qquad g_2=1.017\times10^{-4}x_2^3x_4^3-x_5^2\cdot\cos^3x_6/x_3^2\ge 0\\
s.t.:\quad g_3=9.939\times10^{-5}\cdot(1+x_5/x_3)\cdot x_1^3x_3^2-\cos^3x_6\ge 0\\
\qquad g_4=1.076\times10^{-4}\cdot(31.5+x_5/x_3)\cdot x_2^3x_4^2-x_5^2\cdot\cos^2x_6/x_3^2\ge 0\\
\qquad g_5=x_2x_4\cdot(31.5+x_5/x_3)-x_5/x_3\cdot[2(x_1+50)\cdot\cos x_6+x_1x_5]\ge 0
\end{cases} \tag{17}$$

In this way the volume of speed reducer can be optimized by SGA, QGA and RQGA.

6. Optimization And Results Analysis Of SGA, QGA And RQGA

This section solves the optimization problem respectively with the standard genetic algorithm, quantum genetic algorithm and real coded quantum genetic algorithm, and the results were statistically analyzed and compared [7].

6.1 Control Parameters And Control Strategy

Operation parameters the genetic algorithm need to select is main group size M, crossover probability and mutation probability, the selection of parameters has great influence on genetic algorithm [11].

(1) Main group size M: Generally values range between10~160. When the M is small, although it can improve the operation speed of genetic algorithm can improve the speed of operation, but reduces the diversity of the population, easily causes the premature phenomenon of genetic algorithm; when the M is big, it will seriously affect the operation efficiency of genetic algorithm.

(2) Crossover probability p_c : Crossover operation is the main method of genetic algorithm to generate a new individual, frequency of use, and the crossover probability control the frequency of crossover operation use, so generally cross probability is large. But if the value is too large, it will destroy the good model of the group, is not conducive to the evolution operation; if the value is too small, it will reduce the generation speed of new individuals, making the algorithm into a torpid state. Generally values range between0.4~0.99. In addition, it can also ensure the crossover probability according to the adaptive thought.

(3) Mutation probability p_m: The mutation operator in genetic algorithm is the auxiliary method to produce new individuals, so, if this probability is too big, it can destroy the good model, it is a performance of genetic algorithm similar to the performance of a random search algorithm; if the mutation probability is too small, it will reduce its ability to produce new individuals and inhibition of premature phenomenon. Generally values range between0.0001~0.1. It can also determine the mutation probability according to the adaptive thought.

(4) Optimum reserved strategy: No matter whether the current best individual fitness is more than the fitness of the best individual so far, uses with the best individual so far to replaces the current worst individual current.

(5) Termination generation T: It is an important parameter to control the end conditions of running the genetic algorithm, namely algorithm runs this algebra then stops, and taking the best individual in the group as the optimal solution of the problem to output. Generally values range between100~1000. Of course, we can also use some kinds of judgment criteria to the control the end of the algorithm. Criteria commonly used are:

a) The differences of average fitness of successive generations individuals is less than a minimum threshold;

b) The variance of the individual fitness of all the members is less than a minimum threshold.

6.2 Optimization

Standard genetic algorithm uses binary coding, roulette wheel selection algorithm, single point crossover and gene 0-1 mutation, among them the population size M=10, crossover probability p_c =0.5, mutation probability p_m =0.01, it uses the optimal preservation strategy, and takes the maximum iterations number T=500 as a condition of convergence.

Quantum genetic algorithm and real coded quantum genetic algorithm all generate initial code randomly, use the optimum retention mechanism to update the chromosome, take the maximum iterations number T=500 as convergence condition, and the population size M=10.

The program uses the JAVA language, respectively uses the standard genetic algorithm, quantum genetic algorithm and real coded quantum genetic algorithm to optimize 20 times, the optimal solutions are shown in Table 2 [9].

Table 2. The optimal solution of three algorithm

optimization algorithm / Parameter	SGA	QGA	RQGA
m_{n1}	2.0	2.0	2.0
m_{n2}	4.0	3.0	3.0
Z_1	20.0	21.0	21.0
Z_3	16.0	18.0	18.0
Z_2	100.0	87.0	87.0
β	12.0	13.0	13.0
$F(X)$	361.5	349.2	349.2

The optimal solution of the three optimization algorithm is shown in Figure 13.

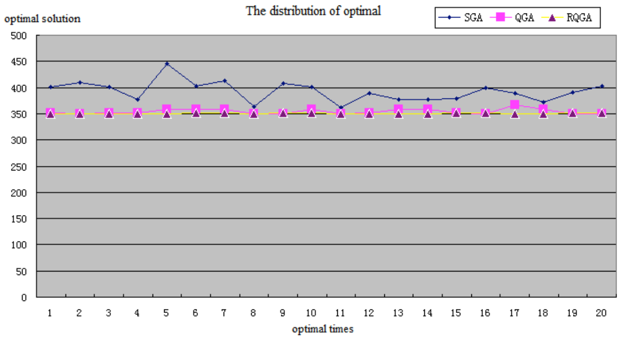


Fig 13. The distribution of optimal solution

6.3 Results Analysis

Statistics the results, it is shown in Table 3.

Table 3. Statistical results of running 20 times

optimization algorithm	SGA	QGA	RQGA
mean value	392.517	353.865	350.527
optimal value	361.5	349.244	349.244
standard deviation	4.386	2.181	1.007
mean time/20 times	2.5734 s	17.349 s	0.4797 s

By analyzing the statistical results, the conclusions are as follows:

(1) Optimal value: The optimal solution values of RQGA and QGA are equal, they are all better than SGA, it shows that the quantum genetic algorithm has better search ability;

(2) Average value: Run 20 times, the average value of RQGA is better than QGA, the average value of QGA is better than SGA;

(3) Standard deviation: Run 20 times, the standard deviation of RQGA is better than QGA, the standard deviation of QGA is better than SGA. So to the stability of solution RQGA is better than QGA, and QGA is better than SGA;

(4) Average calculating time: Run 20 times, To the average speed, RQGA is faster than SGA, SGA is faster than QGA, it shows that using the real coded can avoid the encoding and decoding operations, and save operation time.

7. Conclusions

Based on quantum genetic algorithm, this paper proposes a new real coded quantum genetic algorithm, by comparing and analyzing the optimization results of genetic algorithm and quantum genetic algorithm, it shows that real coded quantum genetic algorithm has better stability of solution and solving speed. In the real coded quantum genetic algorithm, it uses real number coding, has the characteristics of omitting the process of encoding and decoding, high accuracy, then applies it to two grade helical cylindrical gear reducer for optimizing, greatly meets the design requirements. Optimize the two grade helical gear reducer volume with standard genetic algorithm, basic quantum genetic algorithm and real coded quantum genetic algorithm, its results show that the optimal solution, the solution speed and the solution stability of real coded quantum genetic algorithm solution is better than the standard genetic algorithm and basic quantum genetic algorithm.

Acknowledgement

This research reported in the paper is supported by Key Laboratory of Metallurgical Equipment and Control of Ministry of Education in Wuhan University of Science and Technology (2013A04) and Hubei Provincial Department of Education in China (D20121102). This support is greatly acknowledged.

References

1. HAN K H, KIM J H, “Quantum-inspired evolutionary algorithm for a class of combinational optimization”, IEEE Trans Evolutionary Computation, vol.6, no.6, 2002, pp.580-593.
2. MIKKI S M, KISHK A A, “Quantum particle swarm optimization for electromagnetics”, IEEE Transactions on Antennas and Propagation, vol.54, no.10, 2006, pp.2764-2775.
3. YUAN Yun-long, “Optimization design of elbow-bar mechanism in injection molding machine based on genetic algorithm”, Journal of Mechanical & Electrical, Engineering, vol.26, no.4, 2009, pp.22-24.
4. WANG Xiaoqin, WANG Wanliang, XU Xinli, “Hybrid quantum inspired evolutionary algorithm for flowshop scheduling problem”, Journal of Mechanical & Electrical Engineering, vol.26, no.9, pp.5-8.
5. WU Bo, GAO Guibing, DING Yufeng , “The Design of Three-dimension Parametric for Reducer Based on SolidWorks”, Journal of Mechanical & Electrical Engineering, vol.22, no.9, 2005, pp.21-24.
6. YI Shi-min, CHEN Hui-kun, WEN Fu-shuan, “Online analysis and optimization of the generation cost in guangdong power grid”, Journal of Mechanical & Electrical Engineering, vol.26, no.7, 2009, pp.49-52.
7. ZHANG Libing, WU Ting, YUAN Genfu, YAN Xingshu, “Path optimization on laser drilling based on genetic algorithm”, Journal of Mechanical & Electrical Engineering, vol.24, no.6, 2007, pp.77-79.
8. Da-lian Liu, Xiao-hua Chen, Jin-ling Du, “A Hybrid Genetic Algorithm for Constrained Optimization Problems”, Journal of Computers, vol.8, No.2, 2013.
9. Jinfeng Wang, BiQiang Du, HaiMin Ding, “A Modified Genetic Algorithm (GA) for Optimization of Process Planning”, Journal of Computers, vol.6, No.7, 2011.
10. Lijuan Zhou, Xiaoxu He, Kang Li, “An Improved Approach for Materialized View Selection Based on Genetic Algorithm”, Journal of Computers, vol.7, No.7, 2012.
11. Binod Kumar Pattanayak, Sambit Kumar Patra, Bhagabat Puthal, “Optimizing AST Node for Java Script Compiler A lightweight Interpreter for Embedded Device”, Journal of Computers, vol.8, No.2, 2013.
12. Sheng Liu, Xiaoming You, Zhong Wu, “A Cultural Immune Quantum Evolutionary Algorithm and Its Application”, Journal of Computers, vol.8, No.1, 2013.
13. Sergey Anfilets, Vasili Shuts, “The use of natural optimization algorithms for the implementation of adaptive control at the crossroad”, Transport and Telecommunication Institute, vol.6, no.12, 2012, pp.227-233.