Journal of Computational Science & Engineering 6 (2013) 255-259

Journal of Computational Science & Engineering

Available at http://www.asocse.org ISSN 1710-4068

Solving Methods To A Class of Nonlinear Ip Problem Based on Quantum Genetic Algorithm

Jianqiang Zhao[a]*, Bike Chen[a] and Qingsong Dai[b]

[a] *School of Mathematic and Physical Science ,Xuzhou institute of technology*
[b] *Xuzhou institute of technology ,Xuzhou, Jiang Su 221111,China*

Article Information

Article history:
Received 18 March 2013
Revised 15 April 2013
Accepted 20 April 2013
Available online 10 May 2013

Keywords:
Nonlinear problems
Quantum genetic algorithm
Intelligent algorithm

Abstract

This paper constructs a new intelligent algorithm for solving nonlinear problems which is Quantum Genetic hair. In this paper ,it gives the specific steps and associated quantum genetic algorithm code Validated by an example .The results show quantum genetic algorithm for solving nonlinear problem is very effective and practical.

1. Introduction

We often encounter the following issues difficult to deal with, in automatic control, system identification, filtering design engineering practice and scientific research and other fields.

$$\min_{x\in R^n} f(x)=\sum_{i=1}^{m}\left|f_i(x)\right|^p (p>0) \quad , \qquad (1)$$
$$s.t.\ g_j(x)\le 0\ (j=1,2,\cdots,s)$$

Where $f_i(x), i\in I=\{1,2,\cdots,m\}$ is continuously differentiable function, and at least one non-linear. At present, there are many results for solving this problem. Literature [1] takes advantage of multi-objective planning and maximum entropy methods to study on this function, assuming $0<p\le 1$;

Also assuming $0<p\le 1$, literature [2] [3] [4] use the adjustable entropy method to study on such issues, since the variable is not restricted; Using adjustable interval extension of the entropy function and relevant regional delete testing principles ,Literature [5] solves such problems.

In recent years, with the development of

* Corresponding author: zjqghh@xzit.edu.cn

2013 © The American Computational Science Society. All rights reserved.

computer technology, some groups have appeared intelligent optimization algorithm, for example the literature [6] [7]. Among them, the literature [6] uses particle swarm algorithm to solve nonlinear norm minimization problem. However, this method has a drawback that solution accuracy is not high. In this paper, we use quantum genetic intelligent algorithm to solve the above problems, assuming $0 < p$. The advantages of this algorithm is speed and high accuracy. Simulation results show that the algorithm in solving such problems is practical and effective.

2. Quantum Genetic Algorithm

2.1 Quantum Algorithm

Quantum computing uses quantum states as the basic unit of information, making full use of quantum superposition, entanglement and interference and other characteristics. It solves the classical calculation NP-hard problems by quantum parallel computing. In 1994, SHOR presented the first quantum algorithm for solving the prime factorization of large numbers of classical computational problems. The algorithm can be used for public-key systems RSA. In 1996, Grover proposed quantum random database search algorithm.

The algorithm can be realized in a quantum computer database for failure to add integer $\sqrt{N}$ magnitude speed up the search. Since then, quantum computing rapidly becomes a research hotspot.

2.2. Genetic Algorithms

Genetic algorithm is the computing model which simulating Darwinian theory of evolution by natural selection and Mendel genetics of biological evolution.

It is an optimal solution search methods by simulating natural evolutionary process. Genetic algorithms are not constrained by nature of the problem, optimization criteria, etc. It uses objective function to globally adaptive search under the guidance of the probability. It is able to solve complex problems. Traditional optimization methods are difficult to solve these problems. However, this method is prone to premature population, slow convergence

2.3 Quantum Genetic Algorithm

Since parallel processing capabilities of quantum algorithm can compensate for the shortcomings of GA, many scholar have suggested combining the two algorithms and get quantum genetic algorithm. Quantum genetic algorithm[7][10] uses the probability amplitude expressed qubit chromosome coding, Using quantum superposition state revolving door on the role as the evolution operator based on the principle of quantum evolution. Quantum genetic algorithm has better population diversity and parallel computing capabilities.

2.4 Quantum genetic algorithm steps (combined herein example)

Step1: Quantum Coding

Coded quantum genetic algorithm combines quantum algorithm and genetic algorithm encoding.

Using genetic algorithm binary coding, each component can be an independent variable to represent quantum genes, such as available:

$$\begin{bmatrix} \alpha_{i1} & \alpha_{i2} & \cdots & \alpha_{ik} \\ \beta_{i1} & \beta_{i2} & \cdots & \beta_{ik} \end{bmatrix}$$

.

Each gene consists of a number of quantum bits qubits, each qubit has two states. Respectively, with a certain probability that $|\alpha_{ij}|^2$ and $|\beta_{ij}|^2$ are taken to 0 state and the 1 state. For the existence of multi-state problem encoding, each individual available quantum quantum encoded form as follows:

2013 © The American Computational Science Society. All rights reserved.

$$q_j=\begin{bmatrix}\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1k} & \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2k} & \cdots & \alpha_{n1} & \alpha_{n2} & \cdots & \alpha_{nk}\\ \beta_{11} & \beta_{12} & \cdots & \beta_{1k} & \beta_{21} & \beta_{22} & \cdots & \beta_{2k} & \cdots & \beta_{n1} & \beta_{n2} & \cdots & \beta_{nk}\end{bmatrix}$$

Quantum code is corresponding to initial population in the specific example of solving. In the initialization, let $\alpha_{ij}=\beta_{ij}=1/\sqrt{2}$, Binary encoding length of each variable is k. To numerical examples in this article, let k=20.

Step2：Finding the fitness value of the population

In the design of the fitness function, we make $-f(x)$ as fitness function. After the above test population into the fitness function, we can find the fitness function values.

Step3：Quantum revolving door

In quantum genetic algorithm, we usually use the revolving door to quantum information to operate. The process is as follows:

$$\begin{bmatrix}\alpha'_i\\ \beta'_i\end{bmatrix}=\begin{bmatrix}\cos(\Delta\theta_i) & -\sin(\Delta\theta_i)\\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i)\end{bmatrix}\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix},$$

$\Delta\theta$'s amplitude affects the convergence rate.If the amplitude is too large, it will lead to precocious. $\Delta\theta$'s symbol affects the quantum-rotation direction, specifically as follows:

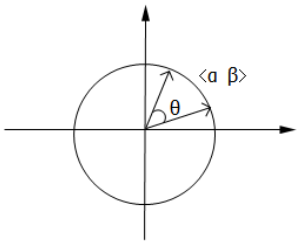

Fig. 1 Quantum plot revolving door

When the quantum bit is in the first and third quadrant, $\Delta\theta_i$ is set to positive (negative),in order to increase the value "1" ("0") the probability; When the quantum bit is in the second, and fourth quadrant, $\Delta\theta_i$ is set to negative (positive),in order to increase the value "0" ("1") the probability.

Table 1-1 Rotation angle selection strategy

x_i	$best_i$	$f(x_i)>f(best_i)$	$\Delta\theta_i$
0	0	*false*	0
0	0	*true*	0
0	1	*false*	0.01π
0	1	*true*	0.01π
0	0	*false*	0.01π
0	0	*true*	0.01π
0	1	*false*	0
0	1	*true*	0

Table 1-2 Rotation angle selection strategy

$s(\alpha_i,\beta_i)$			
$\alpha_i\beta_i>0$	$\alpha_i\beta_i<0$	$\alpha_i=0$	$\beta_i=0$
0	0	0	0
0	0	0	0
+1	-1	0	+1
-1	+1	+1	0
-1	+1	+1	0
+1	-1	0	+1
0	0	0	0
0	0	0	0

Where x_i is the current bit i chromosome; $best_i$ is a current optimal chromosome bit i; $f(x_i)$ is the fitness function; $s(\alpha_i,\beta_i)$ is the angle of rotation direction； $\Delta\theta_i$ is the rotational angular size. According to the choice of strategy of table 1, we will present the individual's fitness value $f(x_i)$ and populations to adapt to the current best individual value $f(best_i)$ Compare. If $f(x_i)>f(best_i)$, We will adjust the current individual corresponding qubits, making probability amplitude for (α_i,β_i) favor x_i appeared toward the direction of evolution; If $f(x_i)>f(best_i)$, We will adjust the current individual corresponding qubits, making probability amplitude for (α_i,β_i) favor $best_i$ appeared toward the direction of evolution.

3. Matlab program design for Quantum genetic algorithm

The following, we design quantum genetic algorithm procedures for solving equations (1). Part

2013 © The American Computational Science Society. All rights reserved.

of the code is as follows:

1）Initialing population and the parameters of $Q(t_0)$.

MAXGEN=200; sizepop=40; lenchrom=[20,20];

chrom=Initpop (sizepop*2,sum(lenchrom));

2）Testing each individual of $Q(t_0)$

Binary=collapse (chrom)

3）Evaluating the fitness of each individual of $Q(t_0)$

[fitness, X]=fitnessfunction (binary, chrom);

4）Let the best individual goals as the next generation evolution

[best. fitness bestindex]=max(fitness);

5）While（i<=MAXGEN）do

Initialing population and the parameters of $Q(t_0)$

(Program code above)

Evaluating the fitness of each individual of $Q(t_0)$

（Program code above）

Updating quantum gate populations, get the next generation population $Q(t+1)$

Chrom = Qgate (chrom, fitness, best, binary);

Recording the best individual and the corresponding fitness value

If new best fitness>best. fitness

best fitness=new best fitness;

best binary=binary(new best index,:);

best chrom=chrom(2*newbestindex-1:2*new best index);

best X=X(new best index)

End

Update iteration：$t = t + 1$

i=i+1

$end \quad while$

6） Returning Q_{last} (last generation) in the best fitness value

4. Numerical examples

For the numerical examples in the literature [4], we let $p = 1, 0.3, 3$:

$$\min_{x \in R^n} f(x) = \sum_{i=1}^{m} \left| f_i(x) \right|^p \ (p > 0)$$

$$s.t.\ g_j(x) \le 0 \ (j = 1, 2, \cdots, s)$$

,

where：

$$f_1(x) = \cos x_1, f_2(x) = -\sin x_2$$

$$f_3(x) = x_1^2 + x_2^2 - x_1 x_2 - 1$$

$$g_1(x) = x_1 - x_2 \le 0$$

$$g_2(x) = {x_1}^2 - x_2 \le 0$$

.

The results：

Table 2 $p = 0.3$

Iterations	x_1	x_2	F
100	-0.607462	0.546705	1.765
300	-0.621243	0.532316	1.75722
500	-0.623721	0.529703	1.75444
1000	-0.623709	0.529715	1.75417

Table 3 $p = 1$

Iterations	x_1	x_2	F
50	-0.682328	0.465571	1.22504
100	-0.682328	0.465571	1.22504
300	-0.682328	0.465571	1.22504

2013 © The American Computational Science Society. All rights reserved.

Table 4 $p = 3$

Iterations	x_1	x_2	F
50	-0.71377	0.509468	0.550252
100	-0.71377	0.509468	0.550252
300	-0.71377	0.509468	0.550252

5. Conclusions

The above example shows that quantum genetic algorithm is better to solve nonlinear lp problems. The above results are within the scope of the precise, compared with the literature[6]. So, Quantum Genetic intelligent algorithm is a better way for solving nonlinear problems.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No : 11001129) and Xuzhou institute of technology research funds(No : XKY2010201).

References

[1] X.H. Wang, X.Z. Qin, Computational Mathematics, 3 (1996) 305-308.

[2] R.P. Wang, Pure and Applied Mathematics Mathe-Matics, 4 (2004) 372-375.

[3] J.Q. Zhao, S.B. Li, Operations Research and Management, 5 (2011) 46-50.

[4] J.Q. Zhao, Journal of Xuzhou Institute of Technology, 5 (2012) 43-45.

[5] S.B. Li, Nanjing University Mathematical, 2 (2007) 376-385.

[6] L. Wang, Control and Decision, 2 (2005) 156-160.

[7] G.X. Zhang, W.D. Jin, Journalof Southwest Jiaotong University, 6 (2003) 717-722.

[8] Y.G. Cai, Systems Engineering Theory and Practice, 10 (2012) 2198-2206.

[9] N. Li, R.M. Wang, Henan Science, 12 (2012) 1742-1745.

2013 © The American Computational Science Society. All rights reserved.