International Journal of Computational Intelligence Systems, Vol. 5, No. 3 (June, 2012), 413-420

A Novel Real-coded Quantum-inspired Genetic Algorithm and Its Application in Data Reconciliation*

GAO Lin[1,2]
[1] *Research Institute of Automation, East China University of Science and Technology, NO.130 Meilong Road Xuhui District, Shanghai, 200237, China*
[2] *School of Automation and Electrical Engineering, Qingdao University of Science and Technology, NO.53 Zhengzhou Road Sifang District, Qingdao, 266042, China*

GU Xingsheng
Research Institute of Automation, East China University of Science and Technology, NO.130 Meilong Road Xuhui District, Shanghai, 200237, China
E-mail: xsgu@ecust.edu.cn, gaolin0619@126.com

Received 28 March 2011
Accepted 13 February 2012

Abstract

Traditional quantum-inspired genetic algorithm (QGA) has drawbacks such as premature convergence, heavy computational cost, complicated coding and decoding process etc. In this paper, a novel real-coded quantum-inspired genetic algorithm is proposed based on interval division thinking. Detailed comparisons with some similar approaches for some standard benchmark functions test validity of the proposed algorithm. Besides, the proposed algorithm is used in two typical nonlinear data reconciliation problems (distilling process and extraction process) and simulation results show its efficiency in nonlinear data reconciliation problems.

Keywords: quantum-inspired genetic algorithm, real-coded, interval division, nonlinear data reconciliation

1. Introduction

Due to the influence of many factors in the petrochemical manufacture process, the data obtained through chemical manufacture devices or measuring meters may be corrupted by kinds of noises, which impose great impacts on the decision-makings of control process. Therefore, the collected data should be preprocessed before been applied to manufacture process analysis and data reconciliation technique receives widespread applications in the petrochemical industry. The nonlinear data reconciliation has always been the research focus of data reconciliation techniques. Normally, there are two major directions, i.e. two-steps matrix projection method[1] and independent logistics based Simpson method[2]. Zhou *et al.*[3] proposed a modified outlier detection method to efficiently decrease the effect of outliers on the reconciled results through distinguishing the outliers of each variable individually and modifying the weight accordingly. LI and Rong[4] improved the efficiency and accuracy of mixed integer linear programming (MILP) approach by reducing the number of binary variables and giving accurate weights for suspected gross errors candidates.

GAO Lin, School of Automation and Electrical Engineering, Qingdao University of Science and Technology, NO.53 Zhengzhou Road Sifang District, Qingdao, 266042, China.
Corresponding Athor: Gu Xingsheng, Research Institute of Automation, East China University of Science and Technology, No.130 Meilong Road, Xuhui District, Shanghai, 200237, China.

Gao *et al.*[5] proposed a new nonlinear dynamic data reconciliation method to get high robustness and simple calculation by introducing a penalty function matrix in a conventional least-square objective function and assigning small weights for outliers and large weights for normal measurements. Mei *et al.*[6] overcame the defects of the nodal test (NT) and the measurement test (MT). Their method avoided some artificial manipulation and more than one gross error problems by combining NT and MT. Zhou *et al.*[7] solved efficiently gross error effect on data reconciliation by using several technologies including linearization method, penalty function, virtual observation equation, and equivalent weights method. Jiang *et al.*[8] proposed a new bias detection strategy which reduced greatly the number of parameters to be estimated and avoided sequential detections and iterations by detecting the presence of measurement bias and its occurrence time. As discussed above, although there are many nonlinear data reconciliation techniques, most of the existing approaches require complicated matrix calculation and transformation. Moreover, when the matrix cannot or is difficult to be solved, the data reconciliation process will become extremely complex. Fortunately, the evolutionary algorithms (EAs) provide new solutions to this problem, which can improve the optimization efficiency and avoid the complex matrix computation process. For instance, genetic algorithms (GAs) were applied into the data reconciliation and achieved satisfactory performance[9-11]. Although, GA ensures colony evolves and solutions change continually, it lacks a strong capacity of producing better offspring and causes slow convergence near global optimum, sometimes may be trapped into local optimum. In this paper, the concepts of quantum computing are adopted to improve the performance of GA.

The quantum-inspired evolutionary algorithms (QEAs)[12] are based on the principles of quantum computing, which can strike right balance between exploration and exploitation more easily when compared with conventional EAs. Meanwhile, the QEAs can explore search space with a smaller number of individuals and exploit global solution within a short span of time[13-15]. Due to the distinguished characteristics, several sub-branches appear in the recent years, i.e. Quantum-inspired Genetic Algorithm (QGA)[16], Quantum-inspired Immune Clone Algorithm (QICA)[17], and Quantum-inspired Particle Swarm Optimization (QPSO)[18]. QGA combines the advantages of quantum computing and GA, which is designed to address some intrinsic problems of genetic algorithms, and has been widely used in many fields[19-24]. Generally, QGA has the characteristics of small population size, fast convergence speed, and robust searching ability. However, in QGA, the chromosome is usually represented by binary code, which has the disadvantage of low computation efficiency due to the repeated encoding and decoding process.

As a sequence, in this paper, following the research of QEAs and GAs, a novel real-coded QGA is proposed. This method adopts real numbers instead of binary code in order to improve algorithm searching ability and population diversity. Therefore, the complex coding and decoding processes can be avoided. Furthermore, the interval division approach is used, which can improve the searching capabilities and reduce computation cost by interval parallel computing.

This paper is organized as follows. Section 1 is the introduction; the QGA will be reviewed in detail. In section 2, the principles and procedures of the proposed algorithm will be discussed, followed by the numerical case studies before being applied into two nonlinear data reconciliation cases. The results and future work will be summarized in the last section.

2 Real-coded Quantum-inspired Genetic Algorithm

2.1. Quantum-inspired Genetic Algorithm (QGA)

It will be very instructive to review the classical QGA first before introducing the proposed algorithm. QGA has stronger search ability and quicker convergence speed since it introduces the concepts of quantum bit and quantum rotation gate. In the QGA, the state of a unit is depicted by quantum bit and angle, which are defined as below.

Quantum bit, the smallest unit in the QGA, is defined as a pair of numbers as shown in Eq. (1),

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{1}$$

The modulus $|\alpha|^2$ and $|\beta|^2$ give the probabilities that the quantum bit exists in states “0” and “1”, respectively, which satisfies Eq. (2),

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

A string of quantum bits consists of a quantum bit individual, which can be defined by Eq. (3),

$$q = \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_k \\ \beta_1 & \beta_2 & \cdots & \beta_k \end{pmatrix} \tag{3}$$

Therefore, a chromosome can be represented as a string of quantum bit individuals as shown in Eq. (4),

$$q_j^t = \begin{pmatrix} \alpha_{11}^t & \alpha_{12}^t & \cdots & \alpha_{1k}^t & \alpha_{21}^t & \alpha_{22}^t & \cdots & \alpha_{2k}^t & \alpha_{m1}^t & \alpha_{m2}^t & \cdots & \alpha_{mk}^t \\ \beta_{11}^t & \beta_{12}^t & \cdots & \beta_{1k}^t & \beta_{21}^t & \beta_{22}^t & \cdots & \beta_{2k}^t & \beta_{m1}^t & \beta_{m2}^t & \cdots & \beta_{mk}^t \end{pmatrix} \tag{4}$$

A quantum bit individual is able to represent a linear superposition of all possible solutions due to its probabilistic representation. This quantum bit representation has better characteristic of generating diversity in population than other representations.

Because of the normalization condition, the quantum angle can be represented by Eq. (5),

$$\begin{cases} |q_j^t\rangle = \cos\theta_j^t\,|0\rangle + \sin\theta_j^t\,|1\rangle \\ \theta_j^t = \arctan\dfrac{\beta_j^t}{\alpha_j^t} \end{cases} \tag{5}$$

The fundamental update mechanism of QGA is evolving quantum bits and angles, by which the updated quantum bits should still satisfy the normalization condition. The quantum rotation gate update equation could be calculated by Eq. (6),

$$\begin{bmatrix} \alpha_j^{t+1} \\ \beta_j^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\Delta\theta_j^{t+1} & -\sin\Delta\theta_j^{t+1} \\ \sin\Delta\theta_j^{t+1} & \cos\Delta\theta_j^{t+1} \end{bmatrix} \begin{bmatrix} \alpha_j^t \\ \beta_j^t \end{bmatrix} \tag{6}$$

Although the quantum bit and rotation gate representation has better characteristics of population diversity, the premature convergence problem could still appear because of the poor performance of binary representations. Therefore, in this paper, a novel quantum-inspired genetic algorithm is proposed, which will be discussed extensively in the next section.

2.2. The Proposed Algorithm

The proposed algorithm in this section adopts real numbers instead of chromosome in order to improve algorithm searching ability and population diversity. The detailed encoding rules and procedures are presented as follows.

2.2.1. *Encoding rules*

Suppose the boundaries of variable X as $[X_{\min}, X_{\max}]$, and divide this interval into L consecutive subintervals such as,

$R_1: [X_{\min}, X_{\min} + (X_{\max} - X_{\min})/L]$,

$R_2: (X_{\min} + (X_{\max} - X_{\min})/L, X_{\min} + 2*(X_{\max} - X_{\min})/L]$,

……

$R_i: (X_{\min} + (i-1)*(X_{\max} - X_{\min})/L, X_{\min} + i*(X_{\max} - X_{\min})/L]$,

……

$R_L: (X_{\min} + (L-1)*(X_{\max} - X_{\min})/L, X_{\max}]$.

then algorithm will works on these L subintervals simultaneously.

For subintervals $R_1, R_2, \cdots R_L$, the chromosomes can be coded by Eq. (7),

$$\begin{cases} \alpha_{R_i} = \sqrt{\dfrac{X_{R_i} - X_{\min}}{(X_{\max} - X_{\min})/L}} \\ \beta_{R_i} = \sqrt{\dfrac{[X_{\min} + (X_{\max} - X_{\min})/L] - X_{R_i}}{(X_{\max} - X_{\min})/L}} \end{cases} \tag{7}$$

Here, $X_{R_i} \in (X_{\min} + (i-1)*(X_{\max} - X_{\min})/L, X_{\min} + i*(X_{\max} - X_{\min})/L]$.

As a sequence, an individual p can be expressed by Eq. (8),

$$\begin{aligned} p &= \begin{bmatrix} \alpha_{R_1} & \alpha_{R_2} & \cdots & \alpha_{R_i} & \cdots & \alpha_{R_L} \\ \beta_{R_1} & \beta_{R_2} & \cdots & \beta_{R_i} & \cdots & \beta_{R_L} \end{bmatrix} \\ &= \left[p_{R_1}, p_{R_2}, \cdots, p_{R_L} \right] \end{aligned} \tag{8}$$

Compared with Eqs. (3)-(4), Eqs. (7)-(8) not only greatly reduces the length of individual coding, but also uses overall interval division covering the entire searching space. The searching efficiency could be improved when subinterval number increases, but algorithm will become more complicated and the associated computational cost (CPU time) will also increase and this result will be seen in the posterior case studies.. So, the number of subintervals which can be adjusted manually or automatically by the program is an important factor for algorithm performance.

2.2.2. *Procedures*

There are six steps in the proposed algorithm.

Step-1(initialization). Set n as the population size, g as iteration variable ($g = 1$ when algorithm starts), $G_{\max}$ as the maximum iteration number, $p^g = \{p_1^g, p_2^g, \cdots, p_n^g\}$ as the initial population. $p_j^g, j = 1, 2, \cdots n$ denotes the j-th individual of the s-th generation as shown in Eq. (9),

$$p_j^g = \begin{bmatrix} \alpha_{j,R_1}^g & \alpha_{j,R_2}^g & \cdots & \alpha_{j,R_i}^g & \cdots & \alpha_{j,R_L}^g \\ \beta_{j,R_1}^g & \beta_{j,R_2}^g & \cdots & \beta_{j,R_i}^g & \cdots & \beta_{j,R_L}^g \end{bmatrix} = \begin{bmatrix} p_{j,R_1}^g & p_{j,R_2}^g & \cdots & p_{j,R_i}^g & \cdots & p_{j,R_L}^g \end{bmatrix} \quad (9)$$

Initially, all the values are set as the median of subintervals, that is,

$$p_j^1 = \begin{bmatrix} \alpha_{j,R_1}^1 & \alpha_{j,R_2}^1 & \cdots & \alpha_{j,R_i}^1 & \cdots & \alpha_{j,R_L}^1 \\ \beta_{j,R_1}^1 & \beta_{j,R_2}^1 & \cdots & \beta_{j,R_i}^1 & \cdots & \beta_{j,R_L}^1 \end{bmatrix} = \begin{bmatrix} h & h & \cdots & h & \cdots & h \\ h & h & \cdots & h & \cdots & h \end{bmatrix} \quad (10)$$

where $h=(X_{\max}-X_{\min})/(2L)$.

Step-2(Fitness Evaluation). Suppose the subinterval evaluation of one individual is the best, fitness evaluation strategy is shown in Eq. (11)[25],

$$\theta = \arcsin\left(\frac{a-b}{c-b}\right) \quad (11)$$

where a is the real value of the corresponding chromosome derived, and b,c are the corresponding upper and lower boundaries, θ is the corresponding angle of the best individual.

Step-3(Judgement). If the satisfactory solution is acquired or the iteration number reaches $G_{\max}$, then the algorithm stops. Otherwise, go to *Step-4*.

Step-4(Crossover). Suppose p_j^g is the best individual stored in $p^g = \{p_1^g, p_2^g, \cdots, p_n^g\}, j = 1,2,\cdots,n.$ p_{j,R_i}^s and θ_{j,R_i}^s are the component of subinterval and corresponding angle respectively. Crossover operation is implemented only in the subintervals where this subinterval located. Then the subinterval component of each individual of next generation will be obtained by Eqs. (12) and (13),

$$\Delta\theta_{k,R_i}^g = \theta_{k,R_i}^g - \theta_{j,R_i}^g, (k = 1,2,\cdots,n) \quad (12)$$

$$p_{k,R_m}^{g+1} = \begin{cases} p_{j,R_m}^g \cos^2\left(\Delta\theta_{k,R_m}^g\right) + p_{k,R_m}^g \sin^2\left(\Delta\theta_{k,R_m}^g\right), & m = i \\ p_{k,R_m}^g, & m \neq i \end{cases} \quad (13)$$

Step-5(Mutation). This operation will be used for poor individuals (unideal fitness values), and can be expressed by Eq. (14),

$$p^{g+1} = p^g \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \quad (14)$$

Step-6(Updating). Let $g = g+1$, and jump to *Step-2*.

In the next section, the proposed algorithm will be tested with several benchmark functions before being applied into real industrial process data reconciliation.

3. Benchmark Functions Test

In this section, the proposed algorithm will be compared with two QGAs using several benchmark functions listed in Tab.1.

Table 1. Benchmark functions

Function	Boundary	Global Optimum	Variable
$F_1 = \sum_{i=1}^{10} x_i^2$	[-100, 100]	0	0
$F_2 = 200 + \sum_{i=1}^{20}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	[-5.12, 5.12]	0	0
$F_3 = 418.9829n - \sum_{i=1}^{30} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500, 500]	0	-420.97
$F_4 = \sum_{i=1}^{49}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	[-30, 30]	0	1

For these QGAs, Parameters are selected samely as follows, maximum iteration number $G_{\max} = 1000$, population dimension $n = 50$, population size $N = 10$, limit probability $\delta = 0.01$. The stop criterion of algorithm is the evolvement reaches the maximum iteration number. If the absolute error between the results and global optimal value is less than 0.001, the optimization process will be considered as success. Besides, for testing subinterval number effect on the final performance of the proposed algorithm, three intervals 2, 3, and 5 are selected respectively. The tested results are shown in Tab.2.

Table 2. Benchmark functions testing results

Function	Algorithm	OV	MIN	CT(s)	Intervals
F_1	QGA	89.3	---	907.1	---
	RCQGA	$3.8*10^{-5}$	860	189.6	---
	the proposed algorithm	$1.93*10^{-4}$	890	200.3	2
		$3.7*10^{-5}$	800	167.9	3
		$1.83*10^{-5}$	817	175.6	5
F_2	QGA	33.58	---	1869.3	---
	RCQGA	$2.5*10^{-5}$	870	453.2	---
	the proposed algorithm	$1.48*10^{-4}$	920	600.4	2
		$5.7*10^{-5}$	850	450.9	3
		$2.1*10^{-5}$	812	476.1	5
F_3	QGA	25.3	---	2312.6	---
	RCQGA	$6.2*10^{-5}$	830	878.3	---
	the proposed algorithm	$1.58*10^{-4}$	900	1005.2	2
		$6.3*10^{-5}$	780	917.6	3
		$4.9*10^{-5}$	801	941.3	5
F_4	QGA	56.87	---	3245.4	
	RCQGA	$8.7*10^{-5}$	890	1468.2	
	the proposed algorithm	$2.21*10^{-4}$	900	1656.3	2
		$1.9*10^{-4}$	820	1472.2	3
		$5.3*10^{-5}$	855	1486.8	5

OV-Optimum Value, MIN-Mean Iteration Number, CT-CPU Time

From Tab.2 it can be seen that the proposed algorithm has better performance that QGA, and if the number of initial subintervals chosen appropriately (such as L=3) it also has better general performance (less MIN and CT) than RCQGA[25]. These results not only illuminate validity of the proposed algorithm, but also indicate the number of initial subintervals effect on it.

4. Nonlinear Data Reconciliation With The Proposed Algorithm

4.1 Distilling Process Data Reconciliation

Conventional distilling process is shown in Fig.1.

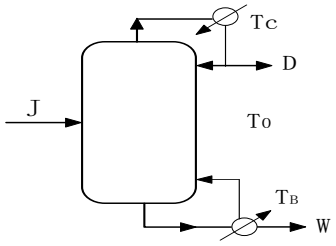

Fig. 1. Conventional distilling process schematic diagram J-input quantity(kmol·h^{-1}), D-distillate quantity(kmol·h^{-1}), W-residue quantity(kmol·h^{-1}), T_0-ambient temperature(℃), T_B-reboiler temperature(℃), T_C-condenser temperature(℃).

There are 10 constraint equations in this typical bilinear process, which are listed in Tab.3.

Table 3. Constraint equations of rectification process

No	Equation	Number of Equations
1	$F_1 = J - D - W = 0$	1
2	$F_j = Jz_i - Dy_i - Wx_i = 0, j = 2\ldots7, i = j-1$	6
3	$F_8 = \sum x_i - 1 = 0$	1
4	$F_9 = \sum y_i - 1 = 0$	1
5	$F_{10} = \sum z_i - 1 = 0$	1

x_i -residue component, y_i -distillate component, z_i -input component

There are 21 variables in this process; the actual experiential values of each variable are as follows,

$$\bar{J} = 450.96 \quad \bar{D} = 196 \quad \bar{W} = 254.96$$
$$\bar{x}_1 = 0 \quad \bar{x}_2 = 0 \quad \bar{x}_3 = 0$$
$$\bar{x}_4 = 0 \quad \bar{x}_5 = 0 \quad \bar{x}_6 = 1$$
$$\bar{y}_1 = 0.0726 \quad \bar{y}_2 = 0.0517 \quad \bar{y}_3 = 0.1674$$
$$\bar{y}_4 = 0.5037 \quad \bar{y}_5 = 0.2407 \quad \bar{y}_6 = 0$$
$$\bar{z}_1 = 0.0315 \quad \bar{z}_2 = 0.0068 \quad \bar{z}_3 = 0.0726$$
$$\bar{z}_4 = 0.2185 \quad \bar{z}_5 = 0.1044 \quad \bar{z}_6 = 0.5662$$

Intervals of every variable are,

$$450 \le J \le 451 \quad 195 \le D \le 197 \quad 254 \le W \le 256$$
$$0 \le x_1 \le 0.0005 \quad 0 \le x_2 \le 0.002 \quad 0 \le x_3 \le 0.0002$$
$$0 \le x_4 \le 0.002 \quad 0 \le x_5 \le 0.002 \quad 0 \le x_6 \le 1.1$$
$$0.06 \le y_1 \le 0.08 \quad 0 \le y_2 \le 0.06 \quad 0.16 \le y_3 \le 0.18$$
$$0.5 \le y_4 \le 0.52 \quad 0.24 \le y_5 \le 0.25 \quad 0 \le y_6 \le 0.01$$
$$0.02 \le z_1 \le 0.04 \quad 0.005 \le z_2 \le 0.007 \quad 0.06 \le z_3 \le 0.08$$
$$0.2 \le z_4 \le 0.23 \quad 0.09 \le z_5 \le 0.11 \quad 0.56 \le z_6 \le 0.57$$

A group of measurement data collected randomly from actual process is,

$$J' = 450.63 \quad D' = 196.52 \quad W' = 255.09$$
$$x_1' = 0.0014 \quad x_2' = 0.0013 \quad x_3' = 0$$
$$x_4' = 0.0007 \quad x_5' = 0.0011 \quad x_6' = 0.9993$$
$$y_1' = 0.0713 \quad y_2' = 0.0526 \quad y_3' = 0.1694$$
$$y_4' = 0.5055 \quad y_5' = 0.2434 \quad y_6' = 0.0003$$
$$z_1' = 0.0336 \quad z_2' = 0.0059 \quad z_3' = 0.0737$$
$$z_4' = 0.2109 \quad z_5' = 0.1022 \quad z_6' = 0.5671$$

The objective function is defined by Eq. (15),

$$E = Min\left[\left(\hat{X}-\bar{X}\right)^T Q_X^{-1}\left(\hat{X}-\bar{X}\right)+\left(\hat{Y}-\bar{Y}\right)^T Q_Y^{-1}\left(\hat{Y}-\bar{Y}\right) +\left(\hat{Z}-\bar{Z}\right)^T Q_Z^{-1}\left(\hat{Z}-\bar{Z}\right)+\sum_{i=1}^{10} F_i^2\right] \quad (15)$$

Where $\bar{X},\bar{Y},\bar{Z}$ are the measured value vector, $\hat{X},\hat{Y},\hat{Z}$ are the reconciled value vector, $Q_X^{-1},Q_Y^{-1},Q_Z^{-1}$ are the covariance matrixes of measurement errors of X,Y,Z which are assumed to be given or estimated.

The number of subintervals is 4, and maximum iteration number is $G_{\max}=1000$. The results are shown in Tab.4 and Fig.2.

Table 4. Data reconciliation results with the proposed algorithm

Variable	0	300	500	800	1000
J	450.63	450.45	450.65	450.87	450.87
D	196.52	196.55	196.33	196.11	196.11
W	255.09	255.21	255.11	255	255
x_1	0.0014	0.0010	0.0008	0.0004	0.0004
x_2	0.0013	0.0010	0.0008	0.0001	0.0001
x_3	0	0	0	0	0
x_4	0.0007	0	0	0	0
x_5	0.0011	0.0014	0.0008	0.0002	0.0002
x_6	0.9993	0.9993	0.9997	0.9997	0.9997
y_1	0.0713	0.0717	0.0727	0.0730	0.0730
y_2	0.0526	0.0520	0.0517	0.0514	0.0514
y_3	0.1694	0.1655	0.1662	0.1666	0.1666
y_4	0.5055	0.5050	0.5041	0.5032	0.5032
y_5	0.2434	0.2437	0.2424	0.2411	0.2411
y_6	0.0003	0	0	0	0
z_1	0.0336	0.0321	0.0317	0.0314	0.0314
z_2	0.0059	0.0066	0.0066	0.0068	0.0068
z_3	0.0737	0.0733	0.0729	0.0722	0.0722
z_4	0.2109	0.2177	0.2196	0.2205	0.2205
z_5	0.1022	0.1021	0.1029	0.104	0.104
z_6	0.5671	0.5673	0.5670	0.5666	0.5666
E	90.7819	65.7062	57.6636	50.7454	50.7454

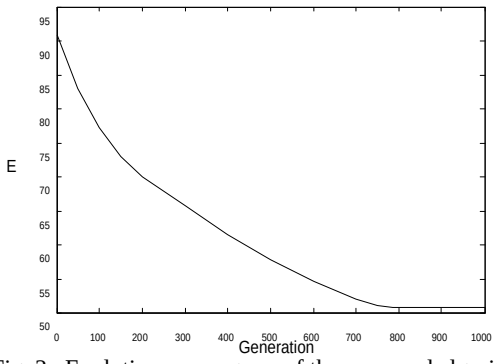

Fig. 2. Evolutionary progress of the proposed algorithm

The simulation results denote that the reconciled value of each amount is no longer change positively after 786 generations. According to the criteria of data reconciliation, reconciled value of every variable satisfies balance equations, the objective function decreased and stabilized after 786 generations (4.3 seconds), reconciliation goal is achieved. Besides, from the results, it is possible to use a great number of generations to obtain good results.

4.2 Extraction Process of Composing Juice Data Reconciliation

Extraction process of composing juice is shown in Fig.3.

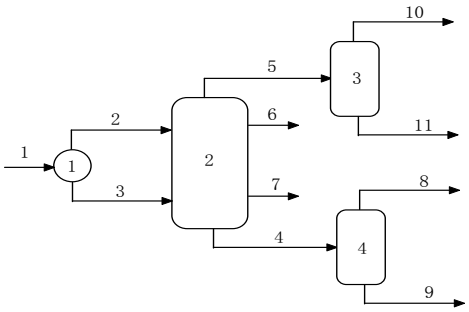

Fig. 3. Extraction progress of composing juice

The process has 4 equipments, 7 components and 88 variables which are 11 matter flow variables $F_j(j=1,2,\cdots,11)$ and 77 component variables $x_{j,k}(j=1,2,\cdots,11.k=1,2,\cdots,7)$.

Parameters of the proposed algorithm are as follows, the number of subintervals is 3, and maximum iteration number is $G_{\max}=1000$. The reconciliation results compared with Crowe[1] and NLP[26] are shown in Tab.5 (a part of variables are listed, * represents unmeasured variable) and Tab.6.

Table 5. Reconciliation results using Crowe, NLP and the proposed algorithm

Variable	Measured Value	Standard Deviation	Crowe	NLP	The Proposed Method
F_1	3707	37.07	3611.9	3642.8	3639.5
C(1,1)	1.64	0.0164	1.5743	1.5822	1.5841
C(1,2)	17.66	0.1766	18.653	18.31	18.472
C(1,3)	0.38	0.01	0.3739	0.3774	0.3821
C(1,4)	3.44	0.0344	3.4132	3.3503	3.4255
C(1,5)	61.47	0.6147	61.137	61.374	61.333
C(1,6)	15.39	0.1539	14.423	14.681	14.711
C(1,7)*	---	---	0.3264	0.325	0.3391
C(2,1)	1.64	0.0164	1.5743	1.5822	1.562
C(3,1)	1.64	0.0164	1.5744	1.5822	1.5617
F_4	2910	29.1	2846.5	2883.1	2879.7
C(4,1)	0.0052	0.01	0.0283	0.0298	0.0319
C(4,2)	0.003	0.01	0.0037	0.0031	0.004
C(4,3)	0.19	0.01	0.1787	0.1847	0.1912
C(4,4)	3.68	0.0368	3.9126	3.8145	3.8423
C(4,5)*	---	---	77.256	77.226	77.177
C(4,6)	18.01	0.1801	18.428	18.549	18.557
C(4,7)*	---	---	0.1925	0.1928	0.1899
F_7	7.6	0.076	7.5991	7.6024	7.6031
C(7,1)	4.19	0.0419	4.2359	4.1924	4.2025
C(7,2)	43.65	0.4365	43.652	43.642	43.641
C(7,3)	7.91	0.0791	7.9116	7.9103	7.9046
C(7,4)	9.05	0.0905	9.0517	9.0513	9.0528
C(7,5)	30.3	0.303	30.305	30.3	30.33
C(7,6)	0	0.01	0	0	0
C(7,7)*	---	---	4.8438	4.9042	4.8507
F_9*	---	---	2741.6	2779.7	2757.8
C(9,1)	0.0013	0.01	0.0248	0.0268	0.0267
C(9,2)	0.0016	0.01	0.0008	0.0002	0.0007
C(9,3)	0.18	0.01	0.1809	0.1871	0.1788
C(9,4)	0.25	0.01	0.2497	0.2462	0.2471
C(9,5)*	---	---	80.211	80.1	80.086
C(9,6)	19	0.19	19.133	19.239	19.245
C(9,7)	0.2	0.01	0.1999	0.2	0.1973
F_{11}*	---	---	672.95	666.06	663.77
C(11,1)	0.006	0.01	0.0149	0.0124	0.016
C(11,2)	98.75	0.9875	97.523	97.527	97.5
C(11,3)	0	0.01	0.0004	0.0007	0.0005
C(11,4)	0.88	0.01	0.8802	0.8806	0.8798
C(11,5)	0.7	0.01	0.7001	0.6997	0.6875
C(11,6)	0	0.01	0	0	0.0003
C(11,7)	0.88	0.01	0.8798	0.8801	0.8728

Table 6. Compare results of Crowe, NLP and the proposed algorithm

No	Crowe	NLP	the proposed algorithm
TA	330.59	191.25	249.36
Te(s)	1.031	8.281	3.035

$TA = \sum_{i=1}^{n} (\frac{x_i^{'} - x_i^{*}}{\sigma_i})^2$. $x_i^{'}$ -measured value of x_i ,

x_i^{*} -reconciliation value of x_i , σ_i -standard deviation of x_i ,

n -number of measurable variables.

$T_e = \sum_{j=1}^{N} t_j / N$. t_j -algorithm run time of each loop.

From the above simulation results, it can be seen that in the three methods, NLP has the best reconciliation precision, but its run time is so longer. Crowe has the shortest run time, but unideal precision. The proposed algorithm obtains better reconciliation results with shorter run time, and shows that it has the best global performance in nonlinear data reconciliation.

5. Conclusions

A novel quantum-inspired genetic algorithm with real-coded, interval division thinking was proposed in this paper, and some benchmark functions simulation illustrated that it could avoid some drawbacks of traditional QGA. Furthermore, the proposed algorithm was applied in industrial manufacture process data reconciliation, and simulation results showed that it had global performance in nonlinear data reconciliation and could be used in the relevant researches and applications.

Acknowledgements

This work is supported by National High Technology Research and Development Program of China (863 Program) (2009AA04Z141) and the Natural Science Foundation of Shandong (Y2008G14).

References

1. C. M. Crowe, Reconciliation of process flow rates by matrix projection, part II, the nonlinear case, *AIChE J*. **32**(4) (l986) 6l6–623.
2. D. E. Simpson and M. G. Everett, Reducing the number of unknowns in a constrained minimization problem-an application to material balances, *Appl. Math. Modeling*. **l2**(4) (1988) 204–212.
3. L. K. Zhou, H. Y. Su and J. Chu, A modified outlier detection method in dynamic data reconciliation, *Chinese Journal of Chemical Engineering*. **13**(4) (2005) 542–547.
4. J. L. Li and G. Rong, Improved mixed integer optimization approach for data rectification with gross error candidates, *Chinese Journal of Chemical Engineering*. **17**(2) (2009) 226–231.
5. Q. Gao, W. W. Yan and H. H. Shao, A novel robust nonlinear dynamic data reconciliation, *Chinese Journal of Chemical Engineering*. **15**(5) (2007) 698–702.

6. C. L. Mei, H. Y. Su and J. Chu, An NT-MT combined method for gross error detection and data reconciliation, *Chinese Journal of Chemical Engineering*. **14**(5) (2006) 592–596.
7. L. K. Zhou, H. Y. Su and J. Chu, A new method to solve robust data reconciliation in nonlinear process, *Chinese Journal of Chemical Engineering*. **14**(3) (2006) 357-363.
8. C. Y. Jiang, T. Qiu, J. S. Zhao and B. Z. Chen, Gross error detection and identification based on parameter estimation for dynamic systems, *Chinese Journal of Chemical Engineering*. **17**(3) (2009) 460–467.
9. S. Sen, S. Narasimhan and K. Deb, Sensor network design of linear processes using genetic algorithms, *Comp. Chem. Engng*. **22**(3) (1998) 385–390.
10. W. Wongphaka, Modified genetic algorithm for nonlinear data reconciliation, *Comp. Chem. Engng*. **29** (2005) 1059–1067.
11. J. X. Gao, H. G. Dong, J. J. Huang, Z. Z. Han and X. M. Xu, Data rectification based on fuzzy self-adaptability genetic algorithm, *Control and Instruments in Chemical Industry*. **34**(4) (2007) 9–13.
12. M. Moore and A. Narayanan, Quantum-inspired computing, *Dept. of Comput. Sci.* (Univ. Exeter, Exeter, U.K, 1995).
13. K. H. Han and J. H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Trans. Evol. Comput.* **6**(6) (2002) 580–593.
14. K. H. Han and J. H. Kim, Quantum-inspired evolutionary algorithms with a new termination criterion, Hc Gate, and two-phase scheme, *IEEE Trans. Evol. Comput.* **8**(2) (2004) 156–169.
15. J. G. Vlachogiannis and K. Y. Lee, Quantum-inspired evolutionary algorithm for real and reactive power dispatch, *IEEE Trans. Power Syst.* **23**(4) (2008) 1627–1636.
16. B. B. Li and L. Wang, A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling, *IEEE Trans. SMC Part B.* **7**(3) (2007) 576–591.
17. L. C. Jiao, Y. Y. Li, M. G. Gong and X. R. Zhang, Quantum-inspired immune clonal algorithm for global optimization, *IEEE Trans. SMC Part B.* **38**(5) (2008) 1234–1253.
18. K. Meng, H. G. Wang, Z. Y. Dong and K. P. Wong, Quantum-inspired particle swarm optimization for valve-point economic load dispatch, *IEEE Trans. Power Syst.* **25** (1) (2010) 215–222.
19. H. Liu and G. Zhang, Quantum genetic local search algorithm and its application of reactive power optimization, *Journal of Power System and Automation*. **21**(2) (2009) 6–10.
20. J. Song, C. Fan and H. Quan, Application of quantum genetic algorithm in fingerprint image segmentation, *Computer Application and Software*. **24**(12) (2007) 171–172.
21. X. Zhang, X. Zhu and J. Lin, Self-tuning PID controller parameters based on quantum genetic algorithm, *Computer Engineering and Applications*. **43**(21) (2007) 218–220.
22. L. Wang and B. B. Li, Quantum-inspired genetic algorithms for flow shop scheduling, *Studies in Computational Intelligence*. **121** (2008) 17–56.
23. Y. M. Wang and P. J. Yao, Simulation and optimization for thermally coupled distillation using artificial neural network and genetic algorithm, *Chinese Journal of Chemical Engineering*. **11**(3) (2003) 307–311.
24. G. X. Zhang and W. D. Jin, Improvement of quantum genetic algorithm and its application, *Journal of Southwest Jiaotong University*. **38**(6) (2003) 717–722. (in Chinese)
25. H. Chen, J. Zhang and C. Zhang, Real-coded chaotic quantum-inspired genetic algorithm, *Control and Decision*. **20**(11) (2005) 1300–1303. (in Chinese)
26. I. Kim, et, al, Robust data reconciliation and gross error detection-the modified MIMT using NLP, *Comp. Chem. Engng*. **21** (1997) 1775–1782.