A Quasi-Newton Population Migration Algorithm for Solving Systems of Nonlinear Equations

Hongxia Liu
College of Mathematics and Computer Science Guangxi, University for Nationalities, Nanning 530006, China
Email: yongquanzhou@126.com

Yongquan Zhou
College of Mathematics and Computer Science Guangxi, University for Nationalities, Nanning 530006, China
Email: yongquanzhou@126.com

Yongmei Li
College of Computer and Electron information, Guangxi University, Nanning 530004, China
Email:liyongmei180@163.com

***Abstract*—In this paper, the problem on solving nonlinear equations is transformed into that of function optimization. A new Quasi-Newton Population Migration Algorithm (QPMA) is proposed via combination of population migration algorithm and Quasi-Newton method. The algorithm has the advantages of the Population Migration Algorithm (PMA) such as region search in a certain extent and avoid getting into the local optimum and the Quasi-Newton method such as Quasi-Newton's local strong searching. Finally, the numerical experiments result show that this algorithm can find the rapid and effective interval solution and the probability of success is higher.**

***Index Terms*—Nonlinear equations, Quasi-Newton method, Population Migration Algorithm**

I. INTRODUCTION

Solving nonlinear equations is the actual engineering is an important problem in numerical weather prediction, oil geological exploration, computing biochemistry, and control field and track design, etc, which has a strong application background. For a long time, people made a lot of research in the theoretical and numerical calculations of the nonlinear equations. But solving for systems of nonlinear equation still haunt the people a problem[1], especially for the high nonlinearity of the practical engineering problems. It is always the lack of efficient and reliable algorithms. Traditional solutions for the problem have mainly iterative method, the gradient method; conjugate direction method and so on. But these methods have high demands for characteristic equations; there is also a big obstacle for many complex equation systems.

Population Migration Algorithm (PMA) [2], which simulates the population with the economic center of gravity and the transfer mechanism, thereby contributing to a better algorithm for selection of regional search, and it simulates the diffusion mechanism with the population pressures increase to a certain extent, it can avoid falling into local minima. Quasi-Newton method has a faster local convergence and is a very effective local search iterative algorithm. However, local search property of these algorithms will be efficient only when the initial values are near the optimal values. Therefore, a new method was proposed combined Population Migration Algorithm's region search with Quasi-Newton' local search.

The paper is organized as follows. The problem on solving nonlinear equations function optimization model is described in Section 2. In Section 3, the Quasi-Newton population migration algorithm is introduced. Some of numerical equations examples application is introduced in Section 4. Finally, a few conclusions are presented.

II. DESCRIPTION OF THE PROBLEM

General form of nonlinear equations (Assume that there are n variables and n equations) is

$$\begin{cases} f_1(x_1, x_2, ..., x_3) = 0 \\ f_2(x_1, x_2, ..., x_3) = 0 \\ \quad\vdots \\ f_n(x_1, x_2, ..., x_3) = 0 \end{cases} \tag{1}$$

Solving this equation is equivalent to a global optimization of the following problem:

$$\begin{cases} find: X = [x_1, x_2, ..., x_n], X \in \Phi \\ s.t. \quad \min f(X) = \sqrt{\sum_{i=1}^{n} f_i^2(X)} \end{cases} \tag{2}$$

In equation (2), Φ is interval of equations, when $f(X)$ minimum value equal to zero, Corresponding X is a solution of equations. Thus the solution of nonlinear equations is transformed into minimum issue of PMA.

doi:10.4304/jcp.6.1.36-42

III. Quasi-Qewton Population Migration Algorithm

A. Quasi-Qewton Method [3-5]

Since a function for computing the Hessian used in computing the direction is rarely available, attention has focused on computing it numerically. The calculation of the Hessian is very expensive computationally, however, and efforts were made to find a way to produce the Hessian more cheaply. The critical insight from which came the current Quasi-Newton methods was made byBroyden use information from the current iteration to compute the new Hessian [6]. Let

$$s_k = x_{m+1} - x_m = \alpha_m \delta_m \quad (3)$$

Be the change in the parameters in the current iteration, and

$$\eta_k = g_{m+1} - g_m \quad (4)$$

be the change in the gradients. Then a natural estimate of the Hessian at the next iteration H_{m+1} would be the solution of the system of linear equations

$$H_{m+1} s_m = et\alpha_m \quad (5)$$

that is, H_{m+1} is the ratio of the change in the gradient to the change in the parameters. This is called the quasi-Newton condition. There are many solutions to this set of equations. Broyden suggested a solution in the form of a secand update

$$H_{m+1} = H_m + uv^t \quad (6)$$

Further work has developed other types of secant updates, the most important of which are the DFP (for Davidon [7], and Fletcher and Powell [8], and the BFGS (for Broyden [7], Fletcher [9], Goldfarb[10], and Shanno, [11]). The BFGS is generally regarded as the best performing method:

$$H_{m+1} = H_m + \frac{\eta_m \eta_m{}^t}{\eta_m{}^t s_m} - \frac{H_m s_m s_m{}^t}{s_m{}^t H_m s_m} = H_m + \frac{\eta_m \eta_m{}^t}{\eta_m{}^t s_m} - \frac{g_m g_m{}^t}{\delta_m{}^t g_m} \quad (7)$$

taking advantage of the fact that

$$H_m s_m = \alpha_m H_m \delta_m = \alpha_m g_m \quad (8)$$

The BFGS method is used in the GAUSS function QNEWTON. However, the update is made to the Cholesky factorization of *H* rather than to the Hessian itself, that is to R where $H = R^t R$ .In QNEWTON *H* itself is not computed anywhere in the iterations. The direction *.m* is computed using *CHOLSOL*, that is as a solution to

$$R^t R_m \delta_m = g_m \quad (9)$$

where R_m and g_m are its arguments, and R_m is the Cholesky factorization of H_m .Then R_{m+1} is computed as an update and a down date to R_m using the GAUSS functions CHOLUP and CHOLDN.

B. CHOLUP and CHOLDN

Suppose one was using a Cholesky factorization of a moment matrix computed on a data set in some analytical method. Then suppose you have acquired an additional observation. The Cholesky factorization could be re-computed from the original data set augmented by the additional row. Such a complex and time consuming computation is not necessary. Numerical methods have been worked out to computer the new updated factorization with the new observation without having to re-compute the moment matrix and factorization from scratch. This update is handled by the GAUSS CHOLUP

Function. In a similar way a Cholesky factorization can be downdated, i.e., a new factorization is computed from a data set with one observation removed. In either of these functions there are two arguments, the original factorization and the observation to be added or removed.

In this section, let back to the BFGS secant update in the Quasi-Newton method. The factorization R_{m+1} is generated by first adding the "observation" $\eta_m / \sqrt{\eta_m^t s_m}$ and then removing the "observation" $g_m / \sqrt{\delta_m^t g_m}$.

C. Population Migration Algorithm

Population migration algorithm (PMA) was proposed by ZHOU Yonghua, MAO Zongyuan in 2003, it is a kind of global optimization algorithm that simulates population migration mechanism. In the PMA, the optimization variables x correspond to the population habitual residence, the objective function $f(x)$ corresponds to the attractive place of residence, the global (local) optimal solution of the problem corresponds to the most attractive (the beneficial) areas. Population Migration Algorithm can be divided into population mobility, population migration and population proliferation of three basic forms: population flow corresponds to the local random search of the algorithm; population migration corresponds to the way of choosing solution according to the rule of people flow to the rich places; population proliferation corresponds to overall search strategy and escape from local optimum strategy according to the rule of moving from developed areas to less developed areas.

As a swarm intelligence optimization algorithm, in the PMA, the optimization variable x refers to living places, the objective function $f(x)$ refers to attractive of the residence place, the optimal solution (local optimal solution) refers to the most attractive place (beneficial region), the "up" or "mountain climbing" of algorithm refers to move to beneficial region, escaping from the local optimal refers to move out of beneficial region as a result population pressure; population flow corresponds to random, local searching method; and population migration corresponds to the way to choose the approximate solution like as population struggles upwards; population proliferation combines the overall searching with escaping from the local optimal strategy.

The general process of PMA is as follows:

BEGIN

Initialize: Initialize the model of the population, record the best individual and best value according to the evaluation function

While (Termination rules not satisfied) *do*

{

Flow in their respective regions, and record the point's value;

Determine the beneficial region according to the largest value point, and produce new population to replace the original population in the new beneficial region;

Contraction the beneficial region, then population flow also produces new population in the region of the largest value point;

If the population pressure is too heavy, then population proliferation, produce new population in the original searching places.

}

End While

Output the best individual and best value.

END

For global optimization problems, optimization problems are usually defined as:

$$\max f(x) \quad s,t. x \in S \tag{10}$$

Where $x = (x_1, x_2, .., x_d, .., x_D) \in S \subseteq R^D, 1 \le d \le D, d \in Z, f(x)$ is the objection function, S is the searching places with D dimensions. For PMA, the global optimization problem (10), where $f: S \to R$ is a real map, $x \in R^n$, $S = \sum_{i=1}^{n} [a_i, b_i]$ is the searching place, and $a_i < b_i, i = 1, 2, ..., n.$ it is assumed that the global optimum value f^* existed; the global optimum solution M is not empty. In the PMA algorithm, optimizing variables x are corresponding to the information of habitual residence population; $x^i = (x_1^i, x_2^i, ..., x_n^i)$ refers to the *i*-th point, $x^i \in R^n; x_j^i$ represents the *j*-th weight of the *i*-th point; $\delta^i \in R^n, \delta_j^i$ is the radius of the j-th weight, $\delta_j^i = (b_j - a_j)/(2N), \delta_j^i > 0;$ $i = 1, 2, .., N; j = 1, 2, ..., n.$ N is the population size. The objective function corresponds to the surface space of population migration as well as all kinds of information; the optimal solution of the question corresponds to the most attractive regions; the algorithm to move up or climbing refers to migrate the beneficial area, algorithm to escape from the local optimization corresponds to people emigrate from the beneficial caused by too heavy population pressure, population flow corresponds to the local random search of the algorithm.

D. Quasi-Newton Population Migration Algorithm (QPMA)

For Optimization problem (10), the Quasi-Newton population migration algorithm was designed. In the algorithm, people and their location were with point. Where $x^i = (x_1^i, x_2^i, \cdots x_n^i)$ refers to the i -th point, $x^i \in R^n$; x_j^i represents the j -th weight of the i -th point; $\delta^i \in R^n, \delta_j^i$ is the radius of the j -th weight, $\delta_j^i = (b_j - a_j)/(2N)$, $\delta_j^i > 0$; $i = 1, 2, .., N; j = 1, 2, ..., n.$ N is the population size.

Step 1. In the search space, uniformly and randomly generate N points $y^1, y^2, \cdots y^N$. For each i , set the center of the region is $center^i = y^i$,determine it upper and lower bound $center^i \pm \delta$, where $\delta_j^i = (b_j - a_j)/(2N)$. $i = 1, 2, \cdots, N, j = 1, 2, \cdots, n$ (the method of δ_j^i makes all of the above are equivalent, so the following steps remove the superscript).

Step 2. Take $y^i, (i = 1, 2, \cdots, N)$ as the initial value of the objective function, uses the quasi-newton population migration algorithm to optimize then get a new set of points $x^i (i = 1, 2, ..., N)$.

Step 3. Calculate the income/ attractive $f(x^i)$ of all the points.

Step 4. In accordance with the calculated values from Step 3, initialize the value of the best point and the best point.

Step 5. Then people flow in their own region. uniformly and randomly change each point:

$$x^i = 2\delta rand(*) + (center^i - \delta)$$

where $rand(*)$ is random value. If $x_j^i > b_j$, then $x_j^i = b_j$; if $x_j^i < a_j$,then $x_j^i = a_j$.

Step 6. Calculate the income/ attractive of all the points.

Step 7. Record the optimal value and the best point.

Step 8. If the number of population movement i is less than the number of pre-specified , then go to Step 5.

Step 9. Population migration: let the most attract point as the center, according to the size of each component δ to determine the beneficial areas. Uniformly and randomly generate N points in this region to replace the original points.

Step 10. Calculate the income/ attractive of all the points.

Step 11. Record the optimal value and the best point.

Step12. The way of shrink the region as follows: $\delta = \delta(1-\Delta), \Delta$ is contraction coefficient, where $0 < \Delta < 1)$.

Step 13. People flow in the beneficial areas and transfer with the economic gravity: let the most attract point as the center, according to the size of each component δ to determine the beneficial areas. uniformly and randomly generate N points in this region to replace the original points.

Step 14. Calculate the income/ attractive of all the points.

Step 15. Record the optimal value and the best point.

Step 16. If $\max \delta_j > \alpha$ (α is population pressure parameter, it is a given value), go to step 12.

Step 17. Report the results.

Step 18. Population flow: uniformly and randomly generate N points in this region to replace the original points. Determine the regions of people movement according to Step 1.

Step 19. Calculate the income/ attractive of all the points.

Step 20. Record the optimal value and the best point.

Step 21. Add1 to the number of iteration, if less than the specified number then goes to Step 5.

Step 22. End.

IV. NUMERICAL EXPERIMENTS

In order to test the performance of QPMA algorithm for solving systems of nonlinear equations, this paper selects nine typical nonlinear equations as examples, and compares the other algorithms with QPMA then the results show that QPMA has a high success rate and precision, strong robust.

Example 1[12]. Solving for nonlinear equation

$$\begin{cases} x_1^{x_2} + x_2^{x_1} - 5x_1x_2x_3 = 85 \\ x_1^3 - x_2^{x_3} - x_3^{x_2} = 60 \\ x_1^{x_3} + x_3^{x_1} - x_2 = 2 \end{cases}$$

where $0 \le x_1, x_2, x_3 \le 10$.

The best result is $x^* = (4,3,1)^T$.

Example 2[12]. Solving for nonlinear

$$\text{equation} \begin{cases} x_1 + x_2 - 2x_3 = 0 \\ x_1 x_2 = 0 \\ x_1^2 + x_2^2 = 2 \end{cases} ,$$

where $0 \le x_1, x_2, x_3 \le 10$.

The best result is $x^* = (1,1,1)^T$.

Example 3[13]. Solving for nonlinear equation

$$\begin{cases} 821x^2 - 263yz + 661 = 0 \\ 613xz - 977yx - 268 = 0 \\ 977xz + 373x - 647yz - 811 = 0 \end{cases}$$

where $0 \le x, y, z \le 10$.

The best result is $x^* = (2,3,5)^T$

Example 4[13]. Solving for nonlinear equation

$$\begin{cases} 3x^2 + 2yz - \sqrt{2}x - 6 = 0 \\ xz - y - \sqrt{2} + 1 = 0 \\ xy - z - \sqrt{2} + 1 = 0 \end{cases}$$

where $0 \le x, y, z \le 2$.

The best result is $x^* = (\sqrt{2},1,1)^T$.

Example 5[14]. Solving for nonlinear equation

$$\begin{cases} -94x^{15} - 64 + 90x^2 - 38xy^3 = 0 \\ 64 - 22y^2z^{20} - 37xy = 0 \\ -20x - 7y + 4y^{20} + 1 + z = 0 \end{cases} ,$$

where $-5 \le x_1, x_2, x_3 \le 5.$.

The best result is unknown.

Example 6[14]. Solving for nonlinear equation

$$\begin{cases} f_1(x) = x_1^2 - x_2 + 1 = 0 \\ f_2(x) = x_1 - \cos(0.5\pi x_2) = 0 \end{cases}$$

where $x \in [-2, 2]$.

The best result is $x^* = (-1/\sqrt{2}, 1.5)^T$, $x^* = (0,1)^T$, $x^* = (-1, 2)^T$.

Example 7[15]. Solving for nonlinear equation

$$\begin{cases} f_1(x) = (x_1 - 5x_2)^2 + 40\sin^2(10x_3) = 0 \\ f_2(x) = (x_2 - 2x_3)^2 + 40\sin^2(10x_1) = 0, \\ f_3(x) = (3x_1 + x_3)^2 + 40\sin^2(10x_2) = 0 \end{cases}$$

where $-1 \le x_1, x_2, x_3 \le 1$.

The best result is $x^* = (0,0,0)^T$.

Example 8. Solving for nonlinear

$$\text{equation} \begin{cases} f_1(x) = x_1^3 + e^{x_1} + 2x_2 + x_3 + 1 = 0 \\ f_2(x) = -x_1 + x_2 + x_2^3 + 2e^{x_2} - 3 = 0, \\ f_3(x) = -2x_2 + x_3 + e^{x_3} + 1 = 0 \end{cases}$$

where $-2 \le x_1, x_2, x_3 \le 2$.

The best result is unknown.

Example 9. Solving for nonlinear

$$\text{equation} \begin{cases} f_1(x) = x_1 + \frac{1}{4}x_2^2 x_4 x_6 + 0.75 = 0 \\ f_2(x) = x_2 + 0.405e^{(1+x_1x_2)} - 1.405 = 0 \\ f_3(x) = x_3 - \frac{1}{4}x_4 x_6 + 1.25 = 0 \\ f_4(x) = x_4 - 0.605e^{(1-x_3^2)} - 0.395 = 0 \\ f_5(x) = x_5 - \frac{1}{2}x_2 x_6 + 1.5 = 0 \\ f_6(x) = x_6 - x_1 x_5 = 0 \end{cases},$$

where $-2 \le x_1, x_2, x_3, x_4, x_5, x_6 \le 2$.

The best result is $x^* = (-1,1,-1,1,-1,1)^T$

Table Ⅰ.

Performance of QPMA comparing with other algorithm

NO.	Algorithm	x_1	x_2	x_3	Success-rate
1	PMA	3.999 991 95	2.999 994 97	0.999 979 86	85%
	References[12]	3.999 4	3.007 9	1.007 9	-
	QPMA	3.999 999 998 62	2.999 999 997 50	0.999 999 998 21	100%
	Best result	4	3	1	-
2	PMA	0.999 561 61	1.004 389 43	0.999 991 94	88%
	References[12]	1.1075	0.9822	0.999 9	-
	QPMA	1.000 361 371 04	0.999 638 496 94	0.999 999 933 88	100%
	Best result	1	1	1	-
3	PMA	1.999 699 23	2.999 960 44	4.999 937 91	87%
	References[13]	1.999 975 1	2.999 960 4	4.999 999 9	-
	QPMA	1.999 999 992 83	2.999 999 990 08	4.999 999 986 60	100%
	Best result	2	3	5	-
4	PMA	1.414 211 12	0.999 996 45	0.999 369 12	80%
	References[13]	1.414 212 02	0.999 998 78	0.999 999 75	-
	QPMA	1.414 213 554 90	1.000 000 012 83	1.000 000 014 52	100%
	Best result	$\sqrt{2}$	1	1	-
5	PMA	0.658 751 12	-1.001 174 17	1.071 822 30	88%
	References[14]	0.658 768 343 3	-1.001 177 172	1.071 892 300	-
	QPMA	0.658 768 345 64	-1.001 177 174 13	1.071 892 301 83	100%
	Best result	-	-	-	

In the test, parameter settings are as follows: Population size N=30, the number of population mobility l=10, Contraction coefficient Δ =0.01, population pressure parameters α =1e-8, Number of iterations is decided by specific examples. Maximum number of iterations is 50. Table 1 shows the results of this paper comparing with other algorithm.

As can be seen from the above results, QPMA algorithm is the best one than those in the literature in the performance, and the success rate achieves 100%.

The success rate of the standard algorithm is less than the QPMA. For example 1 to 2, times for iteration is only ten, we can achieve the best result and spend litter time; for example 3 to 4, the accuracy is 8 about the method of references literature six, while QPMA can get 15, times for iteration are 10; For example 5, references literature 7 use preprocessing and interval method can obtained four peaks of function, but this paper you can quickly find of a solution via iteration 50 times, success rate is 100%, although PMA algorithm have ability to search equations approximate solutions, Success rate is worse than the other algorithms, which success rates are 88% and 86% Respectively. QPMA algorithm is up to 100%.

Table Ⅱ.

Performance of QPMA comparing with other algorithm

N0.	algorithm	times	Success-rate	Mean of x_1	Mean of x_2	Mean of x_3
6	PMA	46	92%	-0.707 106 452 2 0.000 003 071 0	1.500 009 103 0 1.000 000 310 3	23 times 23 times
	References[15]	43	86%	-0.707 724 0.000 114	1.500 668 0.999 817	26 times 17 times
	QPMA	50	100%	-0.707 106 781 2 0.000 000 000 0 -1.000 000 000 0	1.500 000 001 1 1.000 000 000 0 2.000 000 000 0	10 times 35 times 5 times
7	PMA	44	88%	0.000 000 007 0	0.000 000 000 1	0.000 000 001 2
	References 15]	27	54%	0.000 001	-0.000 012	-0.000 001
	QPMA	50	100%	0.000 000 000 0	0.000 000 000 0	0.000 000 000 0
8	PMA	44	88%	-0.767 760 553 3	0.075 330 671 04	-1.162 158 841 0
	QPMA	50	100%	-0.767 760 563 5	0.075 330 676 2	-1.162 151 184 2
9	PMA	43	86%	-1.000 001 231 1 0.999 999 879 9	0.999 969 548 1 -1.000 000 362 2	-1.000 000 012 1 0.999 998 587 4
	QPMA	50	100%	-1.000 000 000 0 1.000 000 000 0	1.000 000 000 0 -1.000 000 000 0	-1.000 000 000 0 1.000 000 000 0

V. CONCLUSIONS

To overcome the problems of the classical algorithms for solving nonlinear equations, such as high sensitivity to the initial guess of the solution, poor convergence reliability and can't get all solutions, etc. This paper presents a new Quasi-Newton Population Migration Algorithm (QPMA) for solving systems of nonlinear equations; it combined with convergence of PMA and Quasi-Newton's local strong searching. In the course of the making algorithm, we insert into Quasi-Newton method in the standard PMA and apply advantages respectively. The numerical experiments results show that the QPMA) algorithm can quickly and effectively find the interval solution of nonlinear equations, and success rate can achieve 100%. It is an effective method to solve nonlinear equations.

ACKNOWLEDGMENT

This work is supported by Grants 60461001 from NSF of China and the project Supported by Grants 0832082, 0991086 from Guangxi Science Foundation

REFERENCES

[1] X.D. Huang, Z.G. Zeng, Y.N. Ma, Nonlinear numerical analysis. Wu Han: Wuhan University Press,2000.

[2] Y.H, Zhou. Z.Y, Mao, A new search algorithm for global optimization-Population Migration Algorithm(I). Journal of South China University of Technology (Natural Science Edition),2003,31(3):1-5

[3] B. L, Chen, The theory of optimum and algorithm. Beijing, Tsinghua University Press,1989.

[4] Broyden C G, Dennis J E, More J J On the local and superlinear convergence of quasi-Newton methods. J Inst Math Appl,1973,12:223-246

[5] Ronald Schoenberg. Optimization with the Quasi-Newton method. Aptech Systems, Inc. Maple Valley, WA. 2001.

[6] Broyden, C.G., A new double-rank minimization algorithm. Notices of the American Mathematical Society, 16:670, 1969.

[7] Davidon, W.C., Variable metric methods for minimization, Argonne National Labs Report ANDL-5990.

[8] Fletcher, R., Powell. M.D.J., A rapidly convergend descent method for minimization, Computer Journal, 7; 149-154, 1963.

[9] Fletcher, R. A new approach to variable metric methods. Computer Journal, 13:317-322, 1970.

[10] Goldfarb, D., A family of variable metric methods derived by variation means, Mathematics of Computation, 24:23-26, 1970.

[11] Shanno, D.F., Conditioning of quasi-Newton methods for function minimization, Mathematics of Computation, 24:145-160, 1970.

[12] Y, Zeng, The application of improved genetic algorithms to solving non-linear equation group. Journal of East China Jiao tong University,2004,21(4):132-134

[13] F.L, Chen, W.Yang, The application of interval arithmetic at Wu elimination method to solving the algebraic equations. Science in China Ser. Mathematics, 2005.35(8):910-921.

[14] Y.H, Li, J.W, Xue, Y, Feng, Solving nonlinear systems via preprocessing and interval method. Journal of Sichuan University: Engineering Science Edition, 2004, 36(5):86-93.
[15] J. K, Zhang, X. Z, Wang, S. Y, Liu. Particle swarm optimization for solving nonlinear equation and system. Computers Engineering and Applications, 2006, 42(7):56-58.
[16] Yongquan Zhou, Huajuan Huang. Hybrid artificial fish school algorithm based on mutation operator for solving nonlinear equation. 2009 International Workshop on Intelligent Systems and Applications (ISA2009), Wuhan, China,23-24 May, 2009,vol.1:108-112.
[17] Aijia Ouyang, Yongquan Zhou, Qifang Luo. Hybrid particle swarm optimization for solving systems of nonlinear equation. 2009 IEEE International Conference on Granular computing. (Grc2009) Nanchang, China, August 17-19, 2009, pp460-465
[18] Yongquan Zhou, Xingqong Wei, Dexiang Luo. A constructor method of interpolation polynomials based on PSO. 2009 International Forum on Information Technology and Applications (IFITA 2009). 15-17 May 2009, Chengdu, China. 2009, vol.2:745-748

Hongxia Liu received the M.S degree in computer from Guangxi University for Nationalities, Nanning, China. He current research interests including computation intelligence.

Yongquan Zhou Ph.D. Professor. His received the B.S. degree in mathematics from Xianyang Normal University, Xianyang, China. In 1983. the M.S. degree in computer from Lanzhou University , Lanzhou, China. In 1993, and Ph.D. in computation intelligence from Xidian University, Xian, China. His current research interests including computation intelligence, neural network.

Yongmei Li received the M.S degree in computer from Guangxi University, Nanning, China. He current research interests including swarm intelligence algorithm, neural network.