Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080453

Functional Network Design using Parallel Cultural Algorithm

Qifang Luo[1], *Yongquan Zhou*[1,2,*], *Peigang Guo*[1] *and Xin Chen*[1]

[1] College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China
[2] Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning 530006, China

Received: 22 Aug. 2013, Revised: 24 Nov. 2013, Accepted: 25 Nov. 2013
Published online: 1 Jul. 2014

Abstract: In this paper, a parallel cultural algorithm for designing functional network structure method is proposed. This method can get the basis function and its parameters with random optimal searching strategy obtained the learning between functional network structure and the functional parameters. Finally, the simulation experiment results show that the proposed constructive functional network optimization design scheme is feasible and effective, and can obtain better generalization property.

Keywords: Functional network, Cultural Algorithms(CA), Genetic Algorithm(GA), Basis function.

1 Introduction

Functional network is introduced by Castillo in 1998 [1]. It is an extension of the standard neural network. Unlike neural networks, it deals with general functional models instead of sigmoid-like ones, in these networks there are no weights associated with the links connecting neurons, and the neural functions are unknown from given families to be estimated during the learning process. We can select appropriate families for each specific problem, such as polynomials, Fourier expansions and trigonometric functions, etc.. At present, the functional network is a very useful general framework for solving a wide range of problems: The solving of differential functional and difference equation [2], nonlinear time series and prediction modeling [3], factorization model of multivariate polynomials [4], the identification of nonlinear system [5], CAD, linear and nonlinear regression [4], etc. The functional networks have shown excellent performance in the above-mentioned problems.

So far, functional network achieved a greater success in the aspect of application, but its mathematics theory foundation is imperfect, to some extent, greatly limits the scope of application of the functional network. These key theories include: first, what type of network structure and the family of basis function, often by experts to make judgments based on empirical knowledge; second, the network structure and functional parameters, this point is often difficult to determine by artificial; third, the network structure is determined, also exist the local minima problem. At present, there are some using genetic programming [6] to solve the above problems, and have achieve good results; but the family of basis function how to select, the theory has not yet been given a general method to avoid the family of basis function sample select insufficient, thereby lead to functional network approximation error is too large or to much which led to overfit problems.

In 1994, Robert G. Reynolds from the simulation of human social and cultural evolution of the idea, proposed cultural algorithm (CA) [7]. Cultural algorithm is an algorithm used to solve complex calculations in the novel global optimization search algorithm. In human society, cultural can be seen as the carrier of information, the information potentially affects all members of society, and the benefit of offspring to solve the problem of guiding their practical activity. Similarly, the cultural algorithm is that the important idea of the evolution of populations from the problem to be solved to obtain the knowledge (i.e. beliefs), and the feedback that knowledge to guide the search process. That the process is, respectively, from the micro and macro level, two different levels of simulated biological evolution and cultural dimensions of evolution, the evolution and influence each other and promote each other, so they

[*] Corresponding author e-mail: yongquanzhou@126.com

© 2014 NSP
Natural Sciences Publishing Cor.

formed a ”dual evolution dual promote” mechanism. In recent years, CA has caused great concern in domestic and foreign scholars [8,9,10,17,18].

This paper, we using parallel cultural algorithm for optimization design functional network structure model, according to the structural characteristics of the functional networks. In this model, the genetic algorithm and multiple population space incorporated into the cultural algorithm framework, composition based on genetic algorithm population space and knowledge space two spaces, this dual structure characteristic of the cultural algorithm, which makes the method, can be used in the process of problem solving from population obtain the experience knowledge to guide the search process, so it has better global optimization performance and faster search speed. Finally, the simulation experiment shows the feasibility and accuracy of the method.

The rest of the paper outlines as follows. In Section 2, the basic cultural algorithm and the parallel cultural algorithm are introduced. The functional network is introduction in Section 3. The optimizing neuron function types in functional network design via parallel cultural algorithm are given in Section 4. The experimental results and analysis is given in Section 5. Finally, Section 6 concludes this paper.

2 Cultural Algorithm

2.1 Basic Cultural Algorithm

The traditional evolutionary computation methods have limited mechanisms for representing, storing and transmitting knowledge from one generation to the next. Cultural algorithm (CA) was proposed by Reynolds in 1994. CA models the evolution of the cultural component in an evolutionary computational system over time. Culture helps human beings adapt to their environments. In human societies, culture can be viewed as a vehicle for encoding, generalization and storage of knowledge potentially accessible to all members of the society. It is useful to guide their problem solving activities. On defining human culture, it is generally assumed that there exists a way to represent, acquire and reason about knowledge in a cultural content. CA is a class of computational models derived from models of cultural evolution processes. These algorithms support the basic mechanisms for cultural changes described in anthropological and archaeological literature.

Seen from a technical point of view, a CA is a knowledge-based evolutionary computational system. Its basic idea is to incorporate knowledge components into traditional evolutionary computational systems. One of its frameworks can be depicted in Figure 1. Its models two levels of evolution: the population space level and the belief space level. The two spaces are connected together by an explicit communication protocol composed of an acceptance function and an influence function, which are denoted here as *Accept*() and *Influence*(), respectively. The acceptance function is used to glean the experience of selected individuals from the population; then the belief space can be modified by an update function, denoted here as *update*(); next, the influence function can make use of the problem-solving knowledge in the belief space to guide the evolution of the population component. In the population space, like traditional evolutionary population models, individuals are first evaluated by a generation function *objective*(). New individuals are created by a generation function *generate*(). Then a selection function *select*(), is used to select the population for next generation. Cultural algorithm pseudo-code is given in the literature [7].

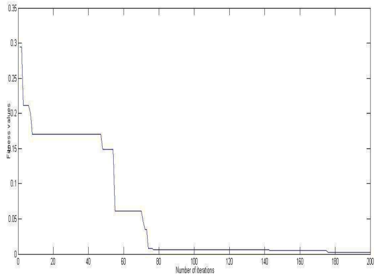


Fig. 1: A framework of cultural algorithm

Cultural algorithm has the characteristics: (1) Dual evolutionary inheritance: In the population space and belief space are inherited parent information; (2) Population space evolution is saved by the belief space knowledge to guide; (3) Support the population space and belief space hierarchy; (4) Support the adaptive evolution of two space; (5) Different space evolution can be carried out at different speeds; (6) Support a hybrid of different algorithms to solve the problem; (7) ”Cultural” change can be expressed in different models within a model. Cultural algorithm flow chart can be depicted in Figure 2.

2.2 Parallel Cultural Algorithm

Parallel cultural algorithm is in the traditional cultural algorithm based on the multiple population space introduced into the framework of the cultural algorithm, thus constitutes multiple populations of parallel cultural algorithm model, shown in Figure 3. From Figure 1 and Figure 3 comparison can be seen, the traditional cultural algorithm only one population space; whereby parallel cultural algorithm model with multiple population space, each population space alone on their own space within the individual fitness calculation and evaluation, to generate region of population space the best individual, and

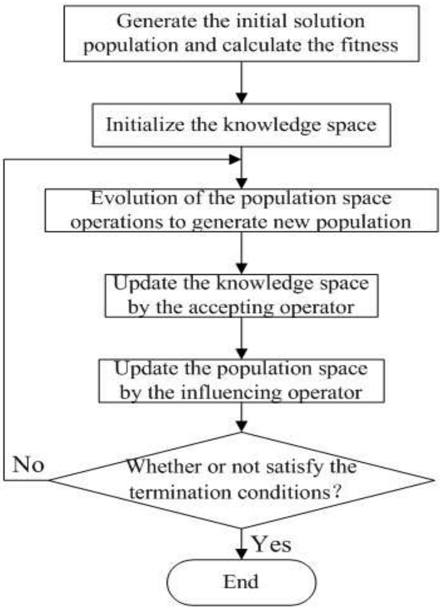

Generate the initial solution population and calculate the fitness
Initialize the knowledge space
Evolution of the population space operations to generate new population
Update the knowledge space by the accepting operator
Update the population space by the influencing operator
Whether or not satisfy the termination conditions?
No
Yes
End

Fig. 2: A flow chart of cultural algorithm

according to the evolution steps of genetic algorithm for each population space within the evolved independently, complete population space updating. The population space performance evaluation and evolution operation using an independent the sub-process to complete, in order to reduce the algorithm degree of coupling. Each population space by synchronous transmission mode regularly through the acceptance operation to update the knowledge space, the knowledge space has also constantly its performance evaluation and evolution, and regularly through the influence operation to guide every population space evolution.

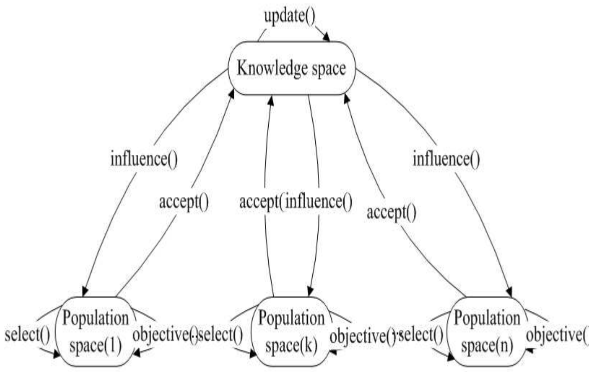


Fig. 3: The model of parallel cultural algorithm

3 Functional Network

In general, a functional network consists of the following elements:

(1) A layer of input units. This first layer contains the input data. Input units are represented by small black circles with its corresponding name.

(2) A layer of output units. This is the last layer, and contains the output data. Output units are also represented by small black circles with its corresponding name.

(3) One or several layers of processing units. A neuron is a computing unit which evaluates a set of input values, coming from the previous layer, of neurons or input units, and gives a set of output values to the next layer, of neurons or output units. The computing units are connected to each other, in the sense that output from one unit can serve as part of input to another neuron or to the units in the output layer. Once the input values are given, the output is determined by the neuron type, which can be defined by a function. For example, assume that we have a neuron with s inputs $(x_1, x_2, ..., x_s)$ and k outputs y_1, $y_2, ..., y_k$, then, we assume that there exist k functions f_j, $j = 1, 2, ..., k$, such that $y_j = f_j(x_1, x_2, ..., x_s)$, $j = 1, 2, ..., k$. The functions f_j are not arbitrary, but determined by the structure of the network, as we shall see later. Neurons are represented by circles with the name of the corresponding f_j function inside.

(4) A set of directed links. They connect the input layer to the first layer of neurons, neurons of one layer to neurons of the next layer, and the last layer of neurons to the output units. Connections are represented by arrows, indicating the information flow direction. All these elements together form the network architecture, which defines the functional capabilities of the network. Network architecture refers to the organization of the neurons and the connections involved. In multilayer networks, units are organized in series of layers. Information flows in only one direction, from the input layer to the output layer. Neuron units receive information only from previous layers of the network, and output information to the next layer of neurons, or to the output units. In Ref [5] one example of a simple functional network is given in Figure 4.

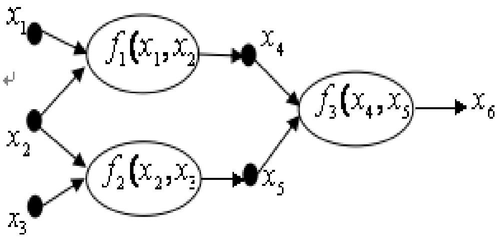


Fig. 4: A typical functional network topology model

© 2014 NSP
Natural Sciences Publishing Cor.

In Figure 4, where the input layer consists of the units $\{x_1, x_2, x_3\}$, the first layer of neurons contains neurons f_1 and f_2, the second layer of neurons contains neurons f_3, and the output layer reduces to the unit x_6. One of the most important is the choice of neurons function $f_i(i = 1,2,3)$. According to Castillo's approach will each neuron functions $f_i(i = 1,2,3)$ is represented as a linear combination of the known functions of a given family. Such as, polynomials, trigonometric functions, Fourier expansion etc. In the neural network, each neuron function f_i, that is, the activation functions often take the Sigma function, hyperbolic tangent function, etc. In standard neural networks the neuron functions f_i are fixed, and some weights associated with the links or connections have to be learned. However, in functional networks there are no weights, and the neuron functions f_i must be learned. As for the other differences with the neural networks[12].

3.1 The general functional networks model

The general functional network model [13],

$$y = \sum_{r_1=1}^{q_1} \dots \sum_{r_k=1}^{q_k} c_{r_1 r_2 \dots r_k} \phi_{r_1}(x_1) \dots \phi_{r_k}(x_k), \qquad (1)$$

When contains the transformation of the corresponding variable y, equation (1) be generalized as,

$$f(y) = \sum_{r_1=1}^{q_1} \dots \sum_{r_k=1}^{q_k} c_{r_1 r_2 \dots r_k} \phi_{r_1}(x_1) \dots \phi_{r_k}(x_k), \qquad (2)$$

Where $c_{r_1 r_2 \dots r_k}$ are unknown parameters expressed in terms of functions Φ_s, where $\Phi_s = \{\varphi_{r_s}(x_s), r_s = 1,2,\dots,q_s\}$ and $s = 1,2,\dots,k$ are linearly independent of the family of basic functions. The general functional network model is given in Figure 5.

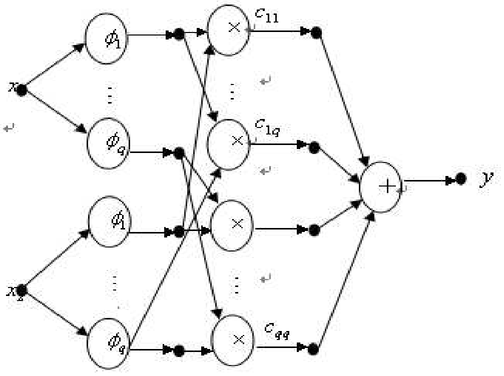


Fig. 5: The general functional network model

3.2 The separable functional networks model

In this section, we demonstrate a simple separable functional network with two inputs and one output [14]. Figure 6 depicts the topology of a separable functional network. The relationship between z, x and y can be defined as follows,

$$z = F(x,y) = \sum_{i=1}^{n} f_i(x) g_i(y) = \sum_{j=1}^{m} h_j(x) k_j(y), \qquad (3)$$

where x,y are the two input variables and z is the output of the functional network, $f_i(\bullet)$, $g_i(\bullet)$, $h_i(\bullet)$ and $k_i(\bullet)$ are the unknown neuron functions. According to Castillo's approach will each neuron functions f_i is represented as a linear combination of the family of basic functions. For example, in Figure 6, each neuron function f_i and g_i can be written as

$$f_i(x) = \sum_{j=1}^{n} a_{ij} \varphi_{ij}(x) \qquad (4)$$

$$g_i(y) = \sum_{j=1}^{n} b_{ij} \psi_{ij}(y) \qquad (5)$$

where a_{ij} and b_{ij} are the functional parameters, $\varphi_{ij}(x)$ and $\psi_{ij}(y)$ are the family of basic functions, both of which can be obtained by learning. In this way, the whole design of the construction of separable functional networks can become single neuron one by one optimization problem.

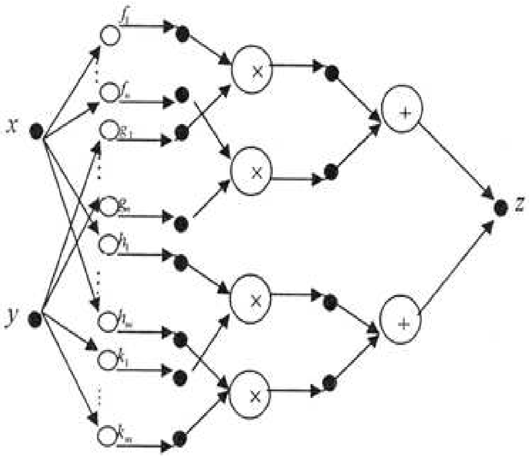


Fig. 6: The separable functional networks model

In general, the construction of separable functional networks design steps as follow:

① Generate the first neuron, the initial sample set P and the target set T as the current training set;

② The current training set for training the newly generated neurons. This article will use the parallel cultural algorithm design, meet the requirements until the training error, training stop, the whole design process is completed, otherwise turn to ③;

③ The newly trained neurons, input P of each vector $x_1, x_2, \ldots, x_m$, calculate the actual output $\widehat{y_1}, \widehat{y_2}, \ldots, \widehat{y_m}$, and T is adjusted to $T' = \{y_j = y_j - \widehat{y}_j, j = 1, 2, \ldots, m\}$;

④ Generate a new neuron, the initial sample set P and adjustment to T' as the current training set, turn to ②.

Thus, for the functional networks affect network performance factors, including the basis function selection and the number of neurons choose. Set consistent with for all neurons function base, function base consists of the M basis functions, networks have D neurons, the total number N[14] of the reference of the results of the neural network training can be approximate is the following estimates (assuming that sample set is Gaussian distribution):

$$N \approx \frac{M * D}{\varepsilon} \tag{6}$$

where ε take the around 10. In general, in practical applications, select the smaller function base, could suitable to increase the number of neurons, thereby reducing the network not match degree; when the network structure not satisfied with the practical requirements, the optional larger scale function base [16].

4 Functional network optimizing design using parallel cultural algorithm

4.1 The family of basic functions and their coding, sorting

(1) The family of basic functions
In functional networks, commonly used the family of basic functions: polynomial basis functions: $\{1, x, x^2, x^3, \ldots\}$; exponential basis functions: $\{1, e^x, e^{2x}, e^{3x}, \ldots\}$; Fourier basis functions: $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots\}$, etc.
(2) Populations space and Knowledge space encoding

Populations space encoding and the basic idea of evolution: functional networks commonly used the family of basic functions into populations space, each type of the family of basic functions corresponds to a population space. For each population space, one of the family of basic function each term as an individual, and according to the binary coding, 0 represents corresponding to of basis function of the bit will not be taken; on the contrary, 1 represents corresponding to of basis function of the bit will be taken. Encoding length according to the scale of the problem was determined. And in the population space generation and binary strings of equal length of real numbers, each time evolution, the real numbers of the corresponding also perform the genetic operation and evolution. Knowledge space and population space used the same encoding.

(3) Sorting Prior to each class of basis function according to dictionary order. For instance, for polynomial basis function $\{1, x, x^2, x^3, \ldots\}$. In ascending order according to this sort, namely the dictionary order. Whereas according to with the encoding on the above of population space, if there is a binary string of , then the polynomial in terms of their corresponding term is $1, x, x^3, x^5, x^6, x^7$. The other family of basis functions and polynomial sorting is similar.

4.2 Parallel cultural algorithm obtained functional network optimization model of neuron types

In accordance with the requirements of the problem, corresponding to the basis function series in population space, every kind of the family of basic function corresponds in a population space, so that through the cultural algorithm framework, the knowledge space and lower the population space through a communication protocol, the knowledge space and population space two large spaces together, in the case, respectively, from the micro (population space) and macro (knowledge space) two different levels simulation of biological level evolution and cultural level of the evolution, the evolution process influence each other and promote each other, so they constitute such ”dual evolution dual promote” mechanism. This constant influence and promotion in order to find the optimal model in the functional network of neuron function of types. Optimizing neuron function types in functional network design model via parallel cultural algorithm is given in Figure 7.

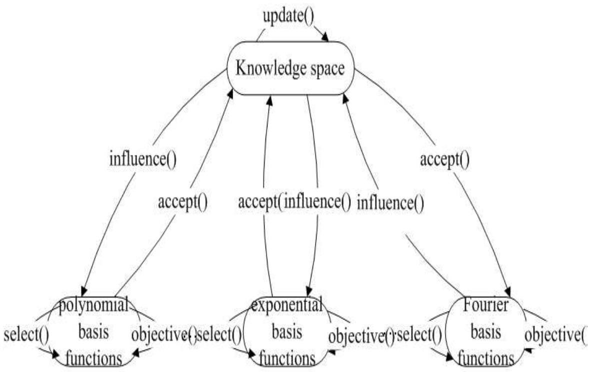


Fig. 7: Optimizing neuron function types in functional network design model via parallel cultural algorithm

© 2014 NSP
Natural Sciences Publishing Cor.

4.3 The objective function, fitness function defined

In general, for the training of neurons, training goal is the desired output and the actual output of the mean square error of the least. We may assume that the definition is to train the neuron function of the objective function as follows:

$$y_i = F(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} f_i(x) \tag{7}$$

where $f_i(x) = \sum_{j=1}^{n} a_{ij}\varphi_{ij}(x)$, a_{ij} is the functional parameters, $\{\varphi_{ij}(x)|j = 1, 2, ..., n\}$ is the family of basic functions.

Fitness function can be defined as,

$$Fint = RMSE = \frac{1}{m}\sqrt{\sum_{j=1}^{m}(y_j - \widehat{y}_j)^2} \tag{8}$$

where m is the sample size, y_i and $\widehat{y}_i$ are the functional network prediction output and actual output. In practical application, in order to test the functional network approximation ability, the study sample will be divided into two parts, the part is used to calculate the fitness value; and the other part is used for test samples.

4.4 Cultural algorithm with two large space communications

(1) Acceptance function-*accept*()

The acceptance function *accept*() is used to choose directly influence the knowledge of individual in the current knowledge space. In other words, in each population space of evolution of individual, each evolution *AcceptStep*, with each population space current global best value to replace the worst individual knowledge space.

$$AcceptStep = RMSE = N_1 + \frac{CurrentStep}{EndStep} \times N_2 \tag{9}$$

(2) Influence function-*influence*()

The Knowledge space by function-*influence*() influence the population space generate the next generation. In other word, in the knowledge space, each evolve *InfluenceStep* generations, the knowledge space group as part of a good fitness value replace the worst individual population.

$$InfluenceStep = N_1 + \frac{EndStep - CurrentStep}{EndStep} \times N_2 \tag{10}$$

where *EndStep* is maximum generations number; *CurrentStep* is current generation number; N_1 and N_2 are predefined constants, Here were taken at 15 and 100.

4.5 Parallel cultural algorithm to achieve the functional networks learning algorithm

Parallel cultural algorithm to achieve the functional networks learning algorithm steps are as follows:

Step1: Each population space structure using a genetic algorithm, generating size *P* of initial population *POP(t)*, encoding method using a binary code$(0,1)$. Moreover, initialize the populations and randomly generated initial population of the same length real number, each time after evolution, real with the evolution of genetic operation.

Step2: The *POP(t)* of each individual fitness is calculated by the objective function *objective()*, the fitness using $RMSE = \frac{1}{m}\sqrt{\sum_{j=1}^{m}(y_j - \widehat{y}_j)^2}$. The fitness function value is smaller, illustrate the better.

Step3: Individuals according to their fitness in ascending order, select the first *P* fitness value of the minimum of the individual, and the minimum fitness value of individuals to save for the initialization (or update) the knowledge space.

Step4: Using the acceptance function *accept*() to select some excellent individuals, to give the knowledge space, and to update the knowledge space.

Step5: The population space of individuals selection, crossover and mutation operations.

Step6: Through the influence function *influence()* in turn affects the lower population space, return step2.

Step7: To determine whether a termination condition is satisfied, yes, to terminate the algorithm; no, return step2.

The algorithm flow chart is shown as follow, in Figure 8.

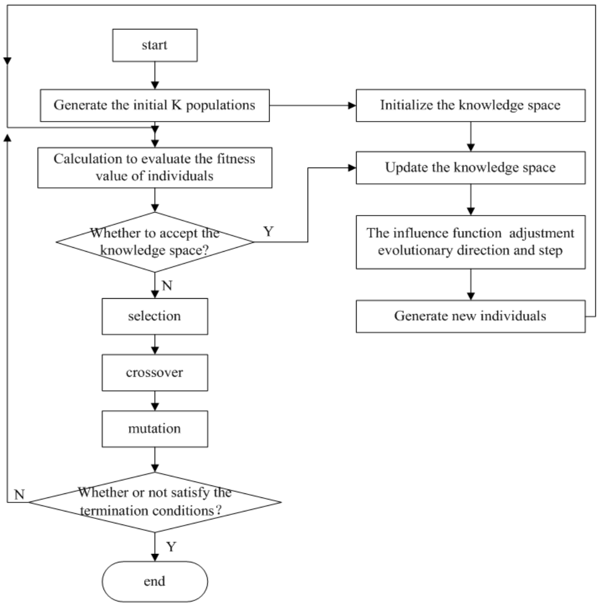

start
Generate the initial K populations
Initialize the knowledge space
Calculation to evaluate the fitness value of individuals
Update the knowledge space
Whether to accept the knowledge space?
Y
The influence function adjustment evolutionary direction and step
N
selection
Generate new individuals
crossover
mutation
N
Whether or not satisfy the termination conditions?
Y
end

Fig. 8: The flow chart of the proposed algorithm

© 2014 NSP
Natural Sciences Publishing Cor.

5 Simulation experiment

The experimental simulation platform: Operating system: Windows7, CPU: Core i3-370, Frequency: 2.40GHz, RAM: 2GB, Integrated development environment: MATLAB R2008a.

5.1 Experimental Results

In the experiment, the population space number is 3, respectively, selecting polynomial basis functions: $\{1, x, x^2, x^3, \ldots\}$; exponential basis functions: $\{1, e^x, e^{2x}, e^{3x}, \ldots\}$; Fourier basis functions: $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots\}$; In practice, according to the need to increase of decrease, increase or decrease the number of population space, i.e., increase or decrease the function type nothing to do with the method itself.

Using the proposed algorithm to test, when the training to reach was given the evolution of algebra, stop training and record the basis function linear combination, and draw the fitness function (i.e., mean square error) of the evolutionary curve and the objective function approximation curve. In which, crossover and mutation probabilities are 0.4 and 0.1. The chromosome string is taken as 5, and the training sample data is randomly generated by the function *Random()*, the sample data of number is not less than 10. The training error is calculated by the fitness formula (8). The selected test functions are as follows: ①$y = x^2$; ②$y = \log(1+e^x)$; ③$y = \sin x + \cos 2x$.

(1) For the test function ①, the best chromosome string 001000 is obtained. Thus, in accordance with the prior agreement of the sorting can know this chromosome string corresponding to the polynomial basis function x^2, and after the proposed algorithm gets and chromosome string corresponding to parameters respectively: 0.185620613472, 0.208013701012, 0.994934689903679, 0.016082726154, 0.1939597712761, 0.586927176159676, The RMSE=2.308080401910119e-004.

That the model is obtained by the proposed algorithm, as follow:

$$\hat{y} = \mathbf{0.994934689903679}x^2 \tag{11}$$

The fitness function evolution curve and the test function approximation curve are shown, in Figure 9, Figure 10.

(2) For the test function ② the best chromosome string 110000 is obtained. Thus, in accordance with the prior agreement of the sorting can know this chromosome string corresponding to the polynomial basis function , and after the proposed algorithm gets and chromosome string corresponding to parameters respectively: 0.669352375369129, 0.623624922225568, 0.00474089042, 0.14970099649, 0.57048373653, 0.24781957054, The *RMSE*=0.002149585692961.

That the model is obtained by the proposed algorithm, as follow:

$$\hat{y} = \mathbf{0.669352375369129} + \mathbf{0.623624922225568}x \tag{12}$$

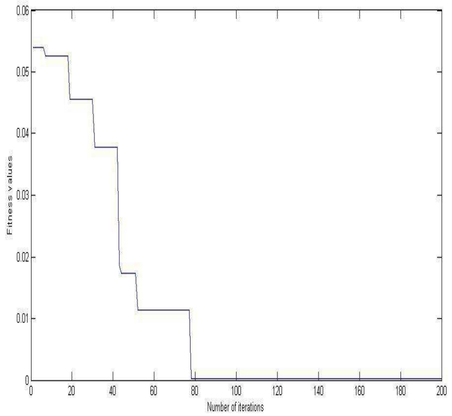


Fig. 9: The fitness function evolution curve

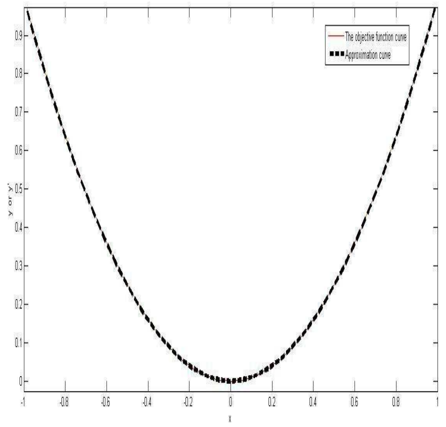


Fig. 10: The test function approximation curve

The fitness function evolution curve and the test function approximation curve are shown, in Figure 11, Figure 12. (3) For the test function ③, the best chromosome string 010010 is obtained. Thus, in accordance with the prior agreement of the sorting can know this chromosome string corresponding to the Fourier basis function $\sin x$, $\cos 2x$, and after the proposed algorithm gets and chromosome string corresponding to parameters respectively: 0.7716262439426, 0.945161987155192, 0.5740817326, 0.474128201, 0.967449759083998, 0.2277986452, The *RMSE*= 0.004037051590859.

That the model is obtained by the proposed algorithm, as follow:

$$\hat{y} = 0.945161987155192\sin x + 0.967449759083998\cos 2x \tag{13}$$

The fitness function evolution curve and the test function approximation curve are shown in Figure 13 and Figure 14.

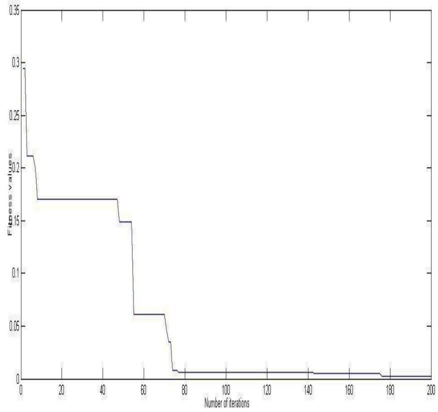


Fig. 11: The fitness function evolution curve

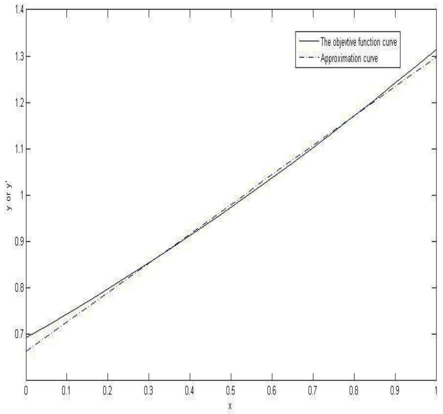


Fig. 12: The test function approximation curve

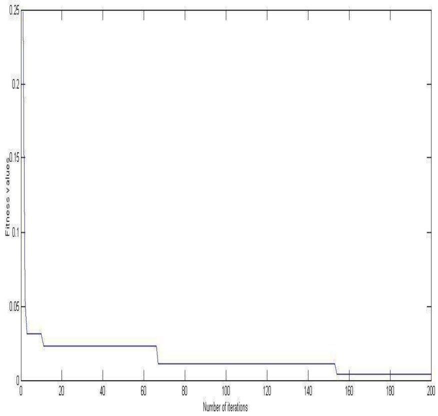


Fig. 13: The fitness function evolution curve

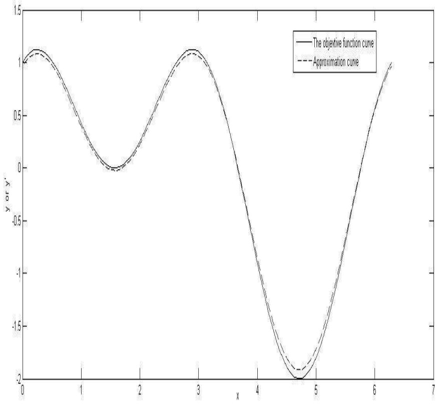


Fig. 14: The test function approximation curve

5.2 Experimental Result Analysis

From equations (11), (12)and (13) model and the 3 test functions compared, it is clear that the proposed algorithm is effective and feasible; and from the Figure 10, Figure 12 and Figure 13 can clearly see that the fitness function(i.e. The root mean square error) of the convergence speed is equivalent fast; whereby Figure10, Figure 12 and Figure 14 gives the test function approximation curve, the solid line as the objective function curve, and the dashed line is the proposed algorithm model. Approximation effect of the proposed algorithm can be seen to be very well, from Figure 10 and Figure 14.

6 Conclusions

In this paper, functional network is regarded as structural and functional parameters of the optimization process, using the basic framework of cultural algorithm, and the introduction of multiple population space and genetic algorithm embedded in each population space in the design of function networks. The proposed algorithm can make full use of the excellent individual contained the population information, and cultural algorithm in practical random search function type space is large, because of cultural algorithm with multiple population space, in the practical application, according to the practical problems that need to be selected, making the generalization performance of the network design is enhanced. In this paper, the proposed algorithm is effective and feasible, but only for the traditional basis function, whereby for the generalized basis function namely hybrid basis function form (i.e.: $x\sin x$, $\sin x\cos x$, $e^x \sin x$, xe^x etc.), this method needs to be improved. The next step is to design the corresponding learning algorithm for optimization functional networks in hybrid basis function.

Acknowledgement

This work was supported in part by NSFC under Grant No. 61165015. Key Project of Guangxi Science Foundation under Grant No. 2012GXNSFDA053028.

References

[1] E. Castillo, Functional networks. Neural Processing Letters, **7**, 151-159 (1998).

[2] E. Castillo, A. Cobo, J.M. Gutierrez and E. Pruneda, Working with differential, functional and difference equations using functional networks, Appl Math Model, **23**, 89-107 (1999).

[3] E. Castillo and J.M. Gutierrez, Nonlinear Time series modeling and prediction using functional networks, Extracting Information Masked by Chaos, Phys. Lett., **A244**, 71-84 (1998).

[4] Y. Q. Zhou, An approximate factorization model of multivariate polynomials based on algebra neural networks and learning algorithm. Journal of Computation Research & Development, **36**, 668-674 (1999) (In Chinese).

[5] E. Castillo, A. Cobo, J. M. Gutierrez and R. E. Pruneda, An Introduction to functional networks with Applications, Kluwer Academic Publishers: New York, (1998).

[6] Y. Q. Zhou, D. D. Wang and M. Zhang, Designing functional networks through evolutionary programming. Proceedings of the 6th World Congress on Intelligent Controland Automation, 21-23 (2006).

[7] R. G. Reynolds, An Introduction to Cultural Algorithms//Proceedings of the Third Annual Conference on Evolutionary Programming, San Diego: World Scientific Press. River Edge, New Jersey, 131-139 (1994).

[8] R. G. Reynolds and S. N. Zhu, Knowledge- based function optimization using fuzzy cultural algorithms with evolutionary programming. IEEE Trans on Systems, Man, and Cybernetics, **31**, 1-18 (2001).

[9] Y. S. Wang, J. B. Ai, Y. J. Shi and H. F. Teng, Cultural-based particle swarm optimization algorithm. Journal of Dalian University of Technology, **47**, 539-544 (2006).

[10] H.Y. Huang, X.S. Gu and M.D. Liu, Research on cultural algorithm for solving nonlinear constrained optimization. ACTA Automatica Sinica, **33**, 1116-1120 (2007).

[11] Y.Q. Zhou and L.C. Jiao, Optimizing neuron function types based on GP in functional network design. Journal of Computer Science, **34**, 7-9 (2007) (In Chinese)

[12] A. Iglesias, B. Arcay and J.M. Cotos, et al. A comparison between functional networks and artificial neural networks for the prediction of fishing catches. Neural computer & applied, **13**, 24-31 (2004).

[13] C. Enrique, S.H. Ali, L. Beatriz and E.P. Rosa, Semi-parametric nonlinear regression and transformation using functional networks. Computational Statistics & Data Analysis, **52**, 2129-2157 (2008).

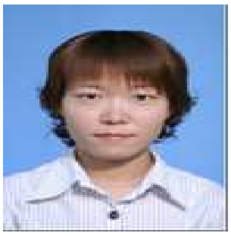
[14] C. Enrique, C. Angel, J. M. Gutirrez and R. E. Pruneda. Functional network with applications. Kluwer Academic Publishers, (1999).

[15] X. Yao, Universal approximation by genetic programming. University of Birmingham, England: Technical Report, http://www.cwi.nl/~bill/fogp/yao.ps.gz, (1999).

[16] M. A. Ahmed and J. K. A. De,Function approximator design using genetic algorithm. In: Proceeding of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), Indianapolis, 519-524 (1997).

[17] Y. Yang and Y. Q. Zhou, A Hybrid Glowworm Swarm Optimization Algorithm for Solving Matrix Eigenvalues, Information-An International Interdisciplinary Journal, **14**, 999-1004 (2011).

[18] Y. Q. Zhou, X. S. Huang, Y. Yang and J. Zh. Wu, Hybrid Optimization Algorithm Based on Mean Particle Swarm and Artificial Fish Swarm. Information-An International Interdisciplinary Journal, **15**, 763-778 (2012).



Qifang Luo, Associate Prof. received his BS degree from Guangxi University of School of Computer and Electronis Information, Guangxi, China, in 1993. He is currently research interest is in computation intelligence, neural networks.

Yongquan Zhou, Ph.D & Prof. He received the MS degree in computer science from Lanzhou University, Lanzhou, China, in 1993 and the Ph.D degree in computation intelligence from the Xiandian University, Xi?an, China, in 2006. He is currently a professor in Guangxi University for Nationalities. His research interests include computation intelligence, neural networks, and intelligence information processing et al. He has published 1 book, and more than 150 research papers in journals.

Peigang Guo, M.S., received his BS degree from Henan University of Science and Technology, Zhengzhou, China, in 2010. He is currently research interest is in computation intelligence, swarm intelligence algorithm.

Xin Chen, M.S., received his BS degree from Anhui University of Science and Technology, Anhui, China, in 2011. He is currently research interest is in computation intelligence, swarm intelligence algorithm.

© 2014 NSP
Natural Sciences Publishing Cor.