# A research on the intelligent multi-objective optimization problem based on wavelet theory and neural networks

**Shunye Wang***

*Department of Mathematic and Computational Science, Langfang Teachers University, Langfang 065000, China*

*Received 1 May 2014, www.cmnt.lv*

**Abstract**

Aiming to solve the multi-objective optimization problem caused by wavelet multiresolution a1nalysis (MRA), the thesis improves the original multi-objective non-dominated genetic algorithm. After fast non-dominated sorting, the evolution of population is achieved through particle swarm optimization (PSO). In this way, the thesis realizes a more effective, organic combination of the multi-objective optimization problem and neural networks. MRA is a natural fit for the multi-objective optimization problem. The ability of neural networks to deal with complex errors is improved through error decomposition based on different wavelet decomposition scales.

*Keywords:* multi-objective optimization problem, wavelet theory, PSO, neural networks

## 1 Introduction

This paper explores the combination of neural network and wavelet theory, focusing on the processing of the noisy data and improves the existing wavelet multi-resolution analysis by combing the learning methods of neural network. The improvements are made on the multi-objective optimization problems caused by the multi-resolution analysis based on the superiority of the non-dominated particle swarm optimization.

Firstly, we turn the problem of predictive modelling into multi-objective optimization problem by using wavelet multi-resolution analysis. Then, we combine the non-dominated quick sorting and particle swarm optimization in dealing with the multi-objective optimization problem. The non-dominated multi-objective particle swarm optimization can converge quickly and it also keeps another advantage of particle swarm, that is, it only needs to know the objective function value instead of the derivative value, thus relaxing the requirements on the objective function. Therefore, the neural network using the parameters of particle swarm optimization network has distinguished advantages. In addition, it is quite easy to realize this kind of combination and it just needs to set the link weights and the output thresholds as the position vectors of the particle, turning the problem into the wavelet multi-resolution expression for the study errors. The network thresholds and weights iterate according to particle swarm optimization and search the optimal value. In this way, it not only avoids the neural network from making assignments randomly to the weights and thresholds in the initial stage, but it also avoids considering the error back propagation as a method to revise the rules so as to trap the network into the local minima values; therefore, it enhances the network stability. The adjustments to the parameters of the wavelet neural network depends on the continuous search of the particle and indirectly determine the various link weights and thresholds of the neural network by updating the velocity and position vectors of the particle. Finally, the optimal value of the parameters of the wavelet neural network is determined by recording the optimal position of the particle.

## 2 Wavelet multiresolution analyses

Wavelets are used in a wide range of applications such as signal analysis, signal compression, finite element methods, differential equations, and integral equations. In the following we will discuss the limitations of traditional basis expansions and show why wavelets are in many cases more efficient representations. A mathematical treatment of second generation wavelets as well as an example will be provided.

1) Localization in space The Fourier transform is localized in frequency but the global support of the basic functions prevents localization in space. For many applications in particular the local behaviour of signals is of interest.

2) Faster transform algorithms. In recent years, the advance of data acquisition technology outpaced the available computing power significantly making the Fast Fourier Transform with its O (n log n) complexity a bottleneck in many applications.

3) More flexibility Traditional basis expansions provide no or almost no flexibility. It is therefore usually not possible to adapt a representation to the problem at hand. An important reason for this lack of flexibility is the orthogonal nature of traditional basis expansions.

* ***Corresponding author*** e-mail: Shunyew@163.com

4) Arbitrary domains Traditional basis representations can only represent functions defined of Euclidean spaces $R^{n2}$. Many real-world problems have embeddings $X \subset R^n$ as domain and it is desirable to have a representation which can be easily adapted for these spaces.

5) Weighted measures and irregularly sampled data traditional transforms can usually not be employed on spaces with weighted measures or when the input data is irregularly sampled.

These limitations motivated the development of wavelets. Many different fields such as applied mathematics, physics, signal processing, and computer science provided contributions and today both a thorough mathematical theory and fast and practical algorithms exist [1].

Wavelets can be categorized into discrete (DWT) and continuous (CWT) wavelet transforms. To speak in broad terms, the basic functions of DWTs are defined over a discrete space which becomes continuous only in the limit case, whereas the basic functions of CWTs are continuous but require discretization if they are to be used on a computer; see for example the book by Antoine et al. for a more detailed discussion of the differences. In signal compression applications mostly discrete wavelets are employed, whereas for signal analysis typically continuous wavelets are used [2].

## 2.1 CONTINUOUS WAVELET TRANSFORM

The Continuous Wavelet Transform (CWT) is a time-frequency representation of signals that graphically has a superficial similarity to the Wigner transform.

A wavelet transform is a convolution of a signal s(t) with a set of functions which are generated by translations and dilations of a main function. The main function is known as the mother wavelet and the translated or dilated functions are called wavelets. Mathematically, the CWT is given by:

$$W(a,b) = \frac{1}{\sqrt{a}} \int s(t) \psi \left( \frac{t-b}{a} \right) dt \,, \tag{1}$$

where $b$ is the time translation and $a$ is the dilation of the wavelet.

From a computational point of view it is natural to use the FFT to compute the convolution which suggests that the results are dependent on proper sampling of $s(t)$.

When the mother wavelet is complex, the CWT is also a complex valued function. Otherwise the CWT is real. The squared magnitude of the CWT $|W(a,b)|^2$ is equivalent to the power spectrum so that a typical display (image) of the CWT is a representation of the power spectrum as a function of time offset $b$. One should note however that the precise form of the CWT depends on the choice of mother wavelet y and therefore the extent of the equivalency between the squared magnitude of the CWT and the power spectrum is application dependent.

The CWT operation is implemented using both the FFT and the discrete sum approach. You can use either one to get a representation of the effective wavelet using a delta function as an input [3].

## 2.2 DISCRETE WAVELET TRANSFORM

The Discrete Wavelet Transform (DWT) is similar to the Fourier transform in that it is a decomposition of a signal in terms of a basis set of functions. In Fourier transforms the basis set consists of sines and cosines and the expansion has a single parameter. In wavelet transform the expansion has two parameters and the functions (wavelets) are generated from a single "mother" wavelet using dilation and offsets corresponding to the two parameters.

$$f(t) = \sum_a \sum_b c_{ab} \psi_{ab}(t) \,, \tag{2}$$

where the two-parameter expansion coefficients are given by:

$$c_{ab} = \int f(t) \psi_{ab}(t) dt \tag{3}$$

and the wavelets obey the condition:

$$\psi_{ab}(t) = 2^{\frac{a}{2}} \psi(2^a t - b) \,. \tag{4}$$

A wavelet is a wave-like oscillation with an amplitude that begins at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a seismograph or heart monitor. Generally, wavelets are purposefully crafted to have specific properties that make them useful for signal processing. Wavelets can be combined, using a "reverse, shift, multiply and integrate" technique called convolution, with portions of a known signal to extract information from the unknown signal [4].

For example, a wavelet could be created to have a frequency of Middle C and a short duration of roughly a 32nd note. If this wavelet was to be convolved with a signal created from the recording of a song, then the resulting signal would be useful for determining when the Middle C note was being played in the song. Mathematically, the wavelet will correlate with the signal if the unknown signal contains information of similar frequency. This concept of correlation is at the core of many practical applications of wavelet theory.

As a mathematical tool, wavelets can be used to extract information from many different kinds of data, including - but certainly not limited to – audio signals and images. Sets of wavelets are generally needed to analyse data fully. A set of "complementary" wavelets will decompose data without gaps or overlap so that the decomposition process is mathematically reversible. Thus, sets of complementary wavelets are useful in wavelet based compression/decompression algorithms where it is desirable to recover the original information with minimal loss [5].

In formal terms, this representation is a wavelet series representation of a square-integrable function with respect to either a complete, orthonormal set of basis functions, or an over complete set or frame of a vector space, for the Hilbert space of square-integrable functions.

Wavelet theory is applicable to several subjects. All wavelet transforms may be considered forms of time-frequency representation for continuous-time (analogue) signals and so are related to harmonic analysis. Almost all practically useful discrete wavelet transforms use discrete-time filter banks. These filter banks are called the wavelet and scaling coefficients in wavelets nomenclature [6]. These filter banks may contain either finite impulse response (FIR) or infinite impulse response (IIR) filters. The wavelets forming a continuous wavelet transform (CWT) are subject to the uncertainty principle of Fourier analysis respective sampling theory: Given a signal with some event in it, one cannot assign simultaneously an exact time and frequency response scale to that event. The product of the uncertainties of time and frequency response scale has a lower bound. Thus, in the scale gram of a continuous wavelet transform of this signal, such an event marks an entire region in the time-scale plane, instead of just one point. Also, discrete wavelet bases may be considered in the context of other forms of the uncertainty principle.

Wavelet transforms are broadly divided into three classes: continuous, discrete and multiresolution-based.

As an important part of wavelet analysis, MRA is widely applied especially to image processing. It is known to all that MRA can be applied to neural networks. To put it simply, the object information described by MRA is decomposed into a series of infinite orthogonal spaces. Thus comes the problem - which spaces (or degrees) of object information do people want to acquire? Below we analyse the concept of MRA. Its expression is composed of closed subspaces $V_j, W_j \in L^2(R)$ that meet the following conditions [7]:

1) $V_j = \text{span}\left\{\phi_{j,k} \in Z\right\}, j \in (0,+\infty)$,

2) $V_j = \text{span}\left\{\psi_{j,k}, j,k \in Z\right\}$,

3) $V_{j+1} = V_j \oplus W_j$,

where $x$, $z$, $R$, and $f(\cdot)$ stands for one-dimensional system input, the set of integers, the set of all real numbers, and system model function, separately. $V_j$ and $W_j$ constitutes a mutually orthogonal space. The respective orthogonal basis within the two can be described as $\phi_{j+1}(x) = 2^{-j/2}\phi(2^j x - k)$ and $\psi_{j,k}(x) = 2^{-j/2}\psi(2^j x - k)$. Meanwhile, the respective base of the mutually orthogonal space represents the wavelet scale and the wavelet function.

$f(x)$ can be expressed as follows if it meets the condition $f(x) \in L^2(R)$:

$$f(x) = \sum_{k \in Z} \phi_{j_{\max},k}, f > \phi_{j_{\max},k}(x) + \sum_{j_{\max} \geq j \geq j_0, k \in Z} < \psi_{j,k}, k > \psi_{j,k}(x), \quad (5)$$

where $j_{\max}$ and $j_0$ means the upper limit and the lower limit of resolution; $< ..., ... >$ represents the inner product in the orthogonal space of the function. Theoretically, it is possible to realize infinite approximation to a function. As shown in Equation (5), a signal is composed of two parts—scaling function and wavelet function. We divide them into the approximation part and the detailed part [8, 9].

## 3 Non-nominated multi-objective particle swarm optimization

The multi-objective particle swarm optimization constructs non-dominated solution set based on Pareto dominance relation and preserves the non-dominated solution set currently found by using archive. To use ε dominance concept to update the archive can give the algorithm with good distributive; fasten the convergence of the solutions by improving the selection methods of global extremum and individual extremum of the algorithm and adopting new particle update strategies; add in self-adaptive mutation particle to avoid being trapped into local Pareto optimal solution and propose non-dominated set construction method based on quicksort to accelerate the operational efficiency of the algorithm[8].

### 3.1 THE MAIN PROCESSES OF MULTI-OBJECTIVE PARTICLE SWAM OPTIMIZATION

The main framework of the algorithm is as follows:

Step 1. Initialize the population.

Step 2. Make iteration loop till the Maxgen.

1) Update the velocity of every particle in the particle swarm according to Equation (6). In Equation (6), the constants $C_1$ and $C_2$ control the effect the individual extremum and the global extremum play on the particle update. Besides, the parameter $W$ is named the inertia weight and $R_1$ and $R_2$ are the random numbers among [0,1].

$$Vel[i] = W \times Vel[i] + C_1 R_1 \left(P_{best}[i] - Pop[i]\right) + C_2 R_2 \times \left(G_{best}[i] - Pop[i]\right). \quad (6)$$

2) According to the update velocity of every particle obtained from the last step; recalculate the new variable value based on Equation (7) and replace the particle with the individual extremum if the new position is dominated by the individual extremum of this particle.

$$Pop[i] = Pop[i] + Vel[i]. \quad (7)$$

Step 3. Implement mutation operation on the particle swarm with self-adaptive mutation probability.

Step 4. Calculate the objective function value of every particle.

Step 5. Solve the non-dominated set *Npop*: find all the non-dominated particles in the non-dominated set through quicksort.
Step 6. Update the individual extremum. Here dominance concept is used. As for every particle in the swarm, if the current position of particle *i* dominates its individual extremum position, then update its individual extremum.

$$P_{best}[i] = Pop[i]. \tag{8}$$

Step 7. Update the archive Rpop, insert the non-dominated set of the swarm in the archive based on the $\varepsilon$-dominance relation and use $\varepsilon$-dominance concept in the update strategy.
Step 8. Update the global extremum and select one particle from the archive *Rpop* as the global extremum of particle *i*.

The archive *Rpop* preserves the optimal result of every generation of operation and after the algorithm is over, all the particles in *Rpop* is the final results of the algorithm [9].

## 3.2 THE KEY OPERATORS OF MULTI-OBJECTIVE PERTICLE SWARM OPTIMIZATION

### *3.2.1 Archive*

The purpose to set an archive apart from the swarm is:

a) To preserve the non-strong ε- domination found in the iteration so as to lead the algorithm to get closed to the Pareto optimal region more quickly;

b) To exist as the candidate set of the global extremum of every particle in the swarm;

c) The solution set preserved by the archive is the result of the algorithm.

Therefore, in order to make the archive play the dominant role and the final solution set preserve excellent distributive, it should be considered how to make the externally-centralized particles distributed uniformly in the archive update. After some analysis, we adopt $\varepsilon$-dominance relation. $\varepsilon$-dominance is defined:

Make $f$, $g \in \mathbb{R}^{+m}$; name $f$ $\varepsilon$-dominance $g(\varepsilon > 0)$; mark $f \succ_{\varepsilon} g$, if:

$$\forall i \in \{1,...m\} : \lfloor f_i / \varepsilon_i \rfloor \le \lfloor g_i / \varepsilon_i \rfloor, \tag{9}$$

$$\forall i \in \{1,...m\} : \lfloor f_j / \varepsilon_j \rfloor \le \lfloor g_j / \varepsilon_j \rfloor. \tag{10}$$

Figure 1 demonstrates the concepts of dominance relation and ε-dominance relation. It can be seen that in ε-dominance relation, the solution space is divided into w meshes and the dominance domain of the individual f expands than the normal dominance domain. Use it in the multi-objective particle swarm optimization to make the solution set preserve better distribution.

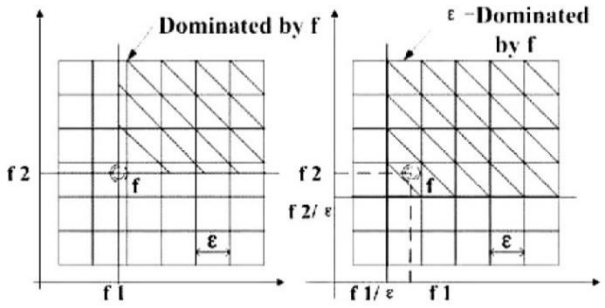

FIGURE 1 Dominance and ε-Dominance

### *3.2.2 Individual update*

When the particle uses Equations (6) and (7) recalculates its decision variable, the particle actually has found a new solution and a new position in the objective function space.

In the experiment, we find the test problems in the local Pareto optimal domain. During the algorithm operation, only a few particles of every generation can jump out from the local Pareto optimal domain and they are still trapped into the local solutions as a whole. More cycle indexes are needed to converge to the global Pareto optimal domain or it may not converge at all. In order to solve this problem and accelerate the convergence, we will compare its new position with the position of the individual extremum when we update the individual position:

a) If the new position is dominated by the individual extremum of this particle, then replace the particle with its individual extremum;

b) Otherwise, it remains unchanged in the new position.

The above comparison and revise make the entire swarm jump out of the local Pareto optimal domain in time and ensure that the whole swarm converges quickly to the global Pareto optimal domain [10].

### *3.2.3 The Selection of the individual extremum and the global extremum*

The selection of the individual extremum and global extremum are greatly important in the particle swarm optimization and it will directly affects the convergence rate of the algorithm and the distributivity of the solution set for the multi-objective optimization problem.

As for the individual extremum, the selection of the individual extremum is different from the single-objective optimization in the multi-objective optimization problem and we will judge the optimal position a particle finds based on the dominance relation.

In terms of the global extremum, the archieve *Rpop* is our candidate set. When starting the algorithm, set the global extremum of every particle as itself and conduct global extremum update on every generation. Randomly select a particle in the archive for every particle *i* in the swarm. If this candidate particle dominates the global extremum of the particle *i*, then set the candidate particle as the new global extremum of the particle *i*; otherwise, we

implement global extremum mutation on particle. As a matter of fact, we have proposed the extremum mutation method, namely set an extremum mutation probability. Produce a random number among [0, 1] for particle *i*. If the value of the random variable is smaller than the extremum mutation probability, then randomly select a particle from the archive to replace the current global extremum of the particle *i*.

*3.2.4 The Design of the Disturbance Operators*

One characteristic of particle swarm optimization is that it has fast convergence; however, it may also lead to being trapped in the local optimal solution; therefore, disturbance is needed to be implemented on the particles to make them jump out of the local optimal domain.

We have designed something similar to the mutation operators in the genetic algorithm to solve this problem. In the early operation of the algorithm, search the entire objective space. With the continuous evolution of the particles in the swarm, it needs to reduce the number of particles participating in the mutation. The mutation probability *Pmutation* is calculated according to Equation (11) and *Currentgen* is the current operational algebra.

$$Pmutation = 1 - Currentgen / Maxgen . \quad (11)$$

For every particle in the swarm, the variable *m_random* is given a random number among [0,1]. If *m_random* is smaller than the mutation probability *Pmutation*, then conduct non-uniform mutation to that particle, namely to make mutation randomly on one of the n-dimensional decision vectors according to Equation (12). The value of θ represents that the forward direction of the mutated particle is the same as or opposite to its original direction; μ stands for the scope of velocity change and the random value of θ is ±1. In order to jump out from the local optimal domain through mutation, the velocity of the particle must be accelerated and μ should be bigger than 1. In the experiment, we set: $\mu$=3.

$$Pop[i] = \theta \times \mu \times (1 - m\_random) Vel[i] + Pop[i] . \quad (12)$$

If the mutation operation makes the decision variable exceed the definition domain, then set that decision variable in the boundary of its definition domain.

*3.2.5 Construct non-dominated set through QuickSort*

The construction of non-dominated set is a significant step in the multi-objective evolutionary algorithm. In order the increase the efficiency to construct non-dominated set, we have proposed to construct non-dominated set algorithm based on quicksort. Select an individual *x* (we usually select the first individual) from the swarm in every cycle of the algorithm and compare the other individuals with *x*. Divide the swarm into two parts through one comparison: the latter part is the individuals dominated by *x* while the former part is the individuals dominating *x* or irrelevant to *x*. If *x* is not dominated by any individual, then integrate *x* into the non-dominated set and repeat the above process to the former part until it is empty.

Figure 2 displays the position changes of the particles in the swarm after a comparison. The particles before 1 is the particles not dominated by it while those after it are dominated by it obviously, only the front particles can enter into the non-dominated set. In this way, it narrows down the comparison range and saves expense.

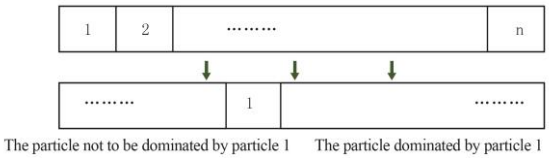


FIGURE 2 Construct Non-dominated Set Based on QuickSort

## 4 Learning algorithm combing wavelet and neural network

### 4.1 WAVELET NEURAL NETWORK OF BP LEARNING ALGORITHM

The hidden layer neuron of wavelet neural network uses wavelet-based function as its network's activation function, which introduces scaling factor and translation parameter to the network in order to substitute original connection weight and output threshold, i.e. connection weight replaced by translation factor and output threshold replaced by scaling factor. Based on BP algorithm i.e. error back propagation algorithm, the training data in network is inputted to the neural network from input layer and then reach output layer via hidden layer, which is also called "pattern forward propagation"; the output value of neural network is compared with the ideal output and then the weights are adjusted starting from the output layer based on the thought of error reduction, which is called "error back propagation". The error accumulation after layer-by-layer transmission are just the signals that we want to catch at last and according to which each weight and threshold are adjusted in layer-by-layer inversed sequence based on error reduction[11]. The following example is a three-layer wavelet neural network (i.e. input, and output hidden layers), in which gradient descent method is used to find out errors and weights in order to get the variation of network weights and then the network is optimized by minimum mean square error objective function to adjust network weights and wavelet coefficient adaptively. The model of wavelet neural network is expressed as:

$$y_i^p(t) = f\left[\sum_{j=1}^{N} w_{ij}(t) \Psi_{a_j,b_j}\left(\sum_{k=1}^{m} w_{jk}(t) x_k^p(t)\right)\right] net_j^p = \sum_{k=1}^{m} w_{jk}(t) x_k^p(t), \quad (13)$$

$$\Psi_{a_j,b_j} net_j^p(t) = \psi\left[\frac{net_j^p(t) - b_j(t)}{a_j(t)}\right], \tag{14}$$

$$E(t) = \frac{1}{2}\sum_{p=1}^{M}\sum_{i=1}^{n}\left[d_i^p - y_i^p(t)\right]^2, \tag{15}$$

where $t$ is iterative steps, $x_k^p$ is the *k-th* input value of the *p-th* sample of input layer, $y_i^p(t)$ is the *i-th* output value of the *p-th* sample of output layer, $d_i^p$ is the *i-th* desired output of the *p-th* sample, $\psi(x)$ is basic wavelet function, $E(t)$ is objective function and $f(x)$ is the purlin of output layer. In this paper, it is supposed that $f(x)=x$, $w_{ij}(t)$ is the weight connecting the node j of hidden layer and the node $i$ of output layer, $w_{jk}(t)$ is the weight connecting the node $k$ of input layer and the node j of hidden layer, $a_j(t)$ and $b_j(t)$ are the scaling and translation coefficients of the node *j-th* of hidden layer respectively, m is the number of nodes of input layer, $N$ is the number of the nodes of hidden layer, $M$ is the number of input samples and n is the number of the nodes of output layer. In this way, after the connection weight $w$, threshold $b$ and the scaling factor and translation parameter $a$, $b$ of wavelet network are initialized and the input information is propagated forward, the actual ouput of the wavelet neural network can be calculated using current network parameters and the network can be optimized as follows:

$$\frac{\partial E(t)}{\partial w_{jk}(t)} = \frac{-\sum_{p=1}^{M}\sum_{i=1}^{n}\left(d_i^p - y_i^p(t)\right)w_{ij}(t)\Psi'_{a_j,b_j} net_j^p(t)\, x_k^p(t)}{a_j(t)}, \tag{16}$$

$$\frac{\partial E(t)}{\partial w_{ij}(t)} = -\sum_{p=1}^{M}\left(d_i^p - y_i^p(t)\right)\Psi_{a_j,b_j}\left(net_j^p(t)\right), \tag{17}$$

$$\frac{\partial E(t)}{\partial a_j(t)} = \frac{-\sum_{p=1}^{M}\sum_{i=1}^{n}\left(d_i^p - y_i^p(t)\right)w_{ij}(t)\Psi'_{a_j,b_j}\left(net_j^p(t)\right)\left(\frac{net_j^p(t) - b_j(t)}{a_j(t)}\right)}{a_j(t)}, \tag{18}$$

$$\frac{\partial E(t)}{\partial w_{ij}(t)} = \frac{-\sum_{p=1}^{M}\sum_{i=1}^{n}\left(d_i^p - y_i^p(t)\right)w_{ij}(t)\Psi'_{a_j,b_j}\left(net_j^p(t)\right)}{a_j(t)}, \tag{19}$$

Parameters adjustment equations are the following:

$$w_{jk}(t+1) = w_{jk}(t) - \eta\frac{\partial E(t)}{\partial w_{jk}(t)} + \mu\Delta w_{jk}(t), \tag{20}$$

$$w_{ij}(t+1) = w_{ij}(t) - \eta\frac{\partial E(t)}{\partial w_{ij}(t)} + \mu\Delta w_{ij}(t), \tag{21}$$

$$a_j(t+1) = a_j(t) - \eta\frac{\partial E(t)}{\partial a_j(t)} + \mu\Delta a_j(t), \tag{22}$$

$$b_j(t+1) = b_j(t) - \eta\frac{\partial E(t)}{\partial b_j(t)} + \mu\Delta b_j(t), \tag{23}$$

where, $\mu$ is a momentum coefficient and also the learning rate of wavelet neural network. Because of the use of derivative, BP algorithm is very dependent on objective function, which is to say, in case of different objective functions, the training process of BP algorithm is also unavailable even if the network structure is the same [12].

## 4.2 WAVELET NEUTRAL NETWORK BASED ON MULTI-SUBSWARM LEARNING ALGORITHM

### *4.2.1 Multi-subswarm particle swarm optimization algorithm*

The main steps based on multi-subswarm particle swarm optimization algorithm include the following:

Step 1. Set up parameter set and assign values to particle swarm randomly.

Step 2. Calculate the fitness values of all current particle $S$ and divide them into two subswarms according to the fitness value.

Step 3. The two subswarms evolve according to their own strategies:

a) Each particle of the subswarm $S_2$ breeds some optional off springs according to the pre-set growth rate $\rho_2$, the fitness function values of those optional offsprings are calculated, and the top $L$ particles from higher fitness value to lower fitness value evolve further and other optional particles are not selected;

b) Each particle of the subswarm $S_1$ breeds an offspring, the $H$ off springs are selected for further evolution and their fitness function values are calculated.

Step 4. The results of the evolution of the two subswarms are combined and the optimal value is updated.

Step 5. Whether to meet END condition is checked. If one of the following is met, i.e the biggest preset evolution algebra and the requirement for accuracy, calculation ends; otherwise skip to Step 2.

It is supported that $S=\{X_i, i=1,2,L\ N\}$ represents the subswarm of $N$ particles and $X_i$ is the *i-th* particle of the subswarm, so the fitness values of all the particles are arranged in ascending order as follows:

$$F=\{(f_1, f_2, L\ \ f_N)\,|\, f_1 \le L \le f_k \le L\ \ f_N\}. \tag{24}$$

$S_1 \cup S_2 = S, S_1 = \Phi$ . $H+L=N; S_1, S_2$ are the subswarms of $H$ and $L$ particles respectively where: $H=N-[L_{\max} \times rand()]$. $L_{\max}$ represents the biggest number of particles in $S_2$ and *rand*() is the random number of [0,1]. $p_2=\frac{l_{\max}}{N}$ result shows that the value range is 0.02~0.10 to the best.

In consideration of the number of the offsprings of subswarms, this paper introduces the growth rate $\rho_1$, so the scale $H_1$ of the offsprings of subswarm $S_1$ can be expressed as: $H_1 = \rho_1 \times H$

Quantities of experiments show that it is proper to set the growth rate $\rho_1$ of subswarm $S_1$ to be 1, so there is: $H_1 = \rho_1 \times H \rho_1 = 1$.

The evolution of subswarm $S_1$ is subject to the evolution process of basic particle swarm algorithm, i.e. each particle only has one offspring and all the off springs are reserved as the parents of the next generation. The evolution of subswarm $S_2$ is comparatively complicated, mainly because local optimum is avoided based on this subswarm and each particle needs to produce several off springs. In convenience for description, $H_2$ is the scale of the off springs of subswarm $S_2$ and $\rho_2$ is the growth rate of subswarm $S_2$, so there is: $H_2 = \rho_2 \times L$, $\rho_2 \ge 1$.

### *4.2.2 Wavelet neural network based on multi-subswarm learning algorithm*

The number of hidden layer neurons is closely related to the approaching ability of network, that is, the more number of neurons, the better of the approaching property of the network. However, in reality, other factors need to be considered like the complexity, so it does not mean the more the better. Therefore, in terms of wavelet neural network, if it can adjust the number of its own neurons dynamically and adaptively, it will increase the scope of application of wavelet neural network. As the learning process of wavelet neural network can be treated as the adjustment process of hidden layer neuron parameter, the position vector of each particle can be defined as:

$$present(i)=[w_i, a_i, b_i, \lambda_i], i=1,2,L,N. \tag{25}$$

The model of optimizing wavelet neural network with multi-subswarm particle swarm is the following:

1) Determine particle dimensions.

In case of one neuron, 4 variable parameters can be used to determine one network output and all the parameters that can be used to determine one ouput are defined as one particle. In this way, if there are N hidden layer neurons, the network will need 4N parameters. As the number of parameters corresponds to the dimensions of searching space, particle dimensions are 4N dimensions.

2) Initialize particle subswarm.

Initialize particle subswarm with random value assignment before network training.

3) Calculate the output of neural network.

Standardized processing is made to the training data sample of wavelet neural network and Mexican Hat Function is used as the activation function of hidden layer neurons. The position and velocity vectors are updated with multi-subswarm particle swarm optimization algorithm and then the actual ouput value and error function value of network can be obtained according to particle position x and training sample.

4) Judge generation quantity.

If the upper limit of generation quantity is not reached, particle swarm shall be updated, and step (3) is taken, or further step can be taken;

5) Judge whether to stop training.

If the optimal particle cannot meet desired error threshold, further step can be taken, or the process shall end;

6) Judge whether to adjust structure.

If the particles near the upper limit of generation quantity can't meet desired variation, i.e. the particles only vary with a small range, the number of neurons shall be increased and Step 2 shall be taken; or, without adjustment of network structure, the particle swarm can be updated and Step 3 can be taken;

The training process of multi-subswarm structure particle swarm wavelet neural network is as follows:

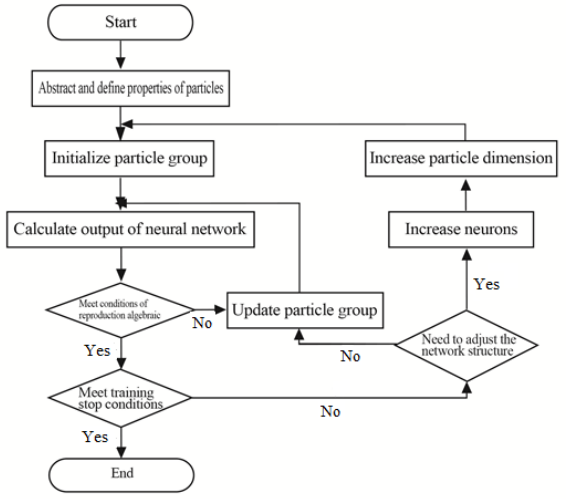


FIGURE 3 The Training Process of Multi-subswarm Particle Swarm-optimized Wavelet Neural Network

## 5 Simulation examples

The thesis chooses Mackey-Glass and the sunspot chaotic time series as the simulation objects. Normalization processing is still an essential step before the algorithm begins. The processing standards should be conducive to the effective implementation of algorithms. Normalization is also closely related to the allocation of wavelet nodes. In this simulation, the wavelet nodes are strictly limited to the domain of definition $[0,1]^n\ (n<4)$.

### 5.1 LOGISTIC SYSTEM

Under the condition of $0 \le t \le \tau$, make $x(t) = 0.9$. The integration step is 0.01, the time delay $\tau$ is 30, and the sampling time interval is 0.06. There are 1,000 data points to be tested and trained. We can select from the generated data at random. Add Gaussian noise of 10% and 20% respectively to the original samples to yield noise data.

The Equation of the Logistic system is:

$$x_{i+1} = 4x_i(1-x_i). \tag{26}$$

We continue the one-step prediction and divide the 1,000 sample points into two parts. The first half is used to train the network, and the second half is used to test the network. According to literature, expand the wavelet network for simulation based on ANOVA. Set the embedded dimension of phase space at 6, where $x_{l_i} = y(t-l_i), i=1,2,3, 1 \le l_1 \le l_2 \le l_3 \le 6$. The note function of the network is still one-, two-, or three-dimensional. Besides, its number and exponents decrease progressively $(g_{l_1}^{[1]}(\cdot), g_{l_1 l_2}^{[2]}(\cdot), g_{l_1 l_2 l_3}^{[3]}(\cdot))$. Equations (28) and (29) respectively stand for the one-dimensional form and the multi-dimensional form of the wavelet.

The network model is as follows:

$$\begin{aligned} y(t) &= g(x_1,x_2,x_3,x_4,x_5,x_6) = \\ &\sum_{1\le l_1\le 6} g_{l_1}^{[1]}(x_{l_1}) + \sum_{1\le l_1\le l_2\le 6} g_{l_1 l_2}^{[2]}(x_{l_1},x_{l_2}) = , \\ &\sum_{1\le l_1\le l_2\le l_3\le 6} g_{l_1 l_2 l_3}^{[3]}(x_{l_1},x_{l_2},x_{l_3}) \end{aligned} \tag{27}$$

$$\psi^{[1]}(x) = (1-x^2)e^{-\frac{1}{2}x^2}, \tag{28}$$

$x$ is one-dimensional input.

$$\psi^{[n]}(x) = (n-\|x\|)e^{-\frac{1}{2}\|x\|^2}, \tag{29}$$

$x$ is multi-dimensional input.

We train the network by combining fast non-dominated sorting and PSO. The optimization objective is described as:

$$\|e_1(x)\| = \left\| \sum_{k\in Z} \phi_{1,k}, e_0 > \phi_{1,k}(x) \right\|. \tag{30}$$

Based on the above objective function form, we can obtain several objective functions by changing the scale of wavelet resolution. Here we adopt two scales starting from the original scale and get two objective functions. Below are the experiment results.

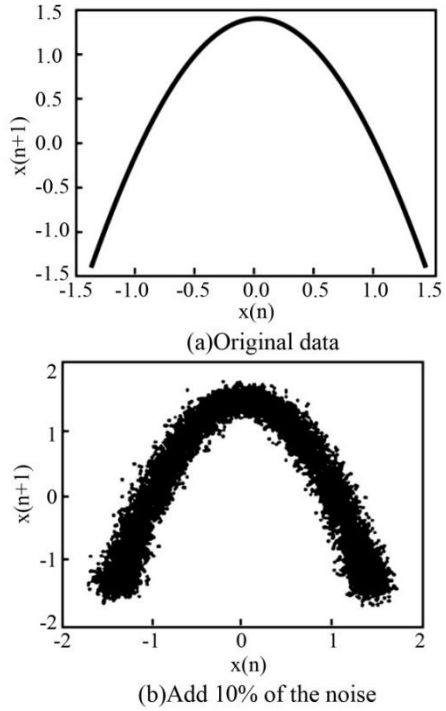


FIGURE 4 Two-dimensional reconstructed attractor of Logistic map

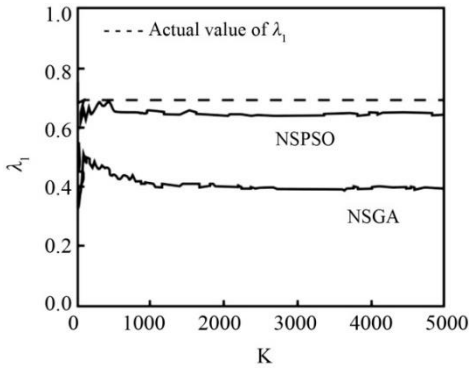


FIGURE 5 The convergence of Lyapunov spectrum of Logistic map time series with 10% noise

Figures 4a and 4b respectively represent the two-dimensional reconstructed attractor before the addition of nosie and after the addition of 10% noise. We can know that Logistic mapping is a parabola in two-dimensional phase space, but the results become a "parabolic band" when it contains noise.

From the simulation results of NSGA, and NSPSO, as the noise increases, the advantage of multi-resolution algorithm becomes more obvious. This is mainly because the multi-objective optimization of multiresolution learning algorithm can cover noise of more frequencies. More importantly, it is notable that the resolution of PSO is better than that of the genetic algorithm.

### 5.2 HÉNON SYSTEM

Secondly, calculate the time series generated by Hénon map, Figures 4a and 4b respectively represent before the addition of noise and after the addition of 15% noise. As shown, we find that the latter one made the reconstructed attractor lose the detailed traits completely. Besides, we take the measures of NSGA as well as NSPSO so as to calculate the time series of the particles containing noise. Under the condition of $\Delta t = 1$, $N_s = 5 \times 105$, $\tau = 1$, $d = 2$, $\lambda_1 = 0.4185$, $\lambda_2 = -1.6225$, add Gaussian noise of 15% to the original samples to yield noise data.

The equation of Logistic system is:

$$\begin{aligned} x_{i+1} &= 4x_i(1-x_i) \\ y_{i+1} &= 0.3x_i \end{aligned}, \tag{31}$$

The wavelet function in the simulation example is Gauss wavelet. The optimization algorithms are the combination of fast non-dominated sorting and heredity as well as particle swarm. The network model and the fitness function expression of non-dominated PSO are as follows:

$$y(t) = g(x) = g\left(y(t-1), y(t-2), ..., y(t-9)\right), \tag{32}$$

$$\|e_0\| = \|y(t) - g(x)\|, \tag{33}$$

$$e_1(x) = \sum_{k \in Z} \langle \phi_{1,k}, e_0 \rangle \phi_{1,k}(x) \tag{34}$$

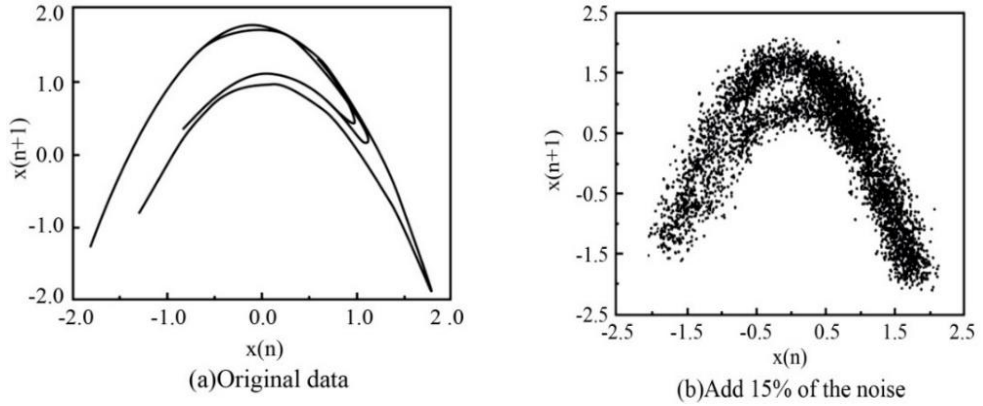


FIGURE 6 Reconstructed attractor of Hénon map

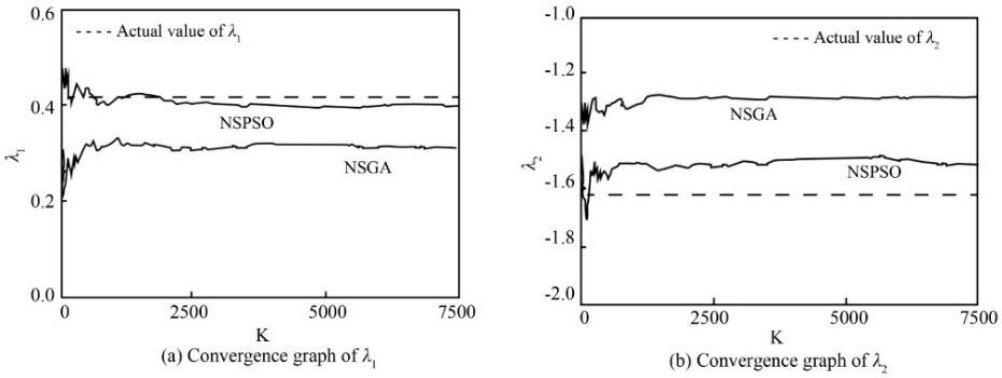


FIGURE 7 The convergence of Lyapunov spectrum of Hénon map time series with 15% noise

Figure 7 indicates that the introduction of MRA greatly improves the ability of wavelet network learning and noisy data prediction. It obviously proves the louder the noise, the higher the efficiency of MRA algorithms. Thus, MRA has more important practical applications.

## 6 Conclusions

The thesis makes a research on noisy data processing through the combination of neural networks, particle swarm optimization and wavelet theory so as to improve the multi-objective optimization problem caused by multiresolution analysis. The improvement is based on the superiority and popularity of PSO over genetic algorithms. Firstly, the thesis converts the predictive modelling problem to the multi-objective optimization problem by dint of MRA. After that, it adopts the training method integrating fast non-dominated sorting and PSO to deal with the multi-objective optimization problem. The simulation results of the experiment shows, the algorithm proposed in the thesis has an advantage when handling the problems of multi-objective optimization and errors.

## References


[1] Al-Geelani N A, Piah M A M, Adzis Z, Algeelani M A 2013 Hybrid regrouping PSO based wavelet neural networks for characterization of acoustic signals due to surface discharges on H.V. glass insulators *Applied Soft Computing* **13**(12) 4622-32

[2] Liu H, Tian H, Chen C, Li Y 2013 An experimental investigation of two Wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization *International Journal of Electrical Power & Energy Systems* **52** 161-173

[3] Liu Z, Cao H, Chen Z, He Z, Shen Z Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings *Neurocomputing* **99** 399-410.

[4] Chatterjee A, Ghoshal S P, MukherjeeV 2011 Craziness-based PSO with wavelet mutation for transient performance augmentation of thermal system connected to grid *Expert Systems with Applications* **38**(6) 7784-94

[5] Tsai H, Jhuang Y, Lai Y 2012 An SVD-based image watermarking in wavelet domain using SVR and PSO Original Research Article *Applied Soft Computing* **12**(8) 2442-53

[6] Shrivastava N A, Panigrahi B K 2014 A hybrid wavelet-ELM based short term price forecasting for electricity markets *International Journal of Electrical Power & Energy Systems* **55** 41-50

[7] Daamouche A, Hamami L, Alajlan N, Melgani F 2012 A wavelet optimization approach for ECG signal classification *Biomedical Signal Processing and Control* **7**(4) 342-9.

[8] Kusetogullari H, Leeson M S, Kole B, Hines E L 2014 Meta-heuristic algorithms for optimized network flow wavelet-based image coding. *Applied Soft Computing* **14** Part C 536-53

[9] Guo J, Ding L, Luo H, Zhou C, Ma L 2014 Wavelet prediction method for ground deformation induced by tunneling *Tunnelling and Underground Space Technology* **41** 137-151

[10] Katunin A, Przystałka P 2014 Damage assessment in composite plates using fractional wavelet transform of modal shapes with optimized selection of spatial wavelets *Engineering Applications of Artificial Intelligence* **30** 73-85

[11] Subasi A 2013 Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders *Computers in Biology and Medicine* **43**(5) 576-86.

[12] Meng Y, Zou J, Gan X, Zhao L 2012 Research on WNN aerodynamic modeling from flight data based on improved PSO algorithm *Neurocomputing* **83** 212-21



**Author**

**Shunye Wang, born in June, 1981, Hebei, China**

**Current position, grades:** lecturer Department of Mathematics and Computational Science, Langfang Teachers University.
**University studies:** M.S degree in Computer Science and Technology, Computer Applications from the China University of Petroleum, China, 2003, 2006.
**Scientific interest:** machine learning and data mining.