A New Hybrid Particle Swarm Optimization with Wavelet Theory Based Mutation Operation

S.H. Ling, C.W. Yeung, K.Y. Chan, H.H.C. Iu, and F.H.F. Leung

Abstract—**An improved hybrid particle swarm optimization (PSO) that incorporates a wavelet-based mutation operation is proposed. It applies wavelet theory to enhance PSO in exploring solution spaces more effectively for better solutions. A suite of benchmark test functions and an application example on tuning an associative-memory neural network are employed to evaluate the performance of the proposed method. It is shown empirically that the proposed method outperforms significantly the existing methods in terms of convergence speed, solution quality and solution stability.**

I. INTRODUCTION

Particle swarm optimization (PSO) is a recently proposed population based stochastic optimization algorithm, which is inspired by the social behaviours of animals like fish schooling and bird flocking [6]. Comparing with other population based stochastic optimization methods, such as the evolutionary algorithms, PSO has comparable or even superior search performance for many hard optimization problems with faster and more stable convergence rates [7]. However, observations reveal that PSO converges sharply in the earlier stages of the searching process, but it saturates or even terminates in the later stages. It behaves like the traditional local searching methods that trap in local optima. It is hard to obtain any significant improvements by examining neighbouring solutions in the later stages of the search. Vaessens *et al.* [11] and Reeves [14] put this searching method into the context of local search or neighbourhood search.

Ahmed *et al.* [1] proposed a hybrid PSO with the integration of GA's mutation with a constant mutating space. In this approach, particles can follow different directions by themselves, and local element of particles can be permutated. The solution space can still be explored in the later stage of the search by the mutation operation, and pre-mature convergence is more likely to be avoided. However, in that approach, the mutating space is kept unchanged all the time throughout the search; so, the space of permutation of particles in PSO is also unchanged. It can be further improved by varying the mutating space along the search.

Manuscript received March 15, 2007.

S.H. Ling is with School of Electrical, Electronic and Computer Engg, University of Western Australia, WA6009, Australia (phone: 61-8-64881592; fax: 61-8-64881065; e-mail: steve@ee.uwa.edu.au).

C.W. Yeung is with the Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Ham, Hong Kong (e-mail: encwy@ eie.polyu.edu.hk).

K.Y. Chan is with Dept. of Industrial and System Engineering, The Hong Kong Polytechnic University, Hung Ham, Hong Kong (e-mail: chankityan18112005@ yahoo.com).

H.H.C. Iu is with School of Electrical, Electronic and Computer Engg, University of Western Australia, WA6009, Australia (e-mail: herbert@ee.uwa.edu.au).

F.H.F. Leung is with the Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Ham, Hong Kong. (e-mail: enfrank@inet.polyu.edu.hk).

On doing GA's mutation operation, the solution space is more likely to be explored in the early stage of the search by setting a larger mutating space, and it is more likely to be fine tuned to a better solution in the later stage of the search by setting a smaller mutating space, based on the properties of wavelet [2]. This technique can also be applied to improve the hybrid PSO with GA's mutation. A mutation with a dynamic mutating space by incorporating the wavelet function [2] is proposed. The wavelet is a tool to model seismic signals by combining dilations and translations of a simple, oscillatory function (mother wavelet) of a finite duration. The PSO's mutating space is varying dynamically based on the properties of the wavelet function. The resulting mutation operation aids the hybrid PSO to perform more efficiently and provide a faster convergence than the standard or other hybrid PSOs [9][1] in solving a suite of 8 benchmark test functions. In addition, it achieves better and more stable solution quality. An application example on tuning an associative-memory neural network is given to show the improved performance of the proposed hybrid PSO.

This paper is organized as follows. Section II presents the operation of the hybrid PSO with wavelet mutation. Experimental studies and analysis are given in Section III. Eight benchmark test functions are used to evaluate the performance of the proposed method. The application example of tuning an associative-memory neural network is given in Section IV. A conclusion will be drawn in Section V.

II. HYBRID PSO WITH WAVELET MUTATION

PSO is a novel optimization method developed by *Eberhart et al.* [6-7]. It models the processes of the sociological behaviour of bird flocking, and is one of the important evolutionary computation techniques. Within a number of particles that constitute a swarm, each particle traverses the search space looking for the global optimum. The standard PSO (SPSO) process is shown in Fig.1. In this paper, a hybrid PSO with wavelet mutation (WPSO) is proposed and shown in Fig. 2. The details of both SPSO and WPSO will be discussed as follows.

A. Standard particle swarm optimization (SPSO)

1-4244-1340-0/07$25.00 ©2007 IEEE

In Fig.1, $X(t)$ denotes a swarm at the t-th iteration. Each particle $\mathbf{x}^p(t) \in X(t)$ contains κ elements $x_j^p(t) \in \mathbf{x}^p(t)$ at the t-th iteration, where $p = 1, 2, \ldots, \gamma$ and $j = 1, 2, \ldots, \kappa$. γ denotes the number of particles in the swarm. First, particles of the swarm are initialized and then evaluated by a defined fitness values. The objective of SPSO is to minimize the fitness value (cost value) of a particle through iterative steps. The swarm evolves from iteration t to $t+1$ by repeating the procedures as given in Fig. 1. The SPSO operations are discussed as follows:

The velocity $v_j^p(t)$ (corresponding to the flight speed in a search space) and the coordinate $x_j^p(t)$ of the j-th element of the p-th particle at the t-th generation can be calculated using the following formulas [12]:

$$v_j^p(t) = k \cdot \left(w \cdot v_j^p(t-1) + \varphi_1 \cdot rand() \cdot \left(pbest_j - x_j^p(t-1)\right) + \varphi_2 \cdot rand() \cdot \left(gbest_j - x_j^p(t-1)\right)\right) \quad (1)$$

$$x_j^p(t) = x_j^p(t-1) + v_j^p(t) \quad (2)$$

where

$pbest = [pbest_1 \quad pbest_2 \quad ,\ldots \quad pbest_\kappa]$

$gbest = [gbest_1 \quad gbest_2 \quad ,\ldots \quad gbest_\kappa]$

$j = 1, 2, \ldots, \kappa$.

The best previous position of a particle is recorded and represented as *pbest*; the position of the best particle among all the particles is represented as *gbest*; w is an inertia weight factor; φ_1 and φ_2 are acceleration constants; *rand*() returns a uniformly random number in the range of [0,1]; k is a constriction factor derived from the stability analysis of equation (2) to ensure the system converges but not prematurely [5]. Mathematically, k is a function of φ_1 and φ_2 as reflected in the following equation:

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad (3)$$

where $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$.

SPSO utilizes *pbest* and *gbest* to modify the current search point to avoid the particles moving in the same direction, but to converge gradually toward *pbest* and *gbest*. A suitable selection of the inertia weight w provides a balance between the global and local explorations. Generally, w can be dynamically set with the following equation [7]:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \times t \quad (4)$$

where t is the current iteration number, T is the total number of iteration, $w_{\max}$ and $w_{\min}$ are the upper and lower limits of the inertia weight, and are set to 1.2 and 0.1 respectively in this paper.

In (1), the particle velocity is limited by a maximum value $v_{\max}$. The parameter $v_{\max}$ determines the resolution of regions between the present position and the target position to be searched. This limit enhances the local exploration of the problem space, and it realistically depicts the incremental changes of human learning. If $v_{\max}$ is too high, particles might fly past good solutions. If $v_{\max}$ is too small, particles may not explore sufficiently beyond local solutions. From many experiments with PSO, $v_{\max}$ was often set at 10% – 20% of the dynamic range of the variables on each dimension.

```
begin
    t→0              // iteration number
    Initialize X(t)    // X(t): Swarm for iteration t
    Evaluate f(X(t)) // f(·): fitness function
while (not termination condition) do
    begin
        t→t+1
        // Process of SPSO //
         Update velocity v(t) and position of each
         particle x(t) based on (1) and (2) respectively
            if v(t)>vmax
            v(t)= vmax
            end
            if v(t)<–vmax
            v(t)= – vmax
            end
        // End of the process of SPSO //
        Reproduce a new X(t)
        Evaluate f(X(t))
    end
end
```

Fig. 1. Pseudo code for SPSO.

B. Hybrid Particle swarm optimization with Wavelet Mutation Operation (WPSO)

From our observation, SPSO [9] works well in the early iteration stage, but it usually presents problems on reaching the near-optimal solution. The behaviour of the SPSO is affected by some important aspects related to the velocity update. If a particle's current position coincides with the global best position, the particle will only move away from this point if its inertia weight and velocity are different from zero. If their velocities are very close to zero, all the particles will stop moving once they catch up with the global best particle, which may lead to a premature convergence and no further improvement can be obtained. This phenomenon is known as *stagnation* [4].

Ahmed *et al.* [1] proposed to integrate GAs' mutation operation into PSO, which aids to overcome *stagnation*. Here, we called this hybrid PSO as APSO. The mutation operation starts with a randomly chosen particle in the swarm and moves to different positions inside the search area. The following mutation operation is used in APSO:

$$mut(x_j) = x_j - \omega \tag{5}$$

where x_j is a randomly chosen particle element from the swarm, and ω is a number randomly generated within the range $\left[0,\quad 0.1\times\left(para_{\max}^{j} - para_{\min}^{j}\right)\right]$ representing 10% of the length of the search space. $para_{\max}^{j}$ and $para_{\min}^{j}$ are the upper and lower boundaries of each particle element. The pseudo code for the hybrid PSO with the mutation operation is shown in Fig. 2, in which the mutation on particles will perform after updating the velocities and positions of the particles. It can also be seen from Fig. 1 and Fig. 2 that the two PSO methods are identical except the mutation operation has been integrated in the second method.

However, it can be noticed from (5) that the mutation space in APSO is limited by ω so that only 10% of the range of the parameter x_j (i.e. $0.1\times\left(para_{\max}^{j} - para_{\min}^{j}\right)$) can be introduced as mutation at most. It may not be a good approach to fix the mutation space throughout the whole time of searching. It can be improved by employing a dynamic mutation operation in which the size of the mutation space varies during the search. We propose a wavelet mutation that varies the mutation space's size based on the wavelet theory. This results in a wavelet mutation based hybrid PSO (WPSO). The WPSO is identical to APSO except the wavelet mutation operation is used, which is discussed in the following sub-section.

C. Wavelet Mutation

1. Wavelet theory

Certain seismic signals can be modelled by combining translations and dilations of an oscillatory function with a finite duration called a "wavelet". A continuous-time function $\psi(x)$ is called a "mother wavelet" or "wavelet" if it satisfies the following properties:

Property 1:

$$\int_{-\infty}^{+\infty} \psi(x)dx = 0 \tag{6}$$

In other words, the total positive momentum of $\psi(x)$ is equal to the total negative momentum of $\psi(x)$.

Property 2:

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 dx < \infty \tag{7}$$

where most of the energy in $\psi(x)$ is confined to a finite duration and bounded. The Morlet wavelet (as shown in Fig. 3) [2] is an example mother wavelet:

$$\psi(x) = e^{-x^2/2}\cos(5x) \tag{8}$$

```
begin
    t→0              // iteration number
    Initialize X(t)  // X(t): Swarm for iteration t
    Evaluate f(X(t)) // f(·): fitness function
while (not termination condition) do
    begin
        t→t+1
        Perform the process of PSO (shown in Fig. 1)
        Perform mutation operation
        Reproduce a new X(t)
        Evaluate f(X(t))
    end
end
```

Fig. 2 Pseudo code for hybrid PSO with mutation operation

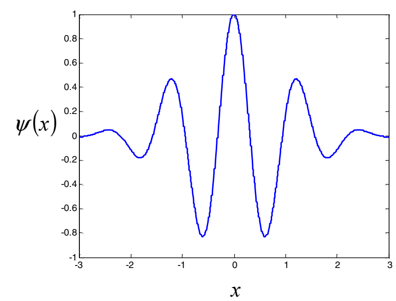

Fig. 3 Morlet wavelet.

The Morlet wavelet integrates to zero (*Property 1*). Over 99% of the total energy of the function is contained in the interval of $-2.5 \le x \le 2.5$ (*Property 2*).

In order to control the magnitude and the position of $\psi(x)$, a function $\psi_{a,b}(x)$ is defined as follows.

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{9}$$

where a is the dilation parameter and b is the translation parameter. Notice that

$$\psi_{1,0}(x) = \psi(x). \tag{10}$$

As

$$\psi_{a,0}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x}{a}\right), \tag{11}$$

it follows that $\psi_{a,0}(x)$ is an amplitude-scaled version of $\psi(x)$. Fig. 4 shows different dilations of the Morlet wavelet. The amplitude of $\psi_{a,0}(x)$ will be scaled down as the dilation parameter a increases. This property is used to do the mutation operation in order to enhance the searching performance.

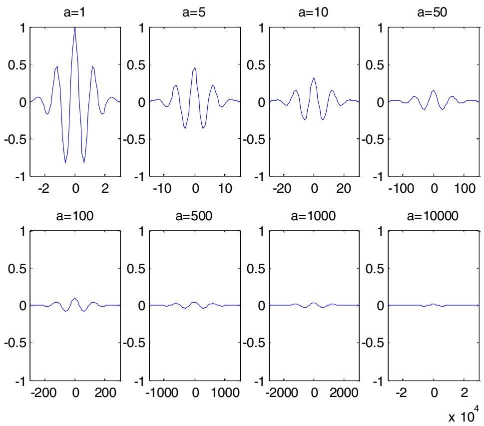

Fig. 4 Morlet wavelet dilated by different values of the parameter a (x-axis: x, y-axis: $\psi_{a,0}(x)$.)

2. Operation of wavelet mutation

The mutation operation is used to mutate the elements of particles. In general, various methods like uniform mutation or non-uniform mutation [8, 10] can be employed to realize the mutation operation. The proposed wavelet mutation (WM) operation exhibits a fine-tuning ability. The details of the operation are as follows. Every particle of the swarm will have a chance to mutate governed by a probability of mutation, $\mu_m \in [0 \quad 1]$, which is defined by the user. For each particle, a random number between 0 and 1 will be generated such that if it is less than or equal to μ_m, the mutation will take place on that element of particle. For instance, if $\mathbf{x}^p(t) = \left[x_1^p(t), \quad x_2^p(t), \quad \ldots \quad, x_\kappa^p(t)\right]$ is the selected p-th particle and the element of particle $x_j^p(t)$ is randomly selected for mutation (the value of $x_j^p(t)$ is inside [$para_{\min}^j, para_{\max}^j$]), the resulting particle is given by $\bar{\mathbf{x}}^p(t) = \left[\bar{x}_1^p(t+1), \quad \bar{x}_2^p(t+1), \quad \ldots \quad, \bar{x}_\kappa^p(t+1)\right]$, where $j \in 1, 2, \ldots \kappa$, κ denotes the dimension of a particle and

$$\bar{x}_j^p(t) = \begin{cases} x_j^p(t) + \sigma \times \left(para_{\max}^j - x_j^p(t)\right) & \text{if } \sigma > 0 \\ x_j^p(t) + \sigma \times \left(x_j^p(t) - para_{\min}^j\right) & \text{if } \sigma \le 0 \end{cases}, \tag{12}$$

$$\sigma = \psi_{a,0}(\varphi) \tag{13}$$

$$\sigma = \frac{1}{\sqrt{a}} \psi\left(\frac{\varphi}{a}\right) \tag{14}$$

By using the Morlet wavelet in (8) as the mother wavelet,

$$\sigma = \frac{1}{\sqrt{a}} e^{-\left(\frac{\varphi}{a}\right)^2/2} \cos\left(5\left(\frac{\varphi}{a}\right)\right) \tag{15}$$

If σ is positive ($\sigma > 0$) approaching 1, the mutated element of the particle will tend to the maximum value of $x_j^p(t)$. Conversely, when σ is negative ($\sigma \le 0$) approaching −1, the mutated element of the particle will tend to the minimum value of $x_j^p(t)$. A larger value of $|\sigma|$ gives a larger searching space for $x_j^p(t)$. When $|\sigma|$ is small, it gives a smaller searching space for fine-tuning. Referring to *Property 1* of the wavelet, the total positive momentum of the mother wavelet is equal to the total negative momentum of the mother wavelet. Then, the sum of the positive σ is equal to the sum of the negative σ when the number of samples is large and φ is randomly generated. That is,

$$\frac{1}{N} \sum_N \sigma = 0 \text{ for } N \to \infty, \tag{16}$$

where N is the number of samples.

Hence, the overall positive mutation and the overall negative mutation throughout the evolution are nearly the same. This property gives better solution stability (smaller standard deviation of the solution values upon many trials). As over 99% of the total energy of the mother wavelet function is contained in the interval [−2.5, 2.5], φ can be generated from $[-2.5, 2.5] \times a$ randomly. The value of the dilation parameter a is set to vary with the value of t/T in order to meet the fine-tuning purpose, where T is the total number of iteration and t is the current number of iteration. In order to perform a local search when t is large, the value of a should increase as t/T increases so as to reduce the significance of the mutation. Hence, a monotonic increasing function governing a and t/T is proposed as follows.

$$a = e^{-\ln(g) \times \left(1 - \frac{t}{T}\right)^{\zeta_{wm}} + \ln(g)} \tag{17}$$

where ζ_{wm} is the shape parameter of the monotonic increasing function, g is the upper limit of the parameter a. The effects of the various values of the shape parameter ζ_{wm} to a with respect to τ/T are shown in Fig. 5. In this figure, g is set as 10000. Thus, the value of a is between 1 and 10000. Referring to (15), the maximum value of σ is 1 when the random number of φ =0 and $a = 1$ (t/T = 0). Then, referring to (12), the offspring particle element $\bar{x}_j^p(t) = x_j^p(t) + 1 \times \left(para_{\max}^j - x_j^p(t)\right) = para_{\max}^j$. It ensures that a large search space for the mutated particle element is given. When the value t/T is near to 1, the value of a is so large that the maximum value of σ will become very small. For example, at t/T =0.9 and $\zeta_{wm} = 1$, the dilation parameter a = 400; if the random value of φ is zero, the value of σ will be equal to 0.0158. With $\bar{x}_j^p(t) = x_j^p(t) + 0.0158 \times \left(para_{\max}^j - x_j^p(t)\right)$, a smaller searching space for the mutated particles element is given for fine-tuning.

After the operation of wavelet mutation, a new swarm is generated. This new swarm will repeat the same process. Such an iterative process will be terminated when a defined number of iteration is met.

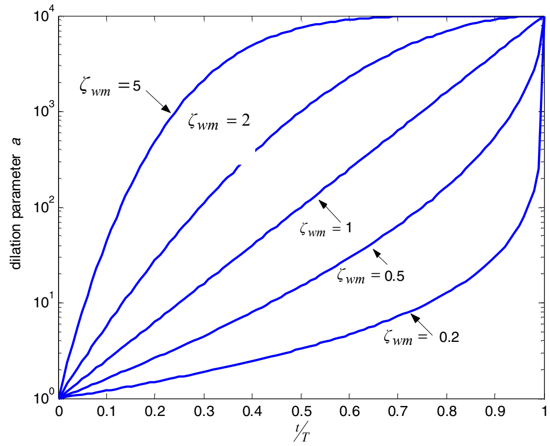

Fig. 5 Effect of the shape parameter ζ_{wm} to a with respect to t/T.

III. BENCHMARK TEST FUNCTIONS AND RESULTS

A. Benchmark test functions

A suit of eight benchmark test functions [13] are used to test the performance of the WPSO. Many different kinds of optimization problems are covered by these benchmark test functions. The benchmark test functions can be divided into three categories: the first is the category of unimodal functions, which are functions with a single minimum; f_1 to f_3 are unimodal functions. The second is the category of multimodal functions with a few local minima; f_4 and f_5 belong to this category. The last is the category of multimodal function with many local minima; f_6 to f_8 belong to this category. The details of these functions are given in Table I.

B. Experimental Setup

The performance of SPSO [9], APSO [1] and the proposed WPSO on solving the benchmark functions is evaluated.

The following simulation conditions are used:

- The shape parameter of the wavelet mutation (ζ_{wm}): It is chosen by trial and error through experiments for good performance for all functions. (ζ_{wm}=2 for f_1-f_2 and f_5-f_8, ζ_{wm}=0.2 for f_3, and ζ_{wm}=1.5 for f_4)
- The acceleration constant φ_1 : 2.05
- The acceleration constant φ_2 : 2.05
- Maximum velocity $v_{\max}$: 0.2
- Swarm size: 30
- Number of runs: 50
- Probability of mutation (p_m): It is chosen by trial and error through experiments for good performance for all functions. (p_m = 0.2 for f_1 and f_8, p_m =0.5 for f_2-f_7)
- Initial population: it is generated uniformly at random

C. Results and Analysis

In this section, the simulation results for the 8 benchmark test functions are given to show the performance of the WPSO. The experimental result in terms of the mean cost value, best cost value, standard deviation and convergence rate are summarized in Table III and Fig. 6.

1. Unimodal function

Function f_1 is a sphere model, which is a smooth and symmetric. The main use of this function is to measure the convergence rate of a searching. It is probably the most widely used test function for searching algorithms. For this function, the results in terms of the mean cost value and the best cost value of WPSO are much better than those of the other two methods. Comparing WPSO to APSO, the mean cost value is almost 150 times better. Still, the standard deviation is much smaller, which means the search solutions are more stable. From Fig. 6(a), the convergence rate of WPSO is seen to be faster than that of APSO and SPSO.

For function f_2, which is a quadratic function padded with noise, the difficulty for searching the minimum value is high since the function would not return the same value at the same point. From Fig. 6(b), it is clearly shown that the WPSO and the SPSO have a better convergence rate. Although the results of these 2 PSO methods do not show a significant difference, the WPSO can give the best result as compared with others.

Function f_3 is associated with the Schwefel's problem 2.21. From the table, although the best cost value of the WPSO is a little bit worse than that of the APSO, the mean cost value and the standard derivation of the WPSO are the best. Thus, the WPSO gives a better solution quality and stability.

2. Multimodal function with a few local minima

For function f_4, which is a multimodal function with only a few local minima, different results from the proposed methods are obtained. According to Fig. 6, although the convergence rate of the WPSO is slower than that of the APSO and the SPSO, the WPSO can provide better results in terms of cost value and standard deviation.

Function f_5 renders no significant difference among the three PSO methods. They all reach or get near to the global optimum, but the WPSO still provides the smallest standard deviation.

3. Multimodal function with many local minima

Functions f_6 to f_8 are multimodal function with many local minima. For function f_6, it can be seen clearly from Fig. 6 that if the mutation operation is not adopted, the searching is easily trapped at some local minimum. From the result obtained, the mean cost value, the best cost value and the standard deviation of the WPSO are better than those of the

APSO; that mean the WPSO can provide more stable and higher quality results.

Function f_7 exhibits similar results as those of function f_6. From Fig. 6, it can be seen that the trend of the curve of the three methods are quite similar; however after 600 times of iteration, the WPSO shows a good searching ability.

For function f_8, it is clearly shown that the searching ability of the three algorithms is quite different. From Fig. 6, it can be seen that they behave similarly at the first 400 times of iteration, but are trapped in some minima afterwards. Still, the WPSO gives the best result. The mean cost value of the WPSO is almost 30000 times better than that of the APSO, and the standard deviation offered by the WPSO is much smaller than the others.

4. *T-test*

The t-test is a statistical method to evaluate the significant difference between two algorithms. The t-value will be positive if the first algorithm is better than the second, and negative if it is poorer. The t-value is defined as follows:

$$t = \frac{\bar{\alpha}_2 - \bar{\alpha}_1}{\sqrt{\left(\frac{\sigma_2^{\,2}}{\xi+1}\right)+\left(\frac{\sigma_1^{\,2}}{\xi+1}\right)}} \tag{18}$$

where $\bar{\alpha}_1$ and $\bar{\alpha}_2$ are the mean value of the first method and the second method respectively; σ_1 and σ_2 are the standard deviations of the first method and the second method respectively; ξ is the degree of freedom. When the t-value is higher than 2.15 (ξ = 49), there is a significant difference between the two algorithms with a 98% confident level. The relevant t-values are shown in Table II, which shows that nearly all t-values are higher than 2.15. Therefore, the performance of WPSO is significantly better than the other PSO methods with a 98% confident level.

IV. PSO WITH WM FOR NEURAL NETWORK TRAINING

An application example on tuning a neural network (linear auto-associative memory) [15] is given in this section. The linear auto-associative memory, which maps its input vector into itself, has ten inputs and ten outputs. Thus, the desired output vector is its input vector. 50 input vectors are used for the learning. The linear auto-associative memory is given by:

$$y_k(t) = \sum_{j=1}^{10} w_{jk} z_j(t),\; k = 1, 2, \ldots, 10 \tag{19}$$

where $z(t)$=[$z_1(t)$ ... $z_{10}(t)$] is the input vector and w_{jk} is the weight of the link between the input and the output. The objective is to minimize the mean square error (MSE), which is defined as follows:

$$\text{MSE} = \frac{\sum_{k=1}^{10}\sum_{t=1}^{50}\left(z_k(t) - y_k(t)\right)^2}{10\times 50} \tag{20}$$

The basic setting of the PSO parameters is the same as that in Section III. The initial range of the weight w_{jk} is between –2 and 2. The number of iteration is set at 1000. Probability of mutation (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are set at 0.5 and 2 respectively. The experimental results are tabulated in Table IV, and the comparison between different PSOs is shown in Fig. 7. The table shows that the mean cost value, best cost value and standard deviation offered by the WPSO are the smallest. Also the t-values of WPSO to APSO, and of WPSO to SPSO are 3.64 and 11.79 respectively, which are higher than 2.15. Thus WPSO is significantly better than both APSO and WPSO with a 98% confident level. From these results, the proposed WPSO is good for training associative memory.

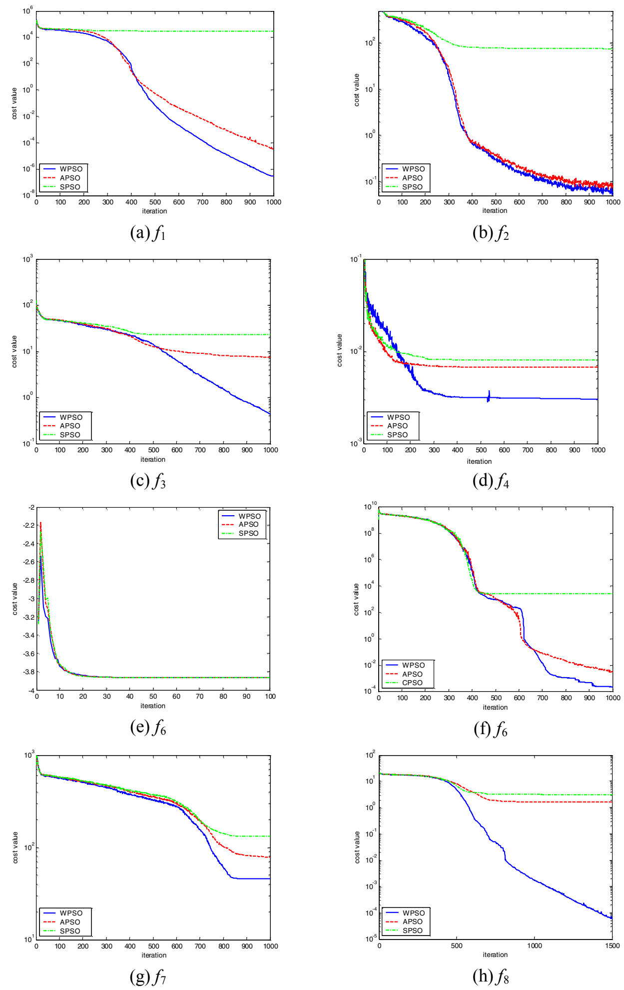

Fig. 6 Comparisons between different PSO methods for f_1 to f_8.

TABLE I. BENCHMARK TEST FUNCTIONS

Test function	Domain range	Optimal point
$f_1(\mathbf{x})=\sum_{i=1}^{30} x_i^2$	$-50 \le x_i \le 150$	$f_1(\mathbf{0})=0$
$f_2(\mathbf{x})=\sum_{i=1}^{30} ix_i^4 + random[0,1)$	$-1.28 \le x_i \le 2.56$	$f_2(\mathbf{0})=0$
$f_3(\mathbf{x})=\max_i\{\lvert x_i \rvert, 1 \le i \le 30\}$	$-150 \le x_i \le 50$	$f_3(\mathbf{0})=0$
$f_4(\mathbf{x})=\sum_{i=1}^{9}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	$-5 \le x_i \le 5$	$f_4(0.1928, 0.1928, 0.1231, 0.1358) \approx 0.0003075$
$f_5(\mathbf{x})=-\sum_{i=1}^{30} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2\right]$	$0 \le x_i \le 1$	$f_5(0.114, 0.556, 0.852) \approx -3.8628$
$f_6(\mathbf{x})=0.1\left\{\begin{array}{l}\sin^2(\pi 3x_1) \\ +\sum_{i=1}^{29}(x_i-1)^2\cdot\left[1+\sin^2(3\pi x_{i+1})\right] \\ +(x_{30}-1)^2\left[1+\sin^2(2\pi x_{30})\right]\end{array}\right\} + \sum_{i=1}^{30} u(x_i,5,100,4)$	$-50 \le x_i \le 50$	$f_6(\mathbf{1})=0$
$f_7(\mathbf{x})=\sum_{i=1}^{30}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	$-5.12 \le x_i \le 10.24$	$f_7(\mathbf{0})=0$
$f_8(\mathbf{x})=-20\exp\left(-0.2\sqrt{\frac{1}{3}\sum_{i=1}^{30}x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right)+20+e$	$-64 \le x_i \le 32$	$f_8(\mathbf{0})=0$

TABLE II. T-VALUE OF WPSO TO THE OTHER PSO METHODS

Functions	*t*-value of WPSO to APSO	*t*-value of WPSO to SPSO
f_1	10.76	29.11
f_2	4.12	8.78
f_3	2.81	7.01
f_4	2.61	3.45
f_5	N/A	3.30
f_6	4.37	5.37
f_7	5.11	10.10
f_8	2.19	4.79

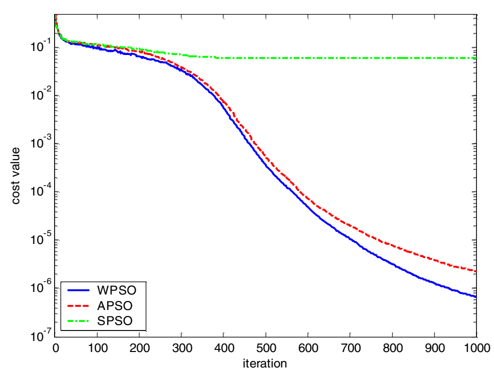

Fig. 7. Comparisons between different PSO methods for tuning a linear auto-associative memory (neural network).

V. CONCLUSION

In this paper, we proposed a new hybrid PSO with wavelet mutation. The wavelet theory is applied to the mutation operation so that the PSO can explore the solution space more effectively on reaching the search solution. Simulation results have shown that the proposed wavelet mutation based hybrid PSO is a useful optimization method. Thanks to the properties of the wavelet, the convergence to global optimum and the stability of the solution quality are improved. On testing a suite of benchmark functions and tuning the parameters of a neural network, the WPSO exhibits better results than the APSO and SPSO. Also, a faster convergence speed can be achieved by the WPSO in general.

TABLE III: COMPARISON BETWEEN DIFFERENT PSO METHODS FOR DIFFERENT FUNCTIONS. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS

f_1 ($\times 10^{-7}$), number of iteration: 1000			
	WPSO	APSO	SPSO
Mean	**2.8608**	328.173	30152x10^7
Best	**0.81486**	57.7915	15000x10^7
Std Dev	**1.6948**	213.7089	7324x10^7
f_2 ($\times 10^{-2}$), number of iteration: 1000			
	WPSO	APSO	SPSO
Mean	**1.1275**	1.4616	7580.1
Best	**0.5892**	0.5951	1.0182
Std Dev	**0.3873**	0.4222	6102.2
f_3 ($\times 10^{-1}$), number of iteration: 1000			
	WPSO	APSO	SPSO
Mean	**4.3048**	73.4740	227.638
Best	2.6381	**1.9197**	24.1125
Std Dev	**0.9796**	174.03476	225.384
f_4 ($\times 10^{-3}$), number of iteration: 1000			
	WPSO	APSO	SPSO
Mean	**2.7059**	6.7076	7.9826
Best	**0.5056**	0.6854	0.4682
Std Dev	**5.9618**	9.0324	9.0317
f_5 ($\times 10^{0}$), number of iteration: 100			
	WPSO	APSO	SPSO
Mean	**-3.8628**	-3.8628	-3.8614
Best	**-3.8628**	-3.8628	-3.8628
Std Dev	**4.0867x10^{-10}**	2.9870x10^{-9}	3.0140x10^{-3}
f_6 ($\times 10^{-4}$), number of iteration: 1000			
	WPSO	APSO	SPSO
Mean	**2.2945**	23.3635	26666529
Best	**0.03287**	5.5488	8376.2988
Std Dev	**15.539**	30.3876	35138848.8
f_7 ($\times 10^{1}$), number of iteration :1000			
	WPSO	APSO	SPSO
Mean	**4.5550**	7.7107	13.2319
Best	**1.3990**	1.4266	3.5107
Std Dev	**2.1928**	3.7735	5.6646
f_8 ($\times 10^{-4}$), number of iteration :1500			
	WPSO	APSO	SPSO
Mean	**0.5825**	16006	31260
Best	**0.2195**	4.9647	336.21
Std Dev	**0.1509**	51803	46185

TABLE IV: COMPARISON BETWEEN DIFFERENT PSO METHODS FOR TUNING

A LINEAR AUTO-ASSOCIATIVE MEMORY (NEURAL NETWORK). ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS

$(x10^{-6})$, number of iteration :1000			
	WPSO	APSO	SPSO
Mean	**0.6725**	2.2726	61246.0
Best	**0.0320**	0.0534	8308.11
Std Dev	**1.1459**	2.8936	36721.0

ACKNOWLEGEMENT

The work described in this paper was partially supported by The University of Western Australia, Australia and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5218/06E)

REFERENCES

[1] A.A.E. Ahmed, L.T. Germano, and Z.C. Antonio, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 859-866, May 2005.

[2] I. Daubechies, *Ten lectures on Wavelets*. Philadelphia. PA: Society for Industrial and Applied Mathematics, 1992.

[3] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th International Symposium on Micro Machine and Human Science*, IEEE Service Center, Nagoya, Oct. 1995, pp.39-43.

[4] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Evolutionary Programming VII*. New York: Springer-Verlag, 1998, vol. 1447, Lecture Notes in Computer Science, pp. 611–616.

[5] R.C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congress on Evolutionary Computing*, vol. 1, Jul. 2000, pp.84-88.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, vol. 4, 1995, pp.1942-1948.

[7] J. Kennedy and R. Eberhart, *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.

[8] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, 2nd extended ed. Springer-Verlag, 1994.

[9] N. Mo., Z.Y. Zou, K.W. Chan, and T.Y.G. Pong, "Transient stability constrained optimal power flow using particle swarm optimization," *IEE Proceedings – Generation, Transmission and Distribution*. (Accepted to be published).

[10] A. Neubauer, "A theoretical analysis of the non-uniform mutation operator for the modified genetic algorithm," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1997, Indianapolis, pp. 93-96.

[11] R.J.M. Vaessens, E.H.L. Aarts, and J.K. Lenstra, "A local search template," *Proceedings of Parallel Problem-Solving from Nature 2*, pp. 65-74, 1992.

[12] B. Zhao, C.X. Guo, and Y.J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Trans. Power Systems.*, vol. 20, no. 2, pp.1070-1078, May 2005.

[13] X. Yao and Y. Liu, "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, July 1999.

[14] C.R. Reeves, "Genetic algorithms and neighbourhood search," *Evolutionary Computing: AISB Workshop*, pp. 115-130, 1994.

[15] J.M. Zurada, *Introduction to Artificial Neural Systems*. West Info Access, 1992.