Proceedings of 2009 APSIPA Summit and Conference, Sapporo, Japan, October 4-7, 2009

A Particle Swarm Optimization with Multi-Mutation Operation Based on Wavelet Theory

C.W. Yeung†, J.C.Y. Lai* and F.H.F. Leung✦
†Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Ham, Hong Kong. E-mail: 07900256r@polyu.edu.hk
*Centre for Signal Processing, Dept. of Electronic and Information Engg., The Hong Kong Polytechnic University, Hung Ham, Hong Kong. E-mail: 08900438r@polyu.edu.hk
✦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

***Abstract*—An improved hybrid particle swarm optimization (PSO) that incorporates a wavelet-based multi-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 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 a faster and more stable convergence rate [7]. However, observations reveal that PSO converges sharply in the early stage of the searching process, but it saturates or even terminates in the later stage. It behaves like the traditional local searching methods that trap in local optima. It is hard to obtain any significant improvement by examining neighbouring solutions in the later stage of the search. Vaessens *et al.* [11] and Reeves [14] put these searching methods into the context of local search or neighbourhood search.

Ahmed *et al.* [1] proposed a hybrid PSO that integrated the Genetic Algorithm (GA) mutation within a constant mutating space. Under this approach, particles can search different directions by themselves, and local positions of particles can be permutated. The solution space can still be explored by the mutation operation in the later stage of the search, and pre-mature convergence is more likely to be avoided. However, under that approach, the mutating space is kept unchanged all the time throughout the search. It can be further improved by varying the mutating space along the search.

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 operation with a dynamic mutating space that incorporates a 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. However, in recent research [16] of PSO with wavelet mutation (WPSO), only one element in each particle may undergo the mutation process in an iteration step. This may pre-maturely restrict the searching space, although the searching space has been varying during the searching process. An improved wavelet mutation is proposed in this paper, which allows more than one element in each particle to be mutated in each searching process. The resulting multi-mutation operation aids the hybrid PSO to perform more efficiently and provide a faster convergence than the PSO with wavelet mutation, the standard PSO, and other hybrid PSOs [1][9] in solving a suite of 8 benchmark test functions.

This paper is organized as follows: Section II presents the operation of the hybrid PSO with multi-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. A conclusion will be drawn in Section IV.

II. HYBRID PSO WITH MULTI-WAVELET MUTATION

PSO is a novel optimization method developed by Eberhart *et al.* [6-7]. It models 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 multi-wavelet mutation (MWPSO) is proposed and shown in Fig. 4. The details of SPSO, WPSO and MWPSO will be discussed as follows.

A. Standard particle swarm optimization (SPSO)

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,... , γ and j = 1, 2,… , κ ; γ denotes the number of particles in the swarm. First, particles of the swarm are initialized and then evaluated by a defined fitness function. The objective of SPSO is to minimize the fitness value (cost value) of a particle through iteration steps. The swarm evolves from iteration t to t +1 by repeating the procedure 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]:

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

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

where $pbest^p = \left[pbest_1^p \quad pbest_2^p \quad ,... \quad pbest_\kappa^p\right]$, $gbest = \left[gbest_1 \quad gbest_2 \quad ,... \quad gbest_\kappa\right], j = 1, 2, \ldots, \kappa$. The best previous position of the p-th particle is recorded and represented as $pbest^p$; the position of 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 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 convergence but not prematurely [5]. Typically, 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 $pbest^p$ 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 as 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, or fitness, 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 the value of $v_{\max}$ is too high, particles might fly past good solutions; if it 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. Recent Hybrid Particle swarm optimization and its limitation

From our observation, SPSO [9] works well in the early iteration stage, but it usually presents problems on reaching a 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 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 call 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 the randomly chosen particle element from the swarm, and ω is a number randomly generated within the range $\left[0,\quad 0.1\times\left(para^j_{\max} - para^j_{\min}\right)\right]$ representing 10% of the length of the search space. $para^j_{\max}$ and $para^j_{\min}$ are the upper and lower bounds of each particle element. The pseudo code of the hybrid PSO with the mutation operation is shown in Fig. 4, 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. 4 that the two PSO methods are identical except the mutation operation has been integrated in the second method. However, (5) indicates that the mutating space in APSO is limited by ω in which 10% of the range of the parameter x is used. It may not be a good approach in fixing the mutating space at all time of the search. It can be further improved by employing a dynamic mutation operation in which the size of the mutating space varies during the search.

C. 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 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:

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

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

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

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 position of $\psi(x)$, a function $\psi_{a,b}(x)$ is defined as follows.

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

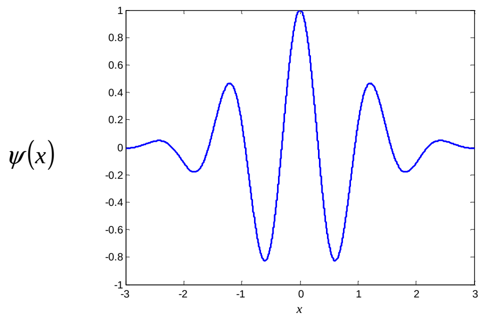


Fig. 2. Morlet wavelet.

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

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

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

It follows that $\psi_{a,0}(x)$ is an amplitude-scaled version of $\psi(x)$. Fig. 3 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.

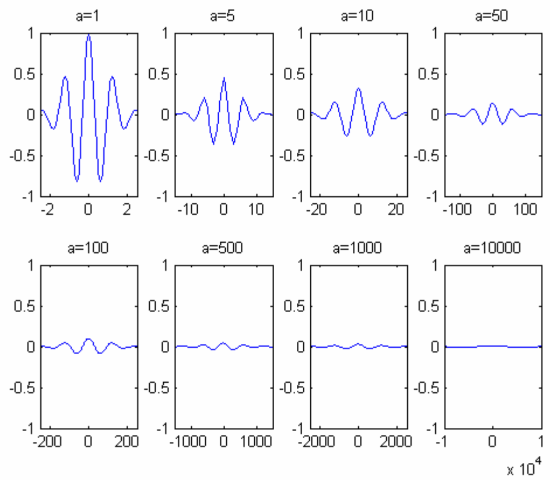


Fig. 3. Morlet wavelet dilated by different values of the parameter a (x-axis: x, y-axis: $\psi_{a,0}(x)$.)

D. WPSO and the Proposed MWPSO

We propose a multi-wavelet mutation that varies the mutating space based on the wavelet theory. The MWPSO is identical to WPSO except the number of elements that undergo the mutation process in each particle can been controlled. Both WPSO and MWPSO will be discussed in the following sub-section.

1. WPSO and its Operation

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 that controls which element in the particle will be mutated, 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 the element's bounds [$para_{\min}^j, para_{\max}^j$]), the resulting particle is given by $\bar{\mathbf{x}}^p(t) = \left[x_1^p(t), \quad x_2^p(t), \quad \ldots, \quad \bar{x}_j^p(t), \quad \ldots \quad , x_\kappa^p(t)\right]$, where $j \in 1, 2, \ldots \ \kappa$; κ denotes the dimension of particle and

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

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

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

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 will tend to the maximum value of $x_j^p(t)$. Conversely, when σ is negative ($\sigma \le 0$) approaching −1, the mutated element 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 energy of the mother wavelet is equal to the total negative energy 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.

```
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 with p_m
                      If  perform multi-mutation
                            Select the elements with N_m
                Reproduce a new X(t)
                Evaluate f(X(t))
          end
```

Fig. 4 Pseudo code for hybrid PSO with mutation operation.

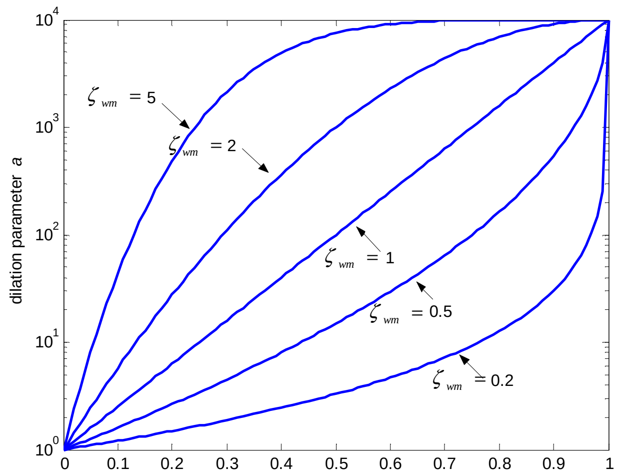


Fig. 5 Effect of the shape parameter ζ_{wm} to a with respect to t/T.

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] 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 gene $\bar{x}_j^p(t) = x_j^p(t)+1\times\left(para_{\max}^j - x_j^p(t)\right) = para_{\max}^j$. It ensures that a large search space for the mutated element of particle 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. A smaller searching space for the mutated element of particles is then 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.

2. The proposed MWPSO

For the proposed MWPSO, one more parameter $N_m \in [0\ \ 1]$ is defined. The value of N_m is randomly set at each iteration step. This parameter control the number of elements in the particle that mutate, such that more than one element in each particle can vary its value and more freedom will be given to the particle to explore the searching space. For instance, if $\mathbf{x}^p(t)=\left[x_1^p(t),\ \ x_2^p(t),\ \ \ldots\ \ , x_\kappa^p(t)\right]$ is the selected p-th particle, the number of elements that undergoes mutation is controlled by:

$$\text{Number of mutated elements} = N_m \times \kappa \tag{18}$$

The elements for doing mutation are randomly selected. The resulting particle is denoted by $\bar{\mathbf{x}}^p(t)=\left[\bar{x}_1^p(t+1),\ \ \bar{x}_2^p(t+1),\ \ \ldots\ \ , \bar{x}_\kappa^p(t+1)\right]$, where $j \in$ 1, 2, … κ

III. Benchmark Test Functions and Results

A. Benchmark test function

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

Table I. Benchmark Test Functions.

Test function	Domain range	Optimal point
Sphere function $f_1(\mathbf{x})=\sum_{i=1}^{30} x_i^2$	$-50\le x_i \le 150$	Min(f_1)= f_1(0)=0
Step function $f_2(\mathbf{x})=\sum_{i=1}^{30}\left(\lfloor x_i+0.5\rfloor\right)^2$	$-5\le x_i \le 10$	Min(f_2)= f_2(0)=0
Schwefel's Problem 2.21 $f_3(\mathbf{x})=\max_i\left\{\lvert x_i\rvert, 1\le i\le 30\right\}$	$-150 \le x_1 \le 50$	Min(f_3)= f_3(0)= −1
Kowalik's function $f_4(\mathbf{x})=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	$-5\le x_i \le 5$	Min(f_4)= f_4([0.1928 0.1908 0.1231 0.1358])= 3.075x10^{-4}
Hartman's Family I $f_5(\mathbf{x})=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{3}a_{ij}\left(x_i-p_{ij}\right)^2\right]$	$0\le x_i \le 1$	Min(f_5)= f_5([0.114 0.556 0.852])= -3.8628
Griewank Function $f_6(x)=\frac{1}{4000}\sum_{i=1}^{30}x_i^2-\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$-1200\le x_i \le 600$	Min(f_6)= f_6(**0**)=0
Generalized Ackley's function $f_{17}(\mathbf{x})=-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30}x_i^2}\right)-\exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right)+20+e$	$-64\le x_i \le 32$	Min(f_7)= f_7(**0**)=0
Schwefel's function $f_8(x)=\sum_{i=1}^{30}\left(x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	$-500\le x_i \le 500$	Min(f_8)= f_8([420.9687, ..., 420.9687])= −12569.5

B. Experimental Setup

The performance of SPSO [9], APSO [1], WPSO and the proposed MWPSO on solving the benchmark test functions is evaluated. The following simulation conditions are used:

- The shape parameter of wavelet mutation (ζ_{wm}): 0.2
- The acceleration constant φ_1 : 2.05
- The acceleration constant φ_2 : 2.05
- Maximum velocity $v_{\max}$: 0.2
- Swarm size: 40
- Number of runs: 50
- Probability of mutation (μ_m): 0.1
- Mutation parameter (N_m): 0.3
- Initial population: generated uniformly at random

C. Results and Analysis

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

1. Unimodal function

Function f_1 is a sphere model. In view of the characteristic of f_1, which is smooth and symmetric, the main purpose is to measure the convergence rate of the searching. It is probably the most widely used test function. For this function, the result in terms of the mean cost value, the best cost value, and the standard deviation of MWPSO and WPSO are much better than those of the other methods. As shown in Fig. 6(a), the convergence rate of MWPSO is higher than that of WPSO, APSO and SPSO.

Function f_2 is a step function, which is a representative of flat surfaces. Flat surfaces are obstacles for optimization algorithms because they do not give any information about the search direction, unless the algorithm has a variable step size. From Fig. 6(b), it is clearly shown that MWPSO has the best convergence rate as compared with SPSO, WPSO and APSO. We see that by increasing the number of elements for mutation, we can enhance the searching space.

Function f_3 is a Schwefel's problem 2.21. According to Fig. 6(c), the performance does not show significant difference at the first 400 iteration steps. From Table II, although the best cost value of the MWPSO is a little bit larger than that of the APSO, the mean cost value and the standard derivation of the MWPSO are the best. Thus, MWPSO can offer better solution quality and stability.

2. Mulitmodal 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. As shown in Fig. 6(d), SPSO, APSO and WPSO are trapped in different local minima, and the convergence rate of the MWPSO is faster than that of others. Moveover, MWPSO can provide the best result in terms of cost value and standard deviation.

From the result obtained from function f_5, we see that there is no significant difference for all the PSO methods; the curves of convergence are quite similar, and they all can reach or get near to the global optimum. However, MWPSO still provides the best standard deviation value.

3. Multimodal function with many local minima

Functions f_6 to f_8 are multimodal functions with many local minima. For function f_6, it can be seen clearly from Fig. 6(f) that MWPSO is the fastest to reach the optimal point. From the result obtained, MWPSO, WPSO and APSO return the same best cost value, but the standard deviation of MWPSO is the best. Hence, MWPSO can provide more stable and high-quality result.

For function f_7, as seen from Fig 6(g), the proposed method has already reached the optimal point after a few iteration steps, while other methods almost use 200 iteration steps to reach the optimal point. It shows that the MWPSO offers a good searching ability thanks to the multi-wavelet mutation in the PSO. Also, the mean and the standard deviation offered by MWPSO are much better than those of others.

For function f_8, it is shown that the searching ability of the proposed method is quite different from the other methods. All the algorithms except MWPSO have similar behaviour at the first 400 iteration steps, and are trapped in some local minima. On the other hand, the cost value offered by MWPSO is decreasing gradually, and it can provide the best result as compared with others.

IV. Conclusion

In this paper, we proposed a new hybrid PSO with multi-wavelet mutation. Our objective is to increase the searching area by increasing the number of elements in a particle that undergo mutation so as to further improve the performance of WPSO. The solution space can be explored more effectively on reaching the optimal solution. Simulation results have shown that the proposed method is a useful technique to solve optimization problems. On solving a suite of benchmark functions, MWPSO offers better results in terms of solution quality and stability than WPSO, APSO and SPSO. Also a faster convergence speed can be achieved by MWPSO.

Acknowledgement

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project No. RGYY).

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 Int. Symp. Micro Machine and Human Science*, Oct. 1995, Nagoya, pp.39-43.

[4] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *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*, Jul. 2000, vol. 1, pp.84-88.

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

[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, 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.

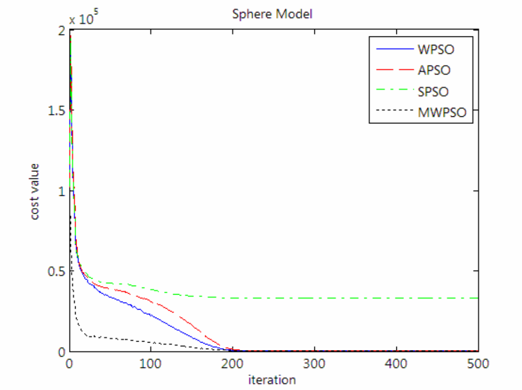
[14] C.R. Reeves, Genetic algorithms and neighbourhood search, *Evolutionary Computing: AISB Workshop*, pp. 115-130, 1994.

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

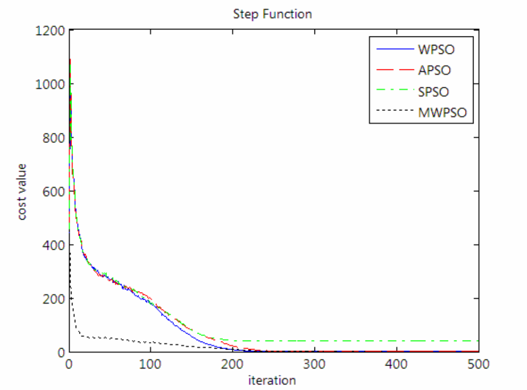
[16] S.H. Ling, C.W. Yeung, K.Y. Chan, H.H.C. Iu, and F.H.F. Leung, "A new hybrid particle swarm optimization with wavelet theory based mutation operation," in *Proc. 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, Singapore, Sep. 25-28, 2007, pp. 1977-1984

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

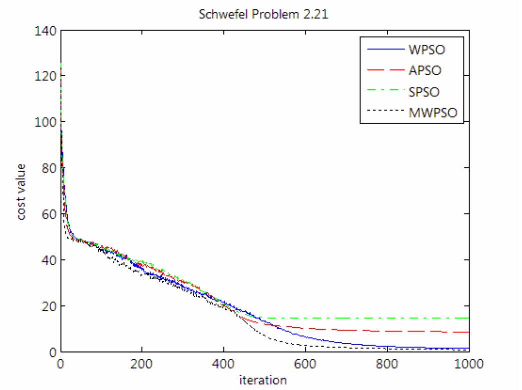
f_1 ($\times 10^0$), number of iteration: 500				
	MWPSO	WPSO	APSO	SPSO
Mean	**0**	**0**	0.0004	32857.9467
Best	**0**	**0**	0.0001	12500.4982
Std Dev	**0**	**0**	0.0003	8626.6124
f_2 ($\times 10^0$), number of iteration: 500				
	MWPSO	WPSO	APSO	SPSO
Mean	**0**	0.62	0.9	39.58
Best	**0**	0	0	0
Std Dev	**0**	1.4553	3.37	34.5154
f_3 ($\times 10^0$), number of iteration: 1000				
	MWPSO	WPSO	APSO	SPSO
Mean	**0.7374**	1.3189	8.4366	14.61
Best	0.2894	0.1854	**0.0743**	1.8902
Std Dev	**0.255**	7.0255	18.421	19.2
f_4 ($\times 10^{-3}$), number of iteration: 1000				
	MWPSO	WPSO	APSO	SPSO
Mean	**1.4**	4.2	6.3	8.5
Best	**0.3**	0.4	0.5	0.3
Std Dev	**3.9**	7.7	8.9	9.3
f_5 ($\times 10^0$), number of iteration: 500				
	MWPSO	WPSO	APSO	SPSO
Mean	**-3.8628**	-3.8628	-3.8628	-3.8625
Best	**-3.8628**	-3.8628	-3.8628	-3.8628
Std Dev	**2.7683e-15**	3.5092e-11	2.7849e-14	0.0016
f_6 ($\times 10^0$), number of iteration: 500				
	MWPSO	WPSO	APSO	SPSO
Mean	**0**	0.1925	**0**	138.1759
Best	**0**	0	**0**	0.0709
Std Dev	**0**	0.2864	**0**	128.0549
f_7 ($\times 10^0$), number of iteration :500				
	MWPSO	WPSO	APSO	SPSO
Mean	**0.0044**	509.6625	0.0179	3278.0132
Best	**0**	0	0	3.8481
Std Dev	**0.0069**	1744.8548	0.112	3581.2285
f_8 ($\times 10^{-4}$), number of iteration :1000				
	MWPSO	WPSO	APSO	SPSO
Mean	**-11210.9656**	-7441.1954	-7180.3602	-6951.7609
Best	**-12352.3469**	-8161.4782	-8159.9612	-8278.3995
Std Dev	578.664	**438.2333**	450.2045	656.8472



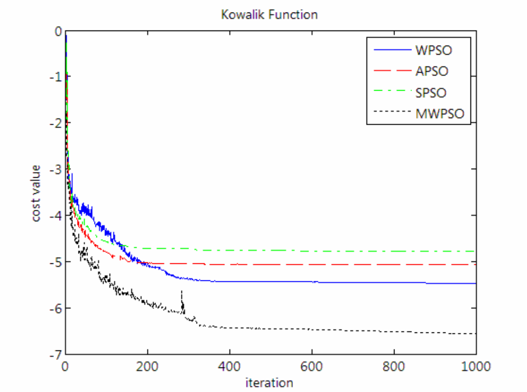
(a) f_1



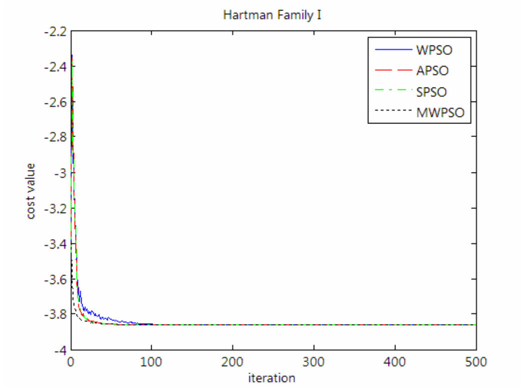
(b) f_2



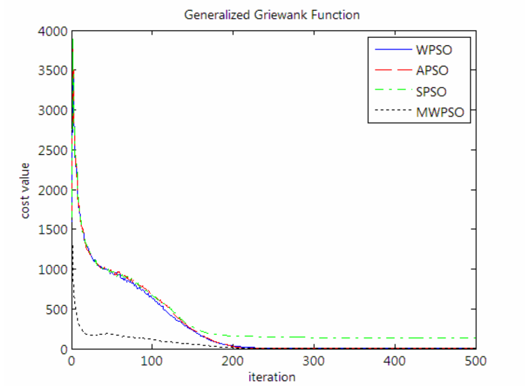
(c) f_3



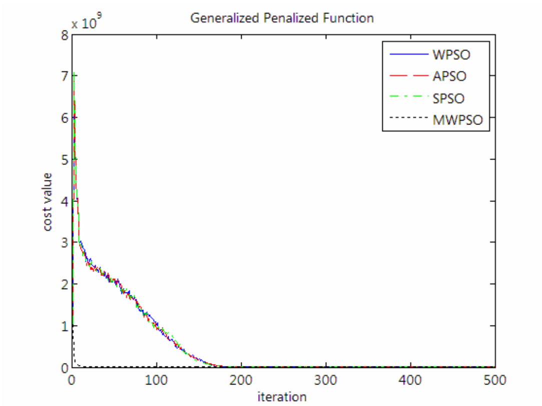
(d) f_4



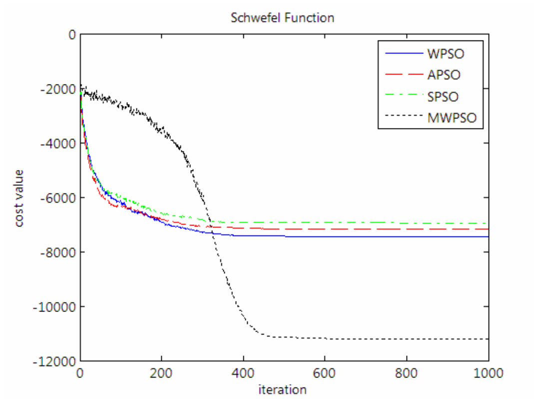
(e) f_5



(f) f_6



(g) f_7



(h) f_8

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