

“© 2008 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Hybrid Particle Swarm Optimization with Wavelet Mutation and its Industrial Applications

¹S.H. Ling *MIEEE*, ²H.H.C. Iu *SrMIEEE*, ³K.Y. Chan, ⁴H.K. Lam *MIEEE*, ⁵C.W. Yeung, and
⁵F.H.F. Leung *SrMIEEE*

¹*Department of Electrical and Computer Engineering, National University of Singapore, Singapore*

²*School of Electrical, Electronic and Computer Engineering, The University of Western Australia, WA, Australia*

³*Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong*

⁴*Department of Electronic Engineering, Division of Engineering, The King's College London, WC2R 2LS, United Kingdom*

⁵*Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong*

Abstract—A new hybrid particle swarm optimization (PSO) that incorporates a wavelet theory based mutation operation is proposed. It applies the wavelet theory to enhance PSO in exploring the solution space more effectively for better solution. A suite of benchmark test functions and three industrial applications (solving the load flow problems, modelling the development of fluid dispensing for electronic packaging, and designing a neural network-based controller) are employed to evaluate the performance and the applicability of the proposed method. Experimental results show 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 evolutionary algorithms, PSO has comparable or even superior search performance for many hard optimization problems with faster and more stable convergence rates [7]. PSO has been used in different industrial areas such as power systems [1, 18-21], parameters learning of neural networks [16, 22], control [23-24], prediction [25], modelling [26-27], etc. However, observations

reveal that PSO converges sharply in the early stages of the searching process, but saturates or even terminates in the later stages. It behaves like the traditional local searching methods that trap in local optima. As a result, 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 these searching methods into the context of local search or neighbourhood search.

Recently, different hybrid PSOs have been proposed to overcome the drawback of trapping in local optima. The hybrid PSO has been first proposed in 1998 [43], in which a standard selection mechanism is integrated with PSO. A new hybrid gradient descent PSO (HGPSO), which is integrated with gradient information to achieve faster convergence without getting trapped in local minima, is proposed by Noel and Jannett [16]. However, the computational demand of HGPSO is increased by the process of the gradient descent. Juang [17] proposed a hybrid PSO algorithm named HGAPSO, which incorporates GA's evolutionary operations of crossover, mutation and reproduction. Ahmed *et al.* [1] proposed a hybrid PSO named HPSOM, in which a constant mutating space is used in mutation. In both HGAPSO and HPSOM, the solution space can be explored by performing mutation operations on particles along the search, and premature convergence is more likely to be avoided. However, the mutating space is kept unchanged all the time throughout the search, and the space for the permutation of particles in PSO is also fixed. It can be further improved by varying the mutating space along the search.

For genetic algorithms (GAs), 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 for better solution in the later stage of the search by setting a smaller mutating space based on the properties of wavelet [2]. This idea can be applied so as to introduce the hybrid PSO with GA's mutation. In this paper, a mutation with a dynamic mutating space by incorporating a wavelet function is proposed. 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 along the search based on the properties of the wavelet function. The resulting mutation operation aids the hybrid PSO to perform more efficiently and provides a faster convergence than the PSO with construction and inertia weight factor [9] and other hybrid PSOs [1, 16-17, 28] in solving a suite benchmark test functions. In addition, it achieves a better and more stable solution quality. Application examples on solving some load flow problems (the

multi-contingency transient stability constrained optimal power flow problem and economic load dispatch with valve-points loading), modelling the development of fluid dispensing for electronic packaging, and designing a neural network-based controller, are employed to demonstrate that better performance can be achieved by 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 discussed in Section III. Eighteen standard benchmark test functions are given to evaluate the performance of the proposed method. Also, five additional benchmark test functions are given in which the global optimal points are shifted and rotated. Furthermore, the sensitivity of the shape parameter and parameter g for the wavelet mutation are discussed in this section. Application examples on the load flow problems, modelling of fluid dispensing for electronic packaging, and the neural network-based controller are 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 Kennedy *et al.* [6]. It models the processes of the sociological behaviour associated with bird flocking, and is one of the evolutionary computation techniques. It considers a number of particles that constitute a swarm. Each particle traverses the search space looking for the global optimum. The standard PSO with constriction and inertia weight factors (SPSO) process is shown in Fig.1. In this paper, a hybrid PSO with wavelet mutation (HPSOWM) is proposed and shown in Fig. 2. The details of both SPSO and HPSOWM will be discussed as follows.

A. Standard particle swarm optimization with constriction and inertia weight factors (SPSO)

In Fig. 1(a), $X(t)$ 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 $p = 1, 2, \dots, \gamma$ and $j = 1, 2, \dots, \kappa$; γ denotes the number of particles in the swarm and κ is the dimension of a particle. First, the particles of the swarm are initialized and then evaluated by a defined fitness function. The objective of PSO is to minimize the fitness values (cost values) of particles iteratively. The swarm evolves from iteration

t to $t + 1$ by repeating the procedure as shown in Fig. 1. The standard PSO [6] operations are discussed as follows.

The velocity $v_j^p(t)$ (corresponding to the flight speed in a search space) and the position $x_j^p(t)$ of the j -th element of the p -th particle at the t -th iteration can be calculated using the following formulae:

$$v_j^p(t) = 2 \cdot rand_j^p(\cdot) \cdot (pbest_j^p - x_j^p(t-1)) + 2 \cdot rand_j^p(\cdot) \cdot (gbest_j - x_j^p(t-1)) \quad (1)$$

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

where

$$pbest^p = [pbest_1^p \quad pbest_2^p \quad \dots \quad pbest_\kappa^p]$$

$$gbest = [gbest_1 \quad gbest_2 \quad \dots \quad gbest_\kappa]$$

$$j = 1, 2, \dots, \kappa;$$

the best previous position of a particle so far is recorded from the previous iteration and represented as $pbest^p$; the position of the best particle among all the particles is represented as $gbest$; $rand()$ returns a uniform random number in the range of $[0, 1]$. In [5], an improved version of PSO is presented, where the constriction factor and inertia weight factor are introduced. Here, when the standard particle swarm optimization with constriction factor and inertia weight factor (SPSO) is used, (1) will be changed to:

$$v_j^p(t) = k \cdot \left\{ w \cdot v_j^p(t-1) + \varphi_1 \cdot rand_j^p(\cdot) \cdot (pbest_j^p - x_j^p(t-1)) + \varphi_2 \cdot rand_j^p(\cdot) \cdot (gbest_j - x_j^p(t-1)) \right\} \quad (3)$$

where w is an inertia weight factor; φ_1 and φ_2 are acceleration constants; k is a constriction factor derived from the stability analysis of equation (3) to ensure the system to be converged but not prematurely [5]. Mathematically, k is a function of φ_1 and φ_2 as reflected in the following equation:

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

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

SPSO utilizes $pbest^p$ and $gbest$ to modify the current search point in order to avoid the particles moving in the same direction, but to converge gradually toward $pbest^p$ 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:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{T} \times t \quad (5)$$

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 (3), the particle velocity is limited by a maximum value v_{\max} . The parameter v_{\max} determines the resolution with which regions are to be searched between the present position and the target position. This limit enhances the local exploration of the problem space and it realistically simulates 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 experience, v_{\max} is often set at 10%–20% of the dynamic range of the element on each dimension.

B. Hybrid particle swarm optimization

We observe that SPSO [5, 9] works well in the early stage, but usually presents problems on reaching the near-optimal solution. The behaviour of SPSO presents some problems with 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, then 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 break through *stagnation*. Here, we called this hybrid PSO as HPSOM. The mutation operation starts with a randomly chosen particle in the swarm, which moves to different positions inside the search area through the mutation. The following mutation operation is used in HPSOM:

$$mut(x_j) = x_j - \omega, r < 0 \quad (6a)$$

$$mut(x_j) = x_j + \omega, r \geq 0 \quad (6b)$$

where x_j is a randomly chosen element of the particle from the swarm, and ω is randomly generated within the range $[0, 0.1 \times (para_{\max}^j - para_{\min}^j)]$, representing one-tenth of the length of the search space. r is a random number between +1 and -1, $para_{\max}^j$ and $para_{\min}^j$ are the upper and lower boundaries of each particle element. The pseudo code of the hybrid PSO with mutation operation is shown in Fig. 1(b), in which the mutation on particles is performed after updating their velocities and positions. It can also be seen from Fig. 1(a) and Fig. 1(b) that the pseudo codes of both PSO methods are identical except the mutation operation is introduced.

However, it can be noticed from (6) that the mutating space in HPSOM is limited by ω . It may not be the best approach in fixing the size of the mutating space all the time along the search. It can be further improved by a dynamic mutation operation in which the mutating space contracts dynamically along the search. We propose a wavelet mutation that varies the mutating space based on the wavelet theory. The resulting HPSOWM (Fig. 2) is identical to HPSOM except for the mutation operation used. The proposed wavelet mutation is discussed in the following sub-section.

C. Wavelet mutation

C.1. Wavelet theory

Certain seismic signals can be modelled by combining translations and dilations of an

oscillatory function within 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:

$$\int_{-\infty}^{+\infty} \psi(x) dx = 0. \quad (7)$$

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

On the other hand, it is possible to show that the admissibility condition implies that $\hat{\psi}(0) = 0$, so that a wavelet must integrate to zero. Notice that $\hat{\psi}$ is the Fourier transform of wavelet ψ and the admissibility condition is defined as follows:

$$0 < C_{\psi} < +\infty \quad (8)$$

where

$$C_{\psi} = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\nu)|^2}{|\nu|} d\nu. \quad (9)$$

Property 2:

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 dx < \infty \quad (10)$$

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:

$$\psi(x) = e^{-x^2/2} \cos(5x). \quad (11)$$

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

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

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad (12)$$

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

$$\psi_{1,0}(x) = \psi(x) . \quad (13)$$

As

$$\psi_{a,0}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x}{a}\right), \quad (14)$$

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.

There are two reasons that wavelet theory is applied to the mutation operation:

1) Improve the solution stability: From (8) and (9) (*Property 1*), the mother wavelet must satisfy an admissibility criterion (which is a kind of half-differentiability.) As a result of this admissibility criterion, the stability of the operation is improved. The solution stability is reflected by the standard deviation of the solutions and can be proved by a set of empirical results. The empirical results will be given in the next section to demonstrate the performance of the solution stability.

2) Fine-tuning ability: By controlling the dilation parameter of the wavelet function, the amplitude of the function can be adjusted. We can use this property to realize a fine-tuning effect to the mutation operation by decreasing the amplitude of the wavelet function to constrain the searching space when the number of iteration increases. Thus, the solution quality can be improved.

C.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, however, exhibits a fine-tuning ability. The details of the operation are as follows. Every particle element of the swarm will have a chance to mutate governed by a probability of mutation, $p_m \in [0 \ 1]$, which is defined by the user. For each particle element, a random number between 0 and 1 will be generated such that if it is less than or equal to p_m , the mutation will take place on that element. For instance, if $\mathbf{x}^p(t) = [x_1^p(t), x_2^p(t), \dots, x_\kappa^p(t)]$ 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 particle element's boundaries $[para_{\min}^j, para_{\max}^j]$), the resulting particle is given by $\bar{\mathbf{x}}^p(t) = [\bar{x}_1^p(t), \bar{x}_2^p(t), \dots, \bar{x}_\kappa^p(t)]$

$$\bar{x}_j^p(t) = \begin{cases} x_j^p(t) + \sigma \times (para_{\max}^j - x_j^p(t)) & \text{if } \sigma > 0 \\ x_j^p(t) + \sigma \times (x_j^p(t) - para_{\min}^j) & \text{if } \sigma \leq 0 \end{cases}, \quad (15)$$

where $j \in 1, 2, \dots, \kappa$; κ denotes the dimension of particle and

$$\sigma = \psi_{a,0}(\varphi) \quad (16)$$

$$= \frac{1}{\sqrt{a}} \psi\left(\frac{\varphi}{a}\right). \quad (17)$$

Different kinds of mother wavelets have been considered during the development of the algorithm, e.g. Mexican hat wavelet (normalized), Mexican hat wavelet, Morlet wavelet, Gaussian wavelet, and Meyer wavelet. By trial and error through experiments for good performance, various wavelet functions have been investigated in terms of cost values. Finally, we choose Morlet wavelet as the mother wavelet in the wavelet mutation operation because the selected wavelet function offers the best performance.

By using the Morlet wavelet in (11) as the mother wavelet,

$$\sigma = \frac{1}{\sqrt{a}} e^{-\left(\frac{\varphi}{a}\right)^2 / 2} \cos\left(5\left(\frac{\varphi}{a}\right)\right). \quad (18)$$

If σ is positive approaching 1, the mutated element of the particle will tend to the maximum value of $x_j^p(t)$. Conversely, when σ is negative ($\sigma \leq 0$) 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 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,

$$\frac{1}{N} \sum_N \sigma = 0 \text{ for } N \rightarrow \infty, \quad (19)$$

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 $[-2.5a, 2.5a]$ 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.

$$a = e^{-\ln(g) \times \left(1 - \frac{t}{T}\right)^{\zeta_{wm}} + \ln(g)} \quad (20)$$

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} and the parameter g to a with respect to t/T are shown in Fig. 5 and Fig. 6, respectively. In this figure, g is set as 10000. Thus, the value of a is between 1 and 10000. Referring to (18), the maximum value of σ is 1 when the random number of $\varphi=0$ and $a=1(t/T=0)$. Then referring to (15), the resulting particle $\bar{x}_j^p(t) = x_j^p(t) + 1 \times (para_{\max}^j - x_j^p(t)) = para_{\max}^j$. It ensures that a large search space for the

mutated 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 $a=4000$; if the random value of φ is zero, the value of σ will be equal to 0.0158. With $\bar{x}_j^p(t) = x_j^p(t) + 0.0158 \times (para_{\max}^j - x_j^p(t))$, a smaller searching space for the mutated element is given for fine-tuning. Changing the parameter ζ_{wm} will change the characteristics of the monotonic increasing function of the wavelet mutation. The dilation parameter a will take a value so as to perform fine-tuning faster as ζ_{wm} is increasing. It is chosen by trial and error, which depends on the kind of the optimization problem. When ζ_{wm} becomes larger, the decreasing speed of the step size (σ) of the mutation becomes faster. In general, if the optimization problem is smooth and symmetric, the searching algorithm is easier to find the solution and the fine-tuning can be done in the early stage. Thus, a larger value of ζ_{wm} can be used to increase the step size of the early mutation. More details about the sensitivity of ζ_{wm} to the wavelet mutation will be discussed in the next section.

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

C.3. Choosing the HPSOWM parameters

HPSOWM is seeking a balance between the exploration of new regions and the exploitation of the already sampled regions in the search spaces. This balance, which critically affects the performance of HPSOWM, is governed by the right choices of the control parameters: Swarm size (γ), the probability of mutation (p_m), the shape parameter (ζ_{wm}) and the parameter g of the wavelet mutation. Some views about these parameters are given as follows:

i) Increasing swarm size (γ) will increase the diversity of the search space, and reduce the probability that HPSOWM prematurely converges to a local optimum. However, it also increases the time required for the population to converge to the optimal region in the search space.

ii) Increasing the probability of mutation (p_m) tends to transform the search into a random search such that when $p_m = 1$, all element of particles will mutate. This probability gives us an expected number ($p_m \times \gamma \times \kappa$) of element of particles that undergo the mutation operation. In other words, the value of p_m depends on the desired number of element of particles that undergo the mutation operation. Normally, when the dimension is very low (number of elements in a particle is less than 5), p_m is set at 0.5 to 0.8. When the dimension is around 5-10, p_m is set at 0.3 to 0.4. When the dimension is in the range of 11 to 100, p_m is set at 0.1 to 0.2. When the dimension is in the range of 101 to 1000, normally p_m is set at 0.05 to 0.1. Lastly, when the dimension is very high (number of elements in a particle is larger than 1000), p_m is set at <0.05 . In principle, when the dimension is high, p_m should be set to a smaller value. It is because if the dimension is high and p_m is set to a larger value, the number of elements of particles undergoing the mutation operation will be large. It will increase the searching time and more importantly destroy the current information about the application in each time of iteration, as all elements of particles are randomly assigned. Generally, by properly choosing the value of p_m , the ratio of the number of elements of particles undergoing mutation operation to the population size can be maintained to prevent the searching process from turning to a random searching one. Thus, the choices of the values of p_m for all the following benchmark functions and application examples are based on this selection criterion, and set by trial and error through experiments for good performance for all functions.

iii) The dilation parameter a is governed by the monotonic increasing function (20), and this monotonic increasing function is controlled by two parameters. They are the shape parameter ζ_{wm} and the parameter g . Changing the parameter ζ_{wm} will change the characteristics of the monotonic increasing function of the wavelet mutation. The dilation parameter a will take a value so as to perform fine-tuning faster as ζ_{wm} is increasing. It is chosen by trial and error, which depends on the kind of the optimization problem. When ζ_{wm} becomes larger, the decreasing speed of the step size (σ) of the mutation becomes faster. In general, if the optimization problem is smooth and symmetric, it is easier to find the solution and the fine-tuning can be done in early iteration. Thus, a larger value of ζ_{wm} can be used to increase the step size of the early mutation. Parameter g is the value of the upper limit of dilation parameter a . A larger value of g implies that the maximum value of a is larger. In other words, the maximum value of $\min(|\sigma|)$ will be smaller

(i.e. a smaller searching limit is given). Conversely, a smaller value of g implies that the maximum value of a is smaller. In other words, the maximum value of $\min(|\sigma|)$ will be larger (i.e. a larger searching limit is given). From our point of view, fixing one parameter and adjusting the other parameter to control the monotonic increasing function is more convenient to find a good setting. In Section III, part D and E, the sensitivity of ζ_{wm} and g to the wavelet mutation with experimental results will be discussed. Based on the results, we suggest fixing the parameter g to 10000 and tuning the parameter ζ_{wm} to optimize the monotonic increasing function for different applications.

III. BENCHMARK TEST FUNCTIONS: RESULTS AND ANALYSIS

A. Benchmark test functions

A suite of eighteen standard benchmark test functions [8, 13] are used to test the performance of HPSOWM. Many different kinds of optimization problems are covered by these benchmark test functions. They can be divided into three categories. The first one is the category of unimodal functions, which is a symmetric model with a single minimum; f_1 to f_7 are unimodal functions. The second one is the category of multimodal functions with a few local minima; f_8 and f_{13} belong to this type. The last one is the category of multimodal functions with many local minima; f_{14} to f_{18} belong to this type. The expressions of these functions are tabulated in Table I. (The details about the parameter a , b , c and function $u(\cdot)$ for function $f_8 - f_9$ and $f_{12} - f_{14}$ are given in [13]).

A.1. Experimental setup

The performance of HPSOM [1], HGAPSO [17], HGPSO [16], SPSO [9], and the proposed HPSOWM on solving the benchmark test functions is evaluated.

The following simulation conditions are used:

- Shape parameter of the wavelet mutation (ζ_{wm}): it is chosen by trial and error through experiments for good performance for all functions. (A discussion for the value of ζ_{wm} will be given in Section III D.)
- Parameter g of the wavelet mutation: 10000. (A discussion for the value of g will be given in Section III E.)

- Acceleration constant φ_1 : 2.05 [9]
- Acceleration constant φ_2 : 2.05 [9]
- Maximum velocity v_{\max} : 0.2 [9]
- Swarm size: 50
- Number of runs: 50
- Probability of mutation for HPSOWM, HPSOM, and HGAPSO (p_m): it is chosen by trial and error through experiments for good performance for all functions. ($p_m = 0.1$ for $f_2 - f_7$; $p_m = 0.2$ for $f_1, f_{14} - f_{18}$; $p_m = 0.3$ for f_{13} ; $p_m = 0.5$ for $f_8 - f_9$ and $f_{11} - f_{12}$; $p_m = 0.8$ for f_{10} .)
- Probability of crossover for HGAPSO (p_c): 0.8
- Initial population: it is generated uniformly at random
- The learning rate of HGAPSO is chosen by trial and error through experiments for good performance for all functions

A.2. Results and analysis

In this section, the results for the 18 benchmark test functions are given to show the merits of HPSOWM. The experimental results in terms of the mean cost value, best cost value, standard deviation and convergence rate are summarized in Table II to Table IV and Figs. 7 - 9.

Category 1: Unimodal function

Function f_1 is a sphere model, which is smooth and symmetric. The main purpose of testing this function is to measure the convergence rate of searching. It is probably the most widely used test function. For this function, the results in terms of the mean cost value and the best cost value of HPSOWM are much better than those of the other methods; the mean cost value of HPSOWM is about 180 to 1.2×10^7 times better. Also, the standard deviation is much better, which means the searched solutions are more stable. In Fig. 7(a), HPSOWM displays a faster convergence rate than other methods thanks to its better searching ability. It reaches approximately 1×10^{-3} in around 500 times of iteration, while the other optimization methods offer about 1×10^{-1} . Function f_2 is a generalized Rosenbrock's function, which is strongly non-separable and the optimum is located in a very narrow ridge. The tip of the ridge is very sharp, and it runs around a parabola. HPSOWM

performs better than the other methods in terms of the mean value and the standard deviation. Also, a good convergence rate for HPSOWM is shown in Fig. 7(b). Function f_3 is a function that is a representation 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, it can get stuck in one of the flat surfaces. All hybrid PSOs that involve the mutation operation are good for this function because it can generate a long jump by using mutation operations in the PSO. Function f_4 is a quadratic function padded with noise, which increases the difficulty for searching the minimum value, since the function would not return the same value at the same point every time. Comparing with the other optimization methods, HPSOWM gives the best mean cost value. Function f_5 is the Schwefel's problem 2.21. From Table II, although the best cost value of HPSOWM is a little bit worse than that of HPSOM, the mean cost value and the standard derivation of the HPSOWM are the best. Thus, HPSOWM gives a better solution quality and stability. Function f_6 is the Schwefel's problem 2.22 and function f_7 is the Eason's function. For these problems, the performance of HPSOWM is better than that of the other methods. The rapid convergence of HPSOWM as shown in Fig. 7(c)-7(d) supports our argument. In short, HPSOWM is the best to tackle unimodal functions comparing with the other methods.

Category 2: Multimodal function with a few local minima

For functions $f_8 - f_{13}$, which are multimodal functions with only a few local minima, different results from the proposed methods are obtained. The experimental results for these functions are tabulated in Table III. Among these functions, four of them ($f_8, f_{10} - f_{12}$) do not show significant differences among the different optimization methods. They all reach or get near to the global optima, but HPSOWM still provides the smallest standard deviation in most cases. For functions f_9 and f_{13} , different results from HPSOWM and the other methods are obtained. HPSOWM gives better results in terms of mean cost value and standard deviation. Thus, the solution's stability and quality are good. According to Figs. 8(a) - (b), the convergence rate of HPSOWM is faster than those of the others.

Category 3: Multimodal function with many local minima

Functions f_{14} to f_{18} are multimodal functions with many local minima. The experimental results for these functions are tabulated in Table IV. Functions f_{14} and f_{16} are the Generalized

Penalized function and the Generalized Rastrigin's function respectively. It can be seen clearly from Figs. 9(a) - (b) that if the PSO does not involve any mutation operation (HGPSO and SPSO), it will be easily trapped at some local minimum. From the results obtained, the mean cost value, the best cost value and the standard deviation of HPSOWM are better than those of the other methods. The HPSOWM can provide more stable and high-quality results. Functions f_{17} and f_{18} are the Ackley's and the Schwefel's functions respectively. From Table IV, we can see that HPSOWM gives better results than the others. In general, HPSOWM is good for handling multimodal functions with many local minima.

In conclusion, HPSOWM gives the best performance for all kinds of optimization problems, especially unimodal functions and multimodal functions with many local minima. It generally outperforms other hybrid PSOs and SPSO.

B. 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.

$$t = \frac{\bar{\alpha}_2 - \bar{\alpha}_1}{\sqrt{\left(\frac{\sigma_2^2}{\xi + 1}\right) + \left(\frac{\sigma_1^2}{\xi + 1}\right)}} \quad (21)$$

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 1.645 ($\xi = 49$), there is a significant difference between the two algorithms with a 95% confidence level. The t -values between HPSOWM and the other optimization methods are shown in Table V. We see that most t -values in this table are higher than 1.645. Therefore, the performance of HPSOWM is significantly better than the other optimization methods with a 95% confidence level.

C. Additional benchmark test functions with shift

In addition, a suite of five benchmark test functions [44] with shift are used. In order to avoid the problems existing in some benchmark functions [45] that they have the same values among all independent variables at the global optima and there is no linking among these variables, we shift the global optimum points and rotate the test functions. That means in order to make the variables to have different numerical values at the optimum point, we randomly generate the global optimum point within a given search space. The search ranges of the variables are also adjusted according to the randomly generated global optimum point in order to avoid different variables to have the same numerical value after normalization at the global optimum point. We test five additional functions with shift: Shifted sphere function ($f_{1\text{-shift}}$), Shifted Schwefel's problem 1.2 ($f_{2\text{-shift}}$), Shifted rotated high conditioned elliptic function ($f_{3\text{-shift}}$), Shifted Rosenbrock's function ($f_{4\text{-shift}}$), and Shifted Rastrigin's function ($f_{5\text{-shift}}$). The first three functions are unimodal functions and the last two are multi-modal functions. The equations, dimension and the range of the variables are given in Table VI. The experimental result in terms of the mean cost value, best cost value, standard deviation, t -value and convergence rate are summarized in Table VII and Fig. 10. The basic experimental setup is the same as that mentioned in Section III, A.1. The shape parameters of the wavelet mutation for $f_{1\text{-shift}}$ to $f_{4\text{-shift}}$ are set at 5, and that for $f_{5\text{-shift}}$ is set at 2. The probability of mutation for HPSOWM, HPSOM, and HGAPSO is set at 0.2 for all functions, which is chosen by trial and error through experiments for good performance. From the table and the figure, we can see that HPSOWM, HPSOM, and HGAPSO show better performance in terms of the mean cost value and standard deviation than HSPSO and SPSO. Based on this observation, we can see that when the PSO is without mutation operation (HSPSO and PSO), it is hard to solve the optimization problems with the global optimum points shifted and rotated. Comparing with the PSO with mutation operation (HPSOWM, HPSOM, and HGAPSO), the performance of HPSOWM is the best in terms of the mean value, standard deviation and convergence rate.

D. Sensitivity of the shape parameter for wavelet mutation

The mean cost values offered by HPSOWM with different values of the wavelet mutation's shape parameter ζ_{wm} for all test functions in part A are tabulated in Table VIII. The functions are tested by using $\zeta_{wm} = 0.2, 0.5, 1, 2, \text{ and } 5$. In this experiment, the parameter g is fixed at 10000. If

the optimization problem needs a more significant mutation to reach the optimal point, a smaller ζ_{wm} should be used. Conversely, if HPSOWM needs to perform the fine-tuning faster, a larger ζ_{wm} should be used. For example, the function f_1 is a sphere model which is smooth and symmetric. Searching algorithms should be fast to jump to the area near the global optimum and then perform fine-tuning. Therefore, a larger ζ_{wm} can be set ($\zeta_{wm}=5$) so that HPSOWM will perform fine-tuning faster. On the other hand, ζ_{wm} can be set as 0.2 for f_3 (step function problem) where the mutation operation is playing a significant role at the later stage. In some cases, ζ_{wm} 's value is not very critical, e.g. in f_7 and f_{11} . For f_7 , the mean cost value for different values of ζ_{wm} is nearly the same. (The best performance is obtained when $\zeta_{wm}=0.5$ because the standard deviation of the HPSOWM for $\zeta_{wm}=0.5$ is the smallest.) However, in some cases, the value of the parameter ζ_{wm} is sensitive to the performance of the searching, e.g. in f_1 and f_{16} . In conclusion, no formal method is available to choose the value of the parameter ζ_{wm} , it depends on the characteristics of the optimization problems.

E. Sensitivity of the parameter “g” for wavelet mutation

The mean cost values offered by HPSOWM with different values of the wavelet mutation's parameter g for all test functions are tabulated in Table IX. The functions are tested by using $g = 100, 1000, 10000, \text{ and } 100000$. In this experiment, the parameter ζ_{wm} is fixed at 5. If we want a smaller value of the upper limit (searching limit) of the particle σ 's mutated element, a larger value of g should be used. In some case, the parameter g is not very sensitive, such as f_1 - f_3 , f_5 - f_8 , f_{10} - f_{14} , and f_{16} - f_{18} . The mean cost values with different value of g have no significant difference. However, in some case, such as f_9 , the value of the parameter g is sensitive to the performance of HPSOWM. In f_9 , the mean cost value is 5.1478×10^{-3} when $g = 100$ and the mean cost value is 1.3275×10^{-3} when $g = 100000$. The difference of them is around 4 times. In conclusion, similar to the parameter ζ_{wm} , no formal method is available to choose the value of the parameter g , and it depends on the characteristics of the optimization method. Comparing with the sensitivity of the shape parameter ζ_{wm} , the parameter g is less sensitive to the performance of the searching. With the results in Table IX, we can see that $g = 10000$ give a better performance in general. As mention

in section II, we suggest fixing one parameter and adjusting another parameter to control the monotonic increasing function. By doing so, it is more convenient to find a good setting. Thus, we fix the parameter g to 10000 and adjust the shape parameters ζ_{wm} to obtain an optimal monotonic increasing function for the wavelet mutation operation.

IV. INDUSTRIAL APPLICATIONS OF HYBRID PSO

In this section, three industrial application examples on solving some load flow problems (multi-contingency transient stability constrained optimal power flow problem and economic load dispatch with valve-points loading problem), modelling development of fluid dispensing process for electronic packaging, and design of a neural network based controller, are used to illustrate the performance and applicability of the proposed hybrid PSO.

A. Application I: Load flow problems

In this section, two application examples on load flow problems are given to show the performance of HPSOWM. The problems are multi-contingency transient stability constrained optimal power flow (MC-TSCOPF) and economic load dispatch with valve-points loading (ELD-VPL). Load flow problem is a multi-modal problem, which is suitable to be solved by PSO.

A.1. Multi-contingency transient stability constrained optimal power flow (MC-TSCOPF)

It aims to achieve an optimal solution of a specific objective function, such as fuel cost, network loss, etc. by setting some system control variables, while ensuring the system to withstand specified contingencies (disturbances) and reach an acceptable steady-state operating condition [29]. On solving the MC-TSCOPF problem, the difficulty mainly comes from the non-convexity nature of OPF and the nonlinear differential-algebraic equations which describe the transient stability constraints of the power system. Nonlinear and semi-definite programming [30-31] techniques have been proposed to solve the MC-TSCOPF problem. However, not only their formulation is complex and heavily tied to the system models, but also they rely on convexity to obtain the global optimum solution and as such are forced to simplify some conditions in order to ensure convexity [32]. Similar to MC-TSCOPF problems, reactive power and voltage control problems, that are also mixed-integer nonlinear optimization problems, can be solved by

MC-TSCOPF with more promising results than the tested methods [47]. Here, a global optimization method, such as PSO, is a good tool for handling the MC-TSCOPF problem.

A.1.1 Mathematical model for MC-TSCOPF

The problem of MC-TSCOPF is mathematically defined as follows.

$$\min f(\mathbf{x}, \mathbf{y}) \quad (22)$$

such that

$$\mathbf{g}(\mathbf{x}, \mathbf{y}) = 0 \quad (23a)$$

$$\mathbf{H}(\mathbf{x}, \mathbf{y}) \leq 0 \quad (23b)$$

$$\mathbf{U}(\mathbf{x}(t), \mathbf{y}) \leq 0, t \in T \quad (23c)$$

where $\mathbf{x}(t)$ is a dependent vector which includes active and reactive power of the swing bus, voltage angle and reactive power of the generator buses, and voltage angle and magnitude of the load buses; $T = [t_0, t_{cl}) \cup (t_{cl}, t_e]$ is the transient period from the occurrence of the disturbance at time t_0 to the clearing time t_{cl} and then to the ending time t_e ; \mathbf{x} represents the initial value of $\mathbf{x}(t)$ at $t = 0$. \mathbf{y} is a control which includes the active power and voltage magnitude of the generator buses, the voltage angle and magnitude of the swing bus, and the tap position of load tap changers (LTCs). $f(\cdot)$ can be expressed as the total generation cost, total network loss, corridor transfer power, total cost of compensation, etc. \mathbf{g} is the set of equality constraints which are usually the power flow constraints for a specified operating condition. \mathbf{H} is the set inequality constraints for the steady-state security limits like bus voltage magnitude limits, generator power limits, thermal limits for transmission lines, etc. The dynamic security constraints set \mathbf{U} is infinite in the functional space. For more details, readers are referred to Mo *et al.* [33].

Since the equality constraints \mathbf{g} are imposed implicitly by the power flow calculation incorporated within the algorithm, and the inequality constraints \mathbf{H} is directly satisfied by the PSO, the MC-TSCOPF can be formulated as a penalty function problem:

$$\tilde{F}(\mathbf{x}) = \min \left\{ f(\mathbf{x}, \mathbf{y}) + \beta \max [\mathbf{U}(\mathbf{x}(t), \mathbf{y})^2] \right\}. \quad (24)$$

Generally, transient stability constraints can be considered as hard constraints that should not be violated whilst the static constraints are soft in nature that slight violation could be tolerated. Comparing with other constraint handling approaches [34-35], the penalty function offers a simple

and flexible strategy to effectively deal with mixed hard and soft constraints. In addition, there is no need to have separate penalty factors for each type of constraints. In (24), any transient instability would introduce a huge angle deviation and thus produce a large violation and discrimination even though the same penalty factor is used for all type of violations. Typically, $\beta = 1000$ works very well in most power systems [33].

A.1.2. Case study

As a case study of solving the optimal power flow problems with stability constraints, the New England 39-bus system is used to demonstrate the effectiveness and robustness of the proposed hybrid PSO based approach for solving the MC-TSCOPF problems. For comparison purpose, HPSOM [1], HGAPSO [17], HGPSO [16], and SPSO [9] are also used in this case study. The system data of the power system are collected in [36-37]. The New England 39-bus test system comprises 10-generator, 39-bus, and 46-line. The Power System Toolbox [36] is employed to perform time-domain transient stability simulations for determining the generator rotor trajectories. The time step adopted is 0.01s and the integration time interval is fixed at 1.5s. The total load for the operating condition considered is 6,098 MW and 1,409 MVAR. There are three on load tap changers connecting buses 11-12, 12-13 and 19-20.

After a complete scan of all possible single line fault contingencies, the following two conflicting contingencies were identified.

Contingency 1: A three phase fault occurred at the end of line 26-27 near bus 26. The fault was cleared by tripping the line at bus 26 after 110 ms and at bus 27 after 120 ms.

Contingency 2: A three phase fault occurred at the end of line 16-17 near bus 16. The fault was cleared by tripping the line at bus 16 after 80 ms and at bus 17 after 100 ms.

The case of the transient stability constrained OPF with contingency 1 and 2 is considered. The basic settings of the parameters of the PSOs are the same as those in Section III. The number of iteration is set at 150. The dimension of this case is 22. The probability of mutation (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are set at 0.1 and 0.5 respectively. The shape parameter of the wavelet mutation ζ_{wm} is chosen by trial and error through experiments for good

performance. In this case, $\zeta_{wm} = 0.2, 0.5, 1, 2, \text{ and } 5$ are tried. Among them, $\zeta_{wm} = 0.5$ gives the best result. The experimental results are tabulated in Table X, and the comparison between different PSOs is shown in Fig. 11. The table shows that the mean cost value, the best cost value and the standard deviation offered by HPSOWM are the smallest. The small standard deviation of HPSOWM implies it provides a stable and quality solution (solution is robust). Also the t -values between HPSOWM and other optimization methods are higher than 2.06, and thus HPSOWM is significantly better with a 98% confidence level. From these results, we can see that the proposed HPSOWM provides a stable and quality solution for the multi-contingency transient stability constrained optimal power flow problem.

A.2. Economic load dispatch with valve-points loading (ELD-VPL)

Economic load dispatch (ELD) is a method to schedule power generator outputs with respect to the load demands and to operate a power system economically, so as to minimize the operation cost of the power system. The input-output characteristics of modern generators are nonlinear by nature because of the valve-point loadings and rate limits. Thus the characteristics of ELD-VPL problems are multimodal, discontinuous and highly nonlinear. PSO has been employed to solve the ELD-VPL problem.

A.2.1 Mathematical model for ELD-VPL

The ELD-VPL problem can be formulated into the following objective function:

$$\text{Min} \sum_{i=1}^n C_i(P_{L_i}), \quad (25)$$

where $C_i(P_{L_i})$ is the operation fuel cost of generator i , and n denotes the number of generators. The problem is subject to balance constraint and generating capacity constraints as follows:

$$D = \sum_{i=1}^n P_{L_i} - P_{Loss}, \quad (26)$$

$$P_{L_{i,\min}} \leq P_{L_i} \leq P_{L_{i,\max}}, \quad i = 1, 2, \dots, n \quad (27)$$

where D is the load demand, P_{L_i} is the output power of the i -th generator, P_{Loss} is the transmission loss, $P_{L_{i,\max}}$ and $P_{L_{i,\min}}$ are the maximum and minimum output powers of the i -th generator

respectively.

The operation fuel cost function with valve-point loadings of the generators is given by,

$$C_i(P_{L_i}) = a_i P_{L_i}^2 + b_i P_{L_i} + c_i + \left| e_i \times \sin \left(f_i \times (P_{L_{i,\min}} - P_{L_i}) \right) \right|, \quad (28)$$

where a_i , b_i , and c_i , are coefficients of the cost curve of the i -th generator, e_i and f_i are coefficients of the valve-point loadings. (The generating units with multivalve steam turbines exhibit a greater variation in the fuel-cost functions. The valve-point effects introduce ripples in the heat-rate curves.)

A.2.2 PSO for ELD-VPL

In this section, PSO is used to solve the ELD problem. The particle (solution representation) is defined as follows:

$$\mathbf{p} = [P_{L_1} \quad P_{L_2} \quad P_{L_3} \quad \cdots \quad P_{L_{n-1}}], \quad (29)$$

where n denotes the number of generators and $P_{L_{i,\min}} \leq P_{L_i} \leq P_{L_{i,\max}}$, $i = 1, 2, \dots, n$. From (26), we have,

$$P_{L_n} = D - \sum_{i=1}^{n-1} P_{L_i} + P_{Loss}. \quad (30)$$

In this paper, the power loss is not considered. So,

$$P_{L_n} = D - \sum_{i=1}^{n-1} P_{L_i}. \quad (31)$$

To ensure P_{L_n} falls within the range $[P_{L_{n,\min}}, P_{L_{n,\max}}]$, the following conditions are considered:

$$\text{if } P_{L_n} > P_{L_{n,\max}} \begin{cases} P_{L_1} = P_{L_1} + (P_{L_n} - P_{L_{n,\max}}) \\ P_{L_n} = P_{L_{n,\max}} \end{cases}, \quad (32)$$

$$\text{if } P_{L_n} < P_{L_{n,\min}} \begin{cases} P_{L_1} = P_{L_1} - (P_{L_{n,\min}} - P_{L_n}) \\ P_{L_n} = P_{L_{n,\min}} \end{cases}. \quad (33)$$

It should be noted from (32) and (33) that if the value of P_{L_n} is outside the constraint boundary. The exceeding portion of the power will be shared by other generators in order to make sure that the output powers of all generators are within the safety range. The objective is to minimize the cost

function of (28) by using PSO.

A.2.3. Case study

In this section, different hybrid PSO methods are applied to a 40-generator system, which is adopted as an example in [46]. The system is a very large one with nonlinearities. The load demands (D) is 10500MW. HPSOWM, HPSOM [1], HGAPSO [17], HGPSO [16], and SPSO [9] are used to solve the ELD-VPL problem. The basic settings of the parameters of the PSOs are the same as those in Section III. All the simulation results are averaged ones out of 50 runs. The dimension of this case is 39. The probability of mutation (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are set at 0.1 and 0.5 respectively. Similar to the example MC-TSCOPF, the shape parameter of the wavelet mutation ζ_{wm} is chosen by trial and error through experiments for good performance. In this case, $\zeta_{wm} = 0.2, 0.5, 1, 2, \text{ and } 5$ are tried. For all approaches, the number of iterations is 2000. The statistical results in terms of the mean cost value, best cost value, standard deviation and the t -test value, the running time and the ranking are shown in Table XI. The convergence rates of different PSOs are shown in Fig. 12. From Table XI, we can see that HPSOWM is the best in terms of costs, t values, and standard deviations. The average cost for the 40-generator system is \$122844.4 and the best (minimum) cost is \$121915.3. All t values are higher than 2.06, implying that HPSOWM is significantly better, with a 98% confidence level, than the other hybrid PSOs. Thanks to the wavelet properties, the stability of the optimization is improved and the smallest standard deviation is obtained by using HPSOWM. To conclude, both the solution quality and stability of HPSOWM are better.

B. Application II: Modelling the fluid dispensing for electronic packaging (MFD-EP)

Fluid dispensing is a manufacturing process by which fluid materials are delivered to substrates, boards or work-pieces in a controllable manner. This process is widely used in various packaging processes in the electronics and semiconductor manufacturing industry such as integrated circuit encapsulation, die bonding and surface mount technology. In the competitive market of today, this manufacturing process needs to be well controlled at each of the many processing steps in the manufacturing line. The process directly affects the overall quality of the finished product, as well as the throughput of the production line. All the variables controlling the

desired outputs in a given process need to be understood and optimized for tight control. To achieve this, it is necessary to develop an accurate model for describing the process.

Neural networks have been used to develop the process models for various manufacturing processes such as abrasive flow machining [38], grinding [39] and die casting [40]. They have the capability to transform a nonlinear mathematical model into a simplified black-box structure. The advantages of using the neural network approach to process modelling are that it can provide learning and generalization abilities for nonlinearities. In this paper, a feed-forward neural network trained by the hybrid PSOs for modelling fluid dispensing for electronic packaging is given to illustrate the merits of the proposed PSO.

B.1. Fluid dispensing process for electronic packaging

Fluid dispensing is an important and popular process for electronics packaging. In this paper, modelling the fluid dispensing for microchip encapsulation is studied. Normally, silicon chips are covered using an X-Y numerically controlled dispensing system that delivers fluid encapsulant through a needle. The material is commonly dispensed in a pattern, working from the centre out. A fluid dam around the die site and second wire bond points can be made to contain the flow material and make a uniform shape as shown in Fig. 13.

Modelling the fluid dispensing process is critical for understanding the process behaviour and achieving the process optimization. To develop a model for relating the process parameters to the quality characteristics of the fluid dispensing, significant process parameters and quality characteristics have to be identified first. With the assistance from the supporting company of this research, three significant process parameters and their normal operating ranges were identified as follows.

- ◆ The compressed air pressure (1 bar to 4 bar), x_1
- ◆ The pump motor speed (400 rpm to 1000 rpm), x_2
- ◆ The height between the substrate and the needle (250 to 2000 steps of stepping motor), x_3 .

Two quality characteristics were studied, which are the encapsulation weight (mg), y , and the encapsulation thickness (mm), z .

B.2. Modelling with neural network

A three-layer feed-forward neural network is used to model the fluid dispensing process. Its structure, as shown in Fig. 14, consists of an input layer in which the input vectors (including process parameters x_1 , x_2 and x_3) are fed, the output layer which produces the output response (either one of the quality characteristics y or z), and one hidden layer in between. The hidden layer links the input and output layers together and allow for complex, nonlinear interactions among the inputs to produce the desired output.

Referring to Fig. 14, the input-output relationship of the proposed three-layer neural networks for the encapsulation weight y and the encapsulation thickness z can be written as follows:

$$y = \sum_{j=1}^{n_h} w_j \text{logsig} \left[\sum_{i=1}^3 (v_{ji} x_i - b_j^1) \right] - b^2 \quad (34)$$

$$z = \sum_{j=1}^{n'_h} w'_j \text{logsig} \left[\sum_{i=1}^3 (v'_{ji} x_i - b_j^1) \right] - b^2 \quad (35)$$

where n_h (or n'_h) denotes the number of the hidden nodes; w_j (or w'_j), $j = 1, 2, \dots, n_h$ (or n'_h), denotes the weight of the link between the j -th hidden node and the output node; v_{ji} (or v'_{ji}), $i = 1, 2, 3$ and $j = 1, 2, \dots, n_h$ (or n'_h), denotes the weight between the i -th input node and the j -th hidden node; b_j^1 (or b_j^1) and b^2 (or b^2) denote the biases for the j -th hidden node and output node respectively; $\text{logsig}(\cdot)$ denotes the logarithmic sigmoid function:

$$\text{logsig}(\alpha) = \frac{1}{1 + e^{-\alpha}}, \quad \alpha \in \mathfrak{R}. \quad (36)$$

To develop the neural network based model for the fluid dispensing process, values of the neural network parameters (i.e.: w_j , v_{ji} , b_j^1 and b^2 with $i = 1, 2, 3$ and $j = 1, 2, \dots, n_h$) and the number of hidden-nodes (n_h) used in the hidden layer need to be determined. These two settings are important because they affect the prediction accuracy of the neural network based process model.

To tune the parameter values of the network, we use the hybrid PSO to minimize the mean square error (MSE) by setting the swarm particle to be $[v_{ji} \quad w_j \quad b_j^1 \quad b_j^2]$ for all i and j . The MSE for the encapsulation weight y and for the encapsulation thickness z are defined as follows:

$$\text{MSE}_y = \frac{\sum_{k=1}^{n_{pat}} (d_k^y - y_k)^2}{n_{pat}} \quad (37)$$

$$\text{MSE}_z = \frac{\sum_{k=1}^{n_{pat}} (d_k^z - z_k)^2}{n_{pat}} \quad (38)$$

where d_k^y and d_k^z denotes the desired value of the encapsulation weight y and the encapsulation thickness z respectively; n_{pat} denotes the number of patterns. After training, the values of these network parameters will be fixed during the operation. The total number of tuned parameters (n_{para}) of the neural network is the sum of the number of parameters between the input and hidden layers, and the number of parameters between the hidden and output layers. Hence,

$$n_{para} = (n_{in} + 1)n_h + (n_h + 1)n_{out} \quad (39)$$

where n_{in} and n_{out} denote the number of input nodes and number of output nodes respectively. For this application, $n_{in} = 3$, $n_{out} = 1$. Thus, $n_{para} = 5n_h + 1$.

B.3. Case study

MFD-EP is a multimodal system. To train the neural network of the MFD-EP system, 87 experimental data of encapsulation weight and encapsulation thickness are used. The training patterns consist of the input vectors and their corresponding expected outputs. In order to test the learning ability of the neural network trained by the proposed HPSOWM, a set of 9 testing patterns is used. For comparison purpose, the neural network models are also trained by HPSOM [1], HGAPSO [17], HGPSO [16], and SPSO [9]. The basic settings of the parameters of the PSOs are the same as those in Section III. The initial ranges of the weights of the neural networks for the

encapsulation weight and the encapsulation thickness are bounded between -4 and 4 . The number of iteration is set at 2000. The probability of mutation (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are set at 0.1 and 1 respectively. The number of hidden nodes (n_h) of the neural network for the encapsulation weight and the neural network for the encapsulation thickness are set at 5 and 7 respectively. In other words, the total numbers of parameters (dimension) are 26 and 36 respectively. The training results are tabulated in Table XII, and the comparisons between different PSOs are shown in Fig. 15 and Fig. 16. The table shows that the mean value, best cost value and standard deviation offered by HPSOWM are the smallest. Also the t -values between HPSOWM and the other optimization methods are higher than 2.06, and thus HPSOWM is significantly better than the other methods with a 98% confidence level. The computational time of HPSOWM is near to that of the other PSOs. (HGPSO needs much more time than others). Nine validation tests are carried out to evaluate the generalization ability of the neural networks with different PSO methods. Table XIII (a) and (b) show the validation results yielded by the neural network models for the encapsulation weight and the encapsulation thickness respectively. From these tables, HPSOWM gives the smallest mean error and standard deviation. The proposed HPSOWM indeed provides a quality and stable solution for tuning the neural network model for the fluid dispensing process in electronic packaging.

C. Application III: Neural network-based controller (NN-BC)

In this application, a neural network-based controller realized by a three-layer feed-forward fully-connected neural network is proposed to stabilize a mass-spring-damper system [41]. The open-loop system can be described as follows.

$$\ddot{x}(t) = -1.27x(t) - 0.1x(t)^3 - 0.1\dot{x}(t) + (1.5387 - 0.13\dot{x}(t)^2)u(t) \quad (40)$$

where u is the force, $x(t)$ and $\dot{x}(t)$ are the displacement and velocity of the mass respectively. This problem is considered as a multi-modal optimization problem. A two-input-two-output neural network with 4 hidden nodes is employed to close the feedback loop. The total number of network parameter is 22. Denoting the outputs of the neural network as $y_1(t)$ and $y_2(t)$, the neural network-based controller takes $x(t)$ and $\dot{x}(t)$ as the inputs and a scalar s as the gain to produce the control signal $u(t)$. Hence, the neural-network-based controller is defined as follows.

$$u(t) = s(y_1(t)x(t) + y_2(t)\dot{x}(t)) \quad (41)$$

The control objective is to stabilize the mass-spring-damper system of (40), i.e. $x(t) \rightarrow 0$ and $\dot{x}(t) \rightarrow 0$ as $t \rightarrow \infty$. To measure the system performance, we consider the following scalar performance index [42].

$$J = \int_0^5 \mathbf{x}(t)^T \mathbf{W} \mathbf{x}(t) dt \quad (42)$$

where $\mathbf{x}(t) = [x(t) \quad \dot{x}(t)]^T$ and $\mathbf{W} = \begin{bmatrix} 500 & 0 \\ 0 & 1 \end{bmatrix}$. It can be seen that the performance index J is contributed by the integral of the energy of the system state vector of $\mathbf{x}(t)$. A smaller value of J indicates a better system performance. By employing different weighting matrix \mathbf{W} , the contribution of the system states to the performance index can be changed to meet different system performance specification. In this example, the weight for $x(t)$ is 500 times higher than that of $\dot{x}(t)$ as the response of $x(t)$ is more concerned. The proposed HPSOWM is employed to minimize the values of J by searching the best values of the connection weights of the neural network and the scalar of s under the initial system state $\mathbf{x}(0) = \begin{bmatrix} \frac{22\pi}{45} & 0 \end{bmatrix}^T$.

For comparison purpose, the neural-network-based controller is also trained by HPSOM [1], HGAPSO [17], HGPSO [16], and SPSO [9]. The basic settings of the parameters of the PSOs are the same as those in Section III. The initial values of the connection weights and the scalar s are generated randomly in the ranges of -1 to 1 and -200 to 200 respectively. The number of iteration is set at 50. The probability of mutation (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are set at 0.05 and 0.2 respectively. 50 runs of training for each learning method are conducted. The training results for various learning methods are tabulated in Table XIV, which shows the mean cost values, best cost values and standard deviations offered by various learning methods. It can be seen that HPSOWM offers the best performance. Also the t -values between HPSOWM and other optimization methods are higher than 2.06, and thus HPSOWM is significantly better than other methods with a 98% confidence level.

V. CONCLUSION

In this paper, we have proposed a hybrid PSO incorporated with wavelet mutation (HPSOWM). Our objective is to apply the properties of the wavelet theory to enhance PSO, so that it can explore the solution space more effectively on reaching the solution. Simulation results have shown that the proposed wavelet mutation based hybrid PSO is a useful tool to solve optimization problems. Thanks to the properties of the wavelet, the solution stability and quality of the hybrid PSO are improved. On solving a suite of benchmark test functions, HPSOWM gives better results than the methods of HPSOM, HGAPSO, HGPSO and SPSO. Also, a faster convergence speed can be achieved by HPSOWM. Comparing their run time (computation time), HGPSO consumes more time because of the process of the gradient descent. The other methods, including HPSOWM, consume almost the same amount of time. To illustrate the applicability of the proposed hybrid PSO, three industrial applications are studied. From the obtained results, HPSOWM shows better performance than other existing PSO methods.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by The University of Western Australia, Australia, and a grant from the Hong Kong Polytechnic University.

REFERENCES

- [1] A. A. E. Ahmed, L. T. Germano, and Z. C. Antonio, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. 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 Symp. 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*, vol. 1447, Lecture Notes in Computer Science. New York: Springer-Verlag, pp. 611-616, 1998.
- [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 Int. Conf. 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," *IET Proceedings – Generation, Transmission and Distribution*, vol. 1, no.3, pp. 476-483, May 2007.
- [10] A. Neubauer, "A theoretical analysis of the non-uniform mutation operator for the modified genetic algorithm," in *Proc IEEE Int. Conf. Evolutionary Computation*, Indianapolis, 1997, pp. 93-96.
- [11] R.J.M. Vaessens, E.H.L. Aarts, and J.K. Lenstra, "A local search template," in *Proc Parallel Problem-Solving from Nature 2*, 1992, pp. 65-74.
- [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," in *Proc Evolutionary Computing: AISB Workshop*, 1994, pp. 115-130.
- [15] J.M. Zurada, *Introduction to Artificial Neural Systems*. West Info Access, 1992.
- [16] M.M. Noel and T.C. Jannett, "Simulation of a new hybrid particle swarm optimization algorithm," in *Proc 36th Southeastern Symposium on System Theory*, 2004, pp. 150-153.
- [17] C.F. Juang, "A hybrid genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. on Systems, Man and Cybernetics: Part B: Cybernetics*, vol. 34, no. 2, pp. 997-1006, 2004.
- [18] T.O. Ting, T.O., M.V.C. Rao, and C.K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," *IEEE Trans. Power Systems*, vol. 21, no. 1, pp. 411-418, Feb 2006.
- [19] B. Zhao, C.X., Guo, and Y.J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. Power Systems," *IEEE Trans. Power Systems*, vol. 20, no. 2, pp. 1070-1078, May 2005.
- [20] J.G. Vlachogiannis and K.Y. Lee, "A comparative study on particle swarm optimization for optimal steady-state performance of power systems" *IEEE Trans. Power Systems*, vol. 21, no. 4, pp. 1718-1728, Nov 2006.
- [21] C.M. Huang, C.J. Huang, and M.L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Trans. Power Systems*, vol. 20, no. 2, pp. 1126-1133, May 2005.
- [22] C. Zhang, H. Shao, and Y. Li, "Particle swarm optimisation for evolving artificial neural network," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 4, Oct., 2000, pp. 2487-2490.
- [23] Y. Song, Z. Chen, and Z. Yuan, "New chaotic PSO-based neural network predictive control for nonlinear process," *IEEE Trans. Neural Networks*, vol. 18, no. 2, pp. 595-601, Mar 2007.

- [24] Y.S. Wang, K.J. Wang, J.S. Qu, and Y.R. Yang, "Adaptive inverse control based on particle swarm optimization algorithm," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, Jul 2005, pp. 2169-2172.
- [25] W. Liu, K. Wang, B. Sun, and K. Shao, "A hybrid particle swarm optimization algorithm for predicting the chaotic time series mechatronics and automation," in *Proc. IEEE Int. Conf. Mechatronics and Automation*, Jun 2006, pp. 2454-2458.
- [26] R. Marinke, E. Araujo, Ld.S. Coelho, and I. Matiko, "Particle swarm optimization (PSO) applied to fuzzy modeling in a thermal-vacuum system," in *Proc. 5th Int. Conf. Hybrid Intelligent Systems*, Nov 2005, pp. 67-72.
- [27] Y. Liu and X. He, "Modeling identification of power plant thermal process based on PSO algorithm," in *Proc. American Control Conference*, Jun 2005, no. 7, pp. 4484-4489.
- [28] X.H. Shi, Y.H. Lu, C.G. Zhou, H.P. Lee, W.Z. Lin, and Y.C. Liang, "Hybrid evolutionary algorithms based on PSO and GA," in *Proc. IEEE Congress Evolutionary Computation*, 2003, vol. 4, pp. 2393-2399.
- [29] J. Carpentier, "Optimal power flows," *Int. J. Elect. Power Energy Syst.*, vol. 1, pp. 3-15, 1979.
- [30] J. A. Momoh, R. J. Koessler, and M. S. Bond "Challenges to optimal power flow," *IEEE Trans. Power Systems*, vol. 12, no. 1, pp. 444-455, Feb. 1997.
- [31] C. A. Poa-Sepulveda and B. J. Pavez-Lazo, "A solution to the optimal power flow using simulated annealing," *Int. J. Electrical Power & Energy Systems*, vol. 25, no. 1, pp. 47-57, Jan. 2003.
- [32] J. A. Momoh and J. Z. Zhu, "Improved interior point method for OPF problems," *IEEE Trans. Power Systems*, vol. 14, no.3, pp. 1114-1120, Aug. 1999.
- [33] N. Mo, G.T.Y. Pong, K.W. Chan, and S.W. Mei, "Multi-contingency transient stability constrained optimal power flow by genetic algorithm," in *Proc. 7th Conf. Advances in Power System Control, Operation and Management*, 2006.

- [34] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evolutionary Computation*, vol. 7, pp. 445-455, 2003.
- [35] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *Evolutionary Computation*, vol. 4, pp. 284-294, 2000.
- [36] J. H. Chow, *Power System Toolbox Version 2.0*. Cherry Tree Scientific Software, 2000.
- [37] R. Zimmerman and D. Gan, *MATPOWER: A Matlab Power System Simulation Package*. 2005.
- [38] K.L. Petri, R.E. Billo, and B. Bidanda, "A neural network process model for abrasive flow machining operations," *Journal of Manufacturing System*, vol. 17, no.1, pp. 52-65, 1998.
- [39] E. Brinksmeier, H.K. Tonshoff, C. Czenkusch, and C. Heinzl, "Modelling and optimisation of grinding processes," *Journal of Intelligent Manufacturing*, vol. 9, pp. 303-314, 1998.
- [40] K.D.V. Prasad and P. Yarlagadda, "Prediction of die casting process parameters by using an artificial neural network model for zinc alloys," *Int. J. Production Research*, vol. 38, no. 1 119-139, 2000.
- [41] B. C. Kuo, *Automatic Control Systems*, Eighth Edition. Prentice Hall, 2003.
- [42] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*. Prentice Hall, 1990.
- [43] P. Angeline, "Using selection to improve particle swarm optimization", in *Proc. IEEE Int. Conf. Evolutionary Computation, Anchorage*, May 1998, pp.84-89.
- [44] J.J. Lang, S. Baskar, P. Suganthan, and A. Qin "Performance evaluation of multi agent genetic algorithm," *Natural Computing*, vol.5, no.1, pp. 83-96, 2006.
- [45] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. IEEE International Swarm Intelligence Symposium*, 2005, pp. 68-75.

[46] P.H. Chen and H.C. Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Trans. Power Systems*, vol. 10, no. 1, pp. 117-124, Feb. 1995.

[47] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. Power Systems*, vol. 15, no. 4, pp. 1232-1239, Nov. 2000.

```

begin
   $t \rightarrow 0$            // iteration number
  Initialize  $X(t)$     //  $X(t)$ : Swarm for iteration  $t$ 
  Evaluate  $f(X(t))$  //  $f(\cdot)$ : fitness function
  while (not termination condition) do
    begin
       $t \rightarrow t+1$ 
      // Process of SPSO //
      Update velocity  $\mathbf{v}(t)$  and position of each particle  $\mathbf{x}(t)$  based on (2) – (5) respectively
      if  $v(t) > v_{max}$ 
         $v(t) = v_{max}$ 
      end
      if  $v(t) < -v_{max}$ 
         $v(t) = -v_{max}$ 
      end
      // End of the process of SPSO //
      Reproduce a new  $X(t)$ 
      Evaluate  $f(X(t))$ 
    end
  end

```

(a)

```

begin
   $t \rightarrow 0$            // iteration number
  Initialize  $X(t)$     //  $X(t)$ : Swarm for iteration  $t$ 
  Evaluate  $f(X(t))$  //  $f(\cdot)$ : fitness function
  while (not termination condition) do
    begin
       $t \rightarrow t+1$ 
      Perform the process of SPSO (shown in Fig. 1a)
      Perform mutation operation (6) with  $p_m$  //  $p_m$  : probability of mutation
      Reproduce a new  $X(t)$ 
      Evaluate  $f(X(t))$ 
    end
  end

```

(b)

Fig. 1. Pseudo code for (a) SPSO and (b) HPSOM (Hybrid PSO with mutation operation)

```

begin
   $t \rightarrow 0$  // iteration number
  Initialize  $X(t)$  //  $X(t)$ : Swarm for iteration  $t$ 
  Evaluate  $f(X(t))$  //  $f(\cdot)$ : fitness function
  while (not termination condition) do
    begin
       $t \rightarrow t+1$ 
      Perform the process of SPSO (shown in Fig. 1)
      // Process of Wavelet mutation operation
      Perform mutation operation with  $p_m$ 
      Update  $\bar{x}_j^p(t)$  based on (15)-(18) and (20)
      // End of the process of wavelet mutation operation
      Reproduce a new  $X(t)$ 
      Evaluate  $f(X(t))$ 
    end
  end

```

Fig. 2. Pseudo code for HPSOWM.

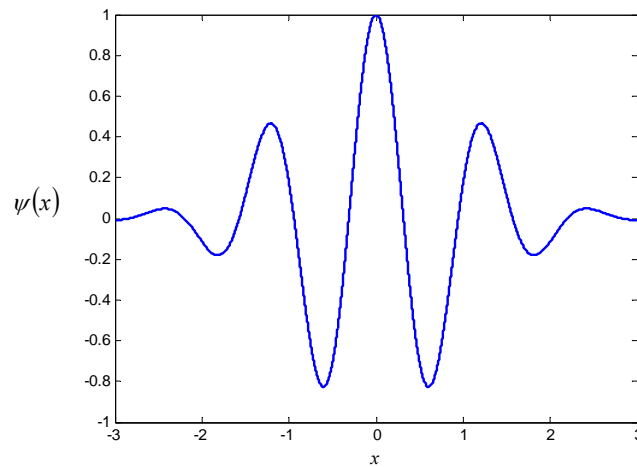


Fig. 3. Morlet wavelet.

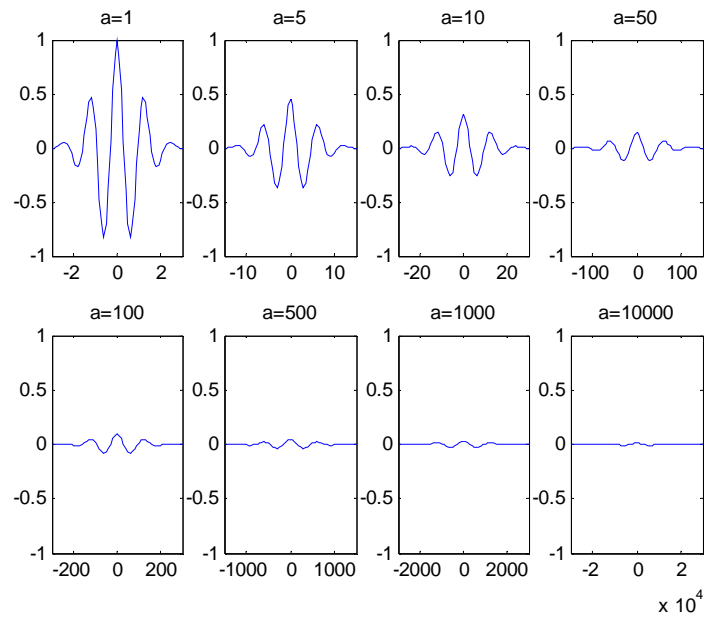


Fig. 4. Morlet wavelet dilated by different values of the parameter a (x-axis: x , y-axis: $\psi_{a,0}(x)$.)

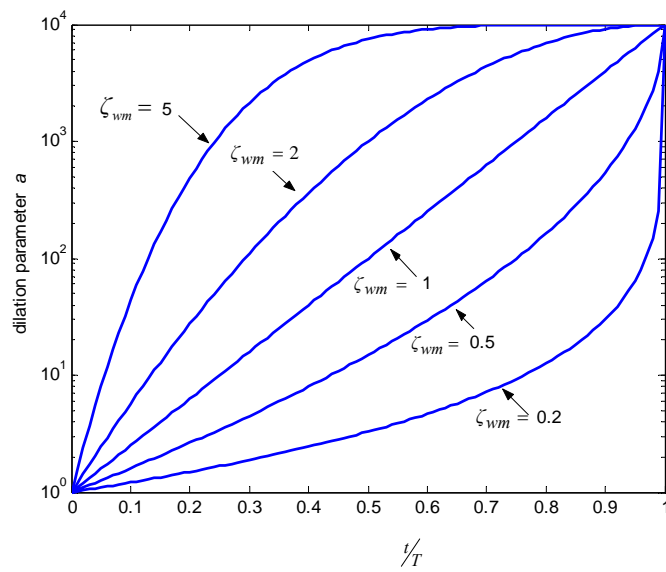


Fig. 5. Effect of the shape parameter ζ_{wm} to a with respect to t/T .

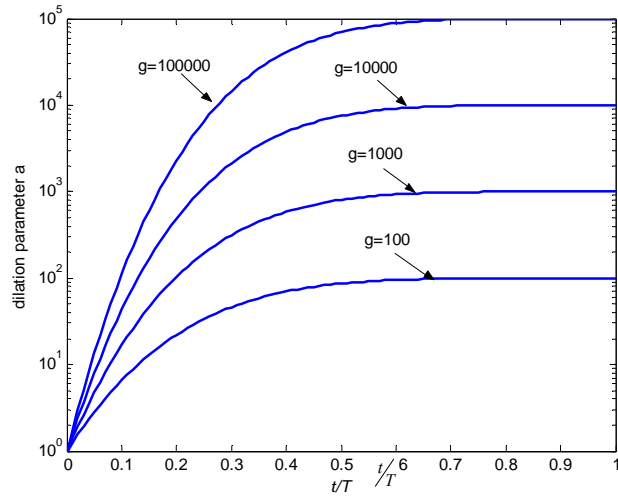


Fig. 6. Effect of the parameter g to a with respect to t/T .

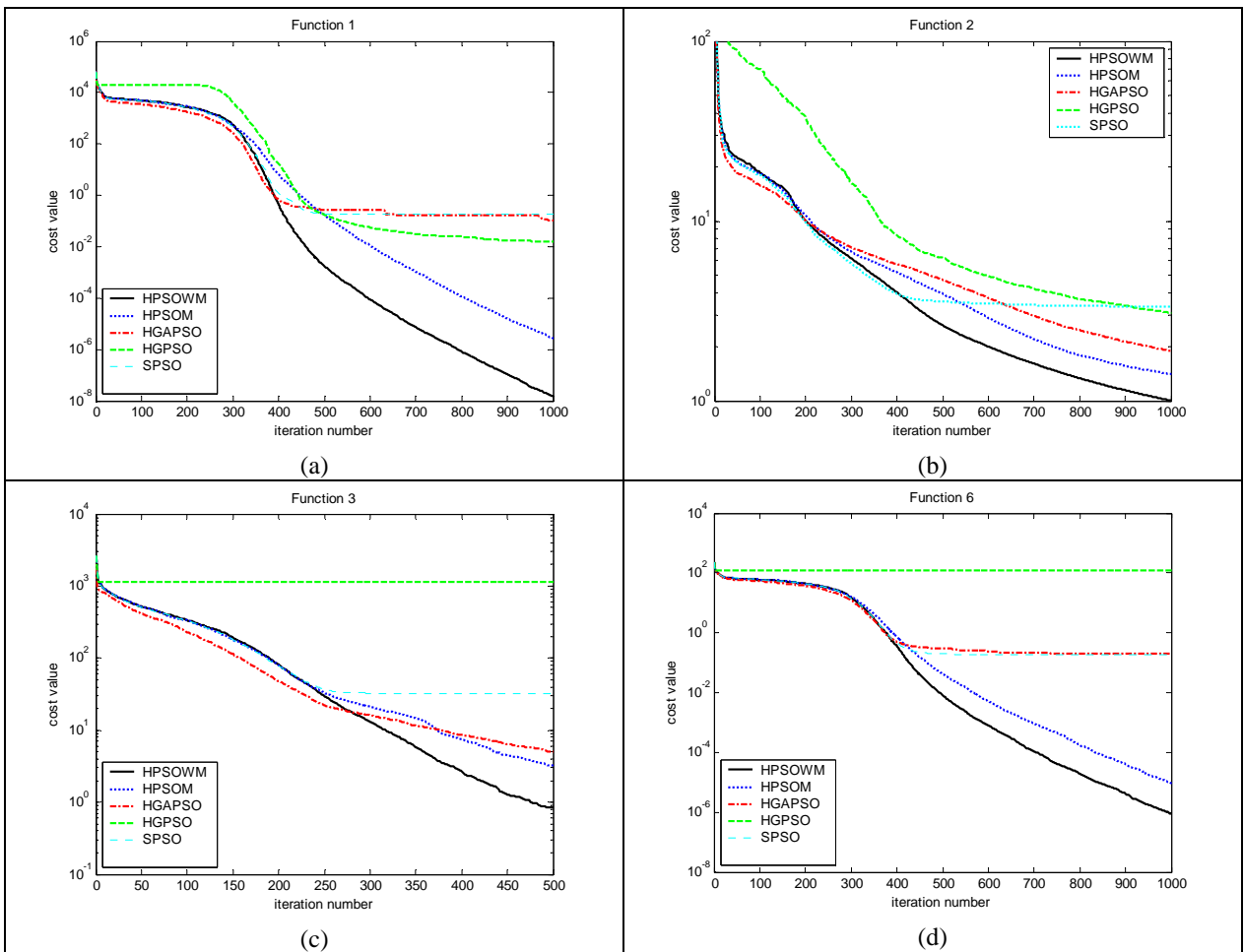


Fig. 7. Comparisons between different PSO methods for unimodal functions.

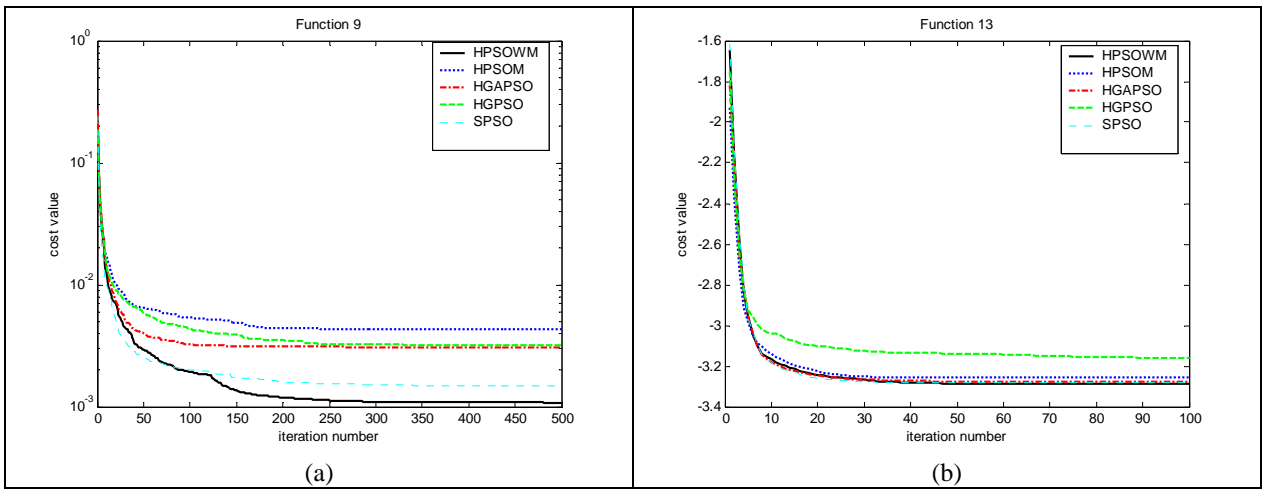


Fig. 8. Comparisons between different PSO methods for multimodal functions with a few local minima.

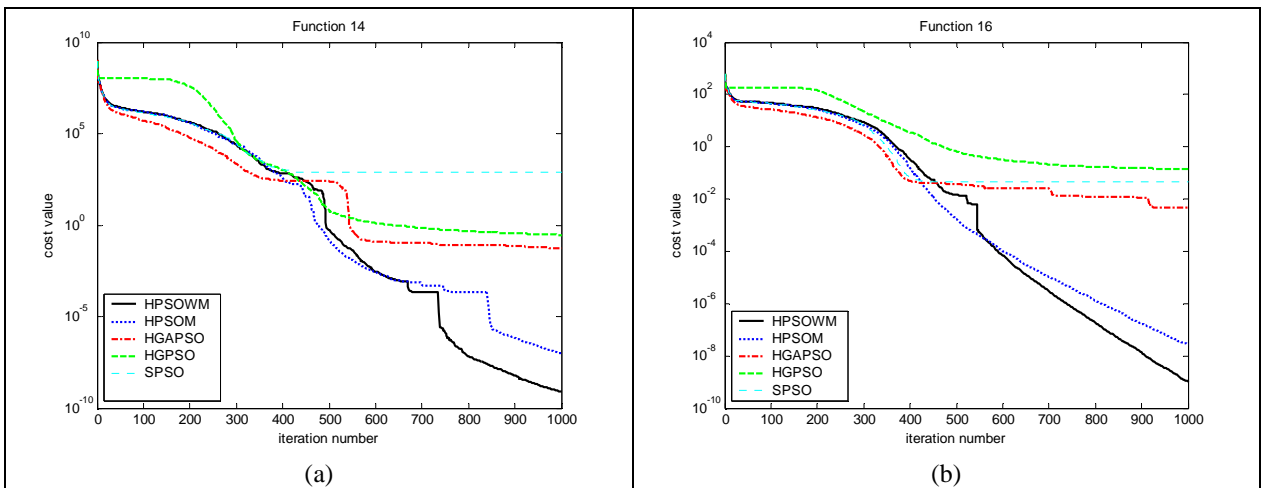


Fig. 9. Comparisons between different PSO methods for multimodal functions with many local minima.

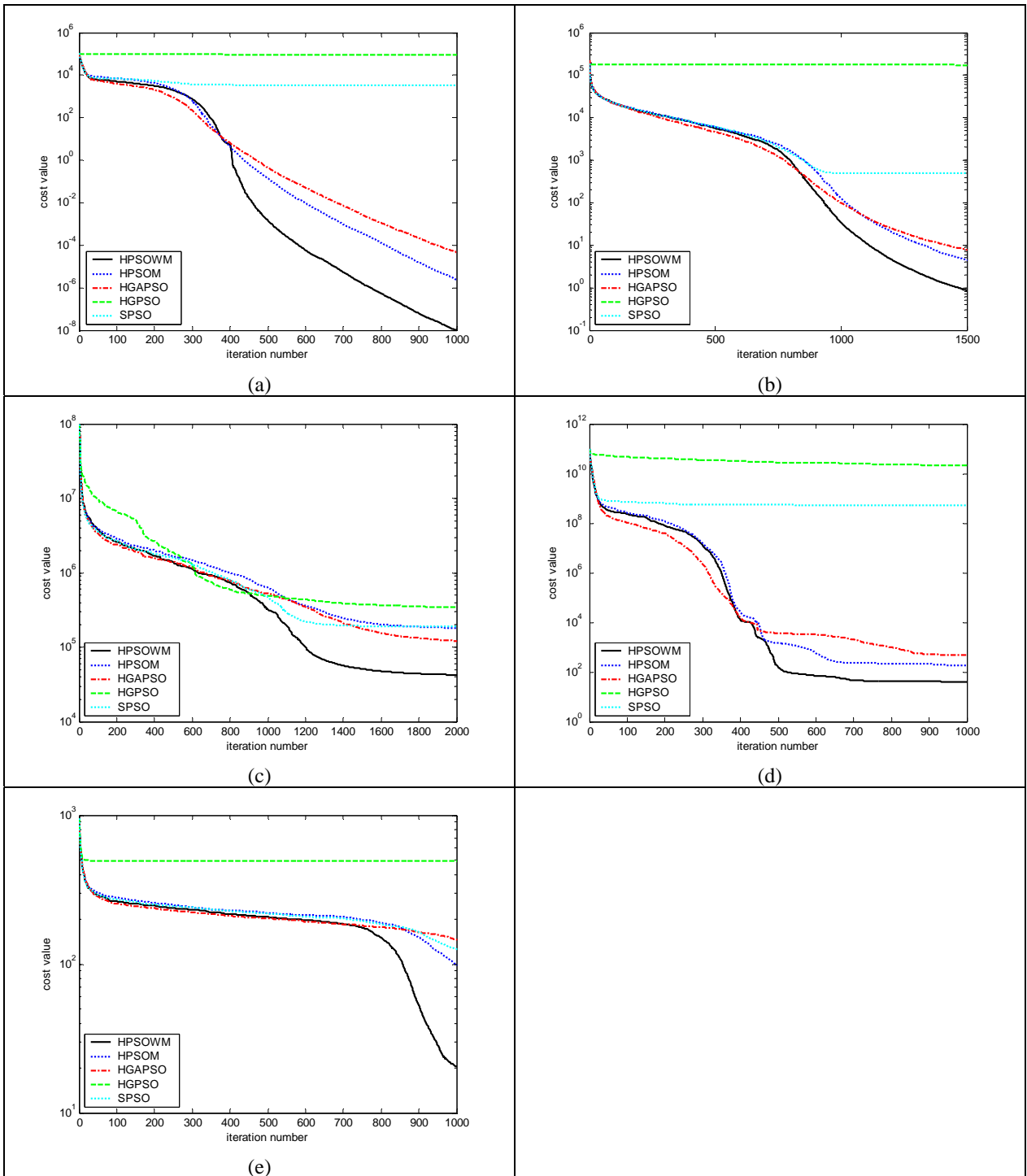


Fig. 10. Comparisons between different PSO methods for benchmark functions with shift.

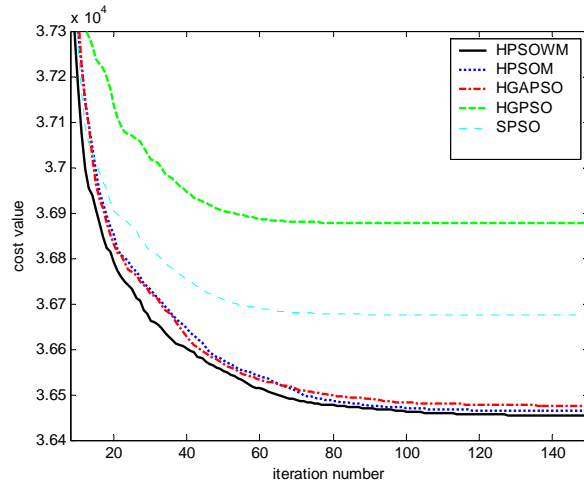


Fig. 11. Comparisons between different PSO methods for MC-TSCOPF.

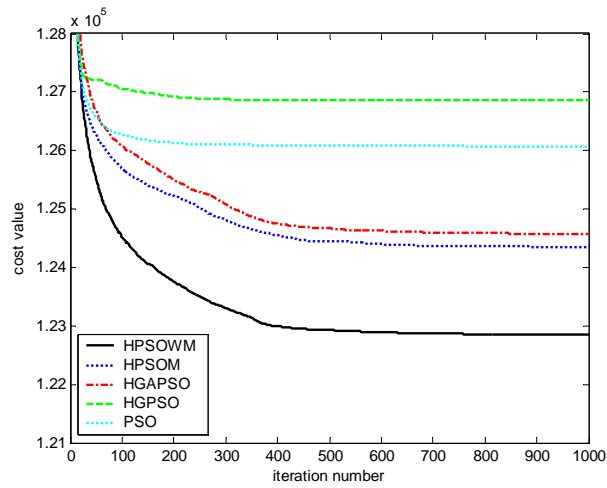


Fig. 12. Comparisons between different PSO methods for ELD-VPL.

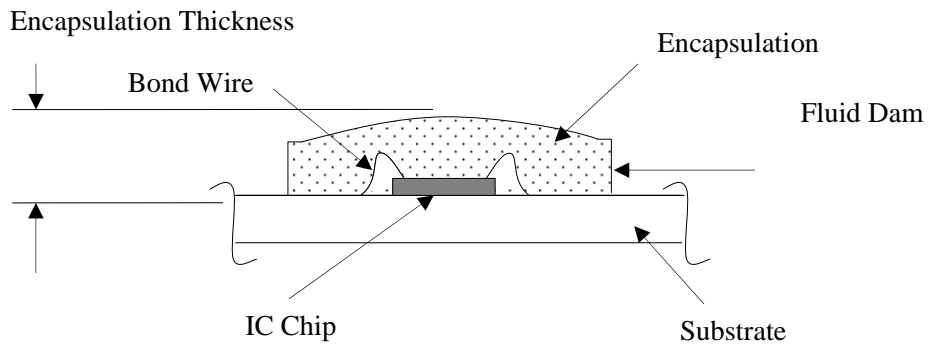


Fig. 13. Encapsulation of microchip.

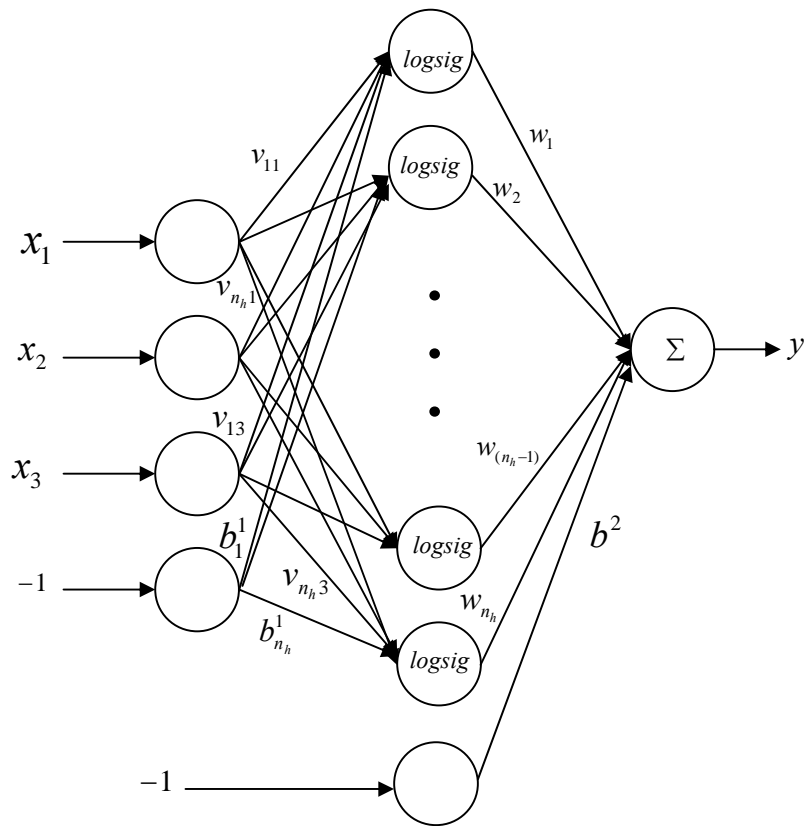


Fig. 14. Structure of the feed-forward neural network.

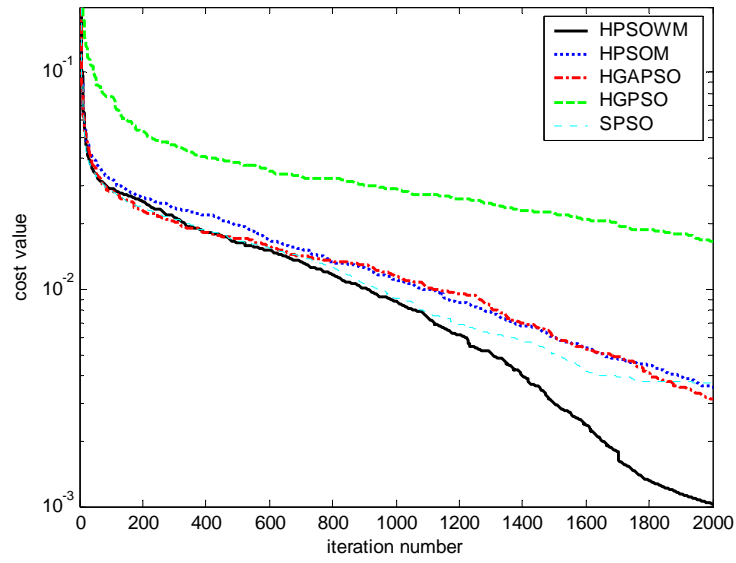


Fig. 15. Comparisons between different PSO methods for MFD-EP (encapsulation weight).

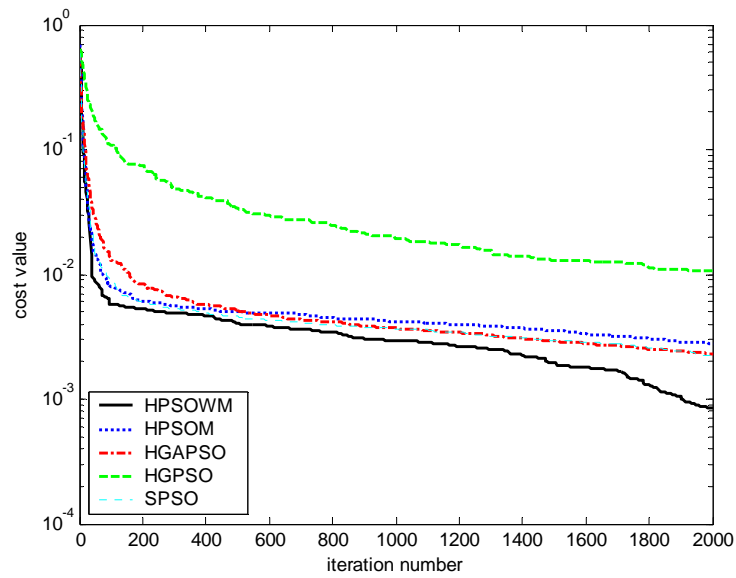


Fig. 16. Comparisons between different PSO methods for MFD-EP (encapsulation thickness).

TABLE I BENCHMARK TEST FUNCTIONS

| Test function | Domain range | Optimal point |
|---|--------------------------------|--|
| $f_1(\mathbf{x}) = \sum_{i=1}^{30} x_i^2$ | $-100 \leq x_i \leq 100$ | $f_1(\mathbf{0}) = 0$ |
| $f_2(\mathbf{x}) = \sum_{i=1}^9 \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ | $-2.048 \leq x_i \leq 2.048$ | $f_2(\mathbf{1}) = 0$ |
| $f_3(\mathbf{x}) = \sum_{i=1}^{100} (x_i + 0.5)^2$ | $-10 \leq x_i \leq 10$ | $f_3(\mathbf{0}) = 0$ |
| $f_4(\mathbf{x}) = \sum_{i=1}^{10} ix_i^4 + \text{random}[0,1)$ | $-2.56 \leq x_i \leq 2.56$ | $f_4(\mathbf{0}) = 0$ |
| $f_5(\mathbf{x}) = \max_i \{ x_i , 1 \leq i \leq 30 \}$ | $-100 \leq x_i \leq 100$ | $f_5(\mathbf{0}) = 0$ |
| $f_6(\mathbf{x}) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $ | $-10 \leq x_i \leq 10$ | $f_6(\mathbf{0}) = 0$ |
| $f_7(\mathbf{x}) = -\cos(x_1) \cdot \cos(x_2) \cdot \exp\left(-\left((x_1 - \pi)^2 + (x_2 - \pi)^2\right)\right)$ | $-300 \leq x_1, x_2 \leq 300$ | $f_7([\pi, \pi]) = -1$ |
| $f_8(\mathbf{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$, | $-65.536 \leq x_i \leq 65.536$ | $f_8([-32, -32]) \approx 1$ |
| $f_9(\mathbf{x}) = \sum_{i=1}^9 \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$ | $-5 \leq x_i \leq 5$ | $f_9(0.1928, 0.1928, 0.1231, 0.1358) \approx 0.0003075$ |
| $f_{10}(\mathbf{x}) = -\frac{\sin(x_1)\sin(x_2)}{x_1 x_2}$ | $-10 \leq x_1, x_2 \leq 10$ | $\lim_{x \rightarrow [0,0]} f_{10}(\mathbf{x}) = -1$ |
| $f_{11}(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$ | $-5 \leq x_1, x_2 \leq 5$ | $f_{11}([0.08983, -0.7126]) =$ $f_{11}([-0.08983, 0.7126]) \approx -1.0316$ |
| $f_{12}(\mathbf{x}) = -\sum_{i=1}^{30} c_i \exp\left[-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right]$ | $0 \leq x_i \leq 1$ | $f_{12}(0.114, 0.556, 0.852)$ ≈ -3.8628 |
| $f_{13}(\mathbf{x}) = -\sum_{i=1}^4 c_i \exp\left[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right]$ | $0 \leq x_i \leq 1$ | $f_{13}([0.201, 0.15, 0.477,$ $0.275, 0.311, 0.627]) \approx -3.32$ |
| $f_{14}(\mathbf{x}) = 0.1 \left\{ \frac{\sin^2(\pi x_1)}{\sum_{i=1}^{29} (x_i - 1)^2 \cdot [1 + \sin^2(3\pi x_{i+1})]} + \sum_{i=1}^{30} u(x_i, 5, 100, 4) \right.$ $\left. + (x_{30} - 1)^2 [1 + \sin^2(2\pi x_{30})] \right\}$ | $-50 \leq x_i \leq 50$ | $f_{14}(\mathbf{1}) = 0$ |
| $f_{15}(\mathbf{x}) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | $-50 \leq x_i \leq 50$ | $f_{15}(\mathbf{0}) = 0$ |
| $f_{16}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$ | $-600 \leq x_i \leq 600$ | $f_{16}(\mathbf{0}) = 0$ |
| $f_{17}(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i\right) + 20 + e$ | $-32 \leq x_i \leq 32$ | $f_{17}(\mathbf{0}) = 0$ |
| $f_{18}(\mathbf{x}) = -\sum_{i=1}^{10} (x_i \sin(\sqrt{ x_i }))$ | $-500 \leq x_i \leq 500$ | $f_{18}([420.9687, \dots, 420.9687])$ $= -10 \times 418.9829 = -4189.829$ |

TABLE II COMPARISON BETWEEN DIFFERENT PSO METHODS FOR BENCHMARK TEST FUNCTIONS
(CATEGORY 1). ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|---|---------|---|-----------------|-----------------------|----------------------|-----------------------|
| f_1 ($\times 10^{-5}$) $T=1000$ | Mean | <u>0.0015</u> | 0.2732 | 10834.8084 | 1586.8234 | 18217.2796 |
| | Best | <u>0.0004</u> | 0.0724 | 121.1258 | 59.7069 | 15.8949 |
| | Std Dev | <u>0.0010</u> | 0.1493 | 18020.5712 | 1368.0663 | 26642.2440 |
| | Rank | <u>1</u> | 2 | 4 | 3 | 5 |
| f_2 ($\times 10^0$) $T=1000$ | Mean | <u>1.0030</u> | 1.4095 | 1.8935 | 3.0879 | 3.3426 |
| | Best | <u>0.5441</u> | 0.8952 | 1.1761 | <u>0.1428</u> | 2.1794 |
| | Std Dev | <u>0.2155</u> | 0.2653 | 0.3300 | 2.8497 | 0.4905 |
| | Rank | <u>1</u> | 2 | 3 | 4 | 5 |
| f_3 ($\times 10^0$) $T=500$ | Mean | <u>0.8400</u> | 3.2000 | 5.0200 | 1140.22 | 32.0800 |
| | Best | <u>0.0000</u> | 0.0000 | 0.0000 | 933.000 | 13.0000 |
| | Std Dev | <u>0.9116</u> | 2.6186 | 11.7603 | 87.6000 | 24.1313 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_4 ($\times 10^{-2}$) $T=1000$ | Mean | <u>0.5126</u> | 0.7845 | 0.5496 | 44.9282 | 0.7065 |
| | Best | <u>0.1783</u> | 0.2405 | 0.2934 | 16.0263 | 0.2758 |
| | Std Dev | 0.1687 | 0.2162 | <u>0.1341</u> | 17.2645 | 0.2295 |
| | Rank | <u>1</u> | 4 | 2 | 5 | 3 |
| f_5 ($\times 10^0$) $T=1000$ | Mean | <u>0.2587</u> | 0.2684 | 1.9689 | 9.1558 | 2.6777 |
| | Best | 0.1057 | <u>0.0504</u> | 1.0284 | 0.5074 | 1.5338 |
| | Std Dev | <u>0.1070</u> | 0.1129 | 0.4418 | 14.6778 | 0.6149 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_6 ($\times 10^{-5}$) $T=1000$ | Mean | <u>0.0872</u> | 0.9139 | 19560.3257 | 121.66×10^5 | 18189.8149 |
| | Best | <u>0.0224</u> | 0.2745 | 2067.5365 | 91.52×10^5 | 775.5302 |
| | Std Dev | <u>0.0440</u> | 0.5361 | 16787.6184 | 11.21×10^5 | 18356.8947 |
| | Rank | <u>1</u> | 2 | 4 | 5 | 3 |
| f_7 ($\times 10^0$) $T=100$ | Mean | <u>-1.0000</u> | -0.9949 | -0.9999 | -0.3060 | -0.9999 |
| | Best | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| | Std Dev | <u>1.00×10^{-9}</u> | 0.0364 | 1.36×10^{-5} | 0.4609 | 4.02×10^{-5} |
| | Rank | <u>1</u> | 4 | 2 | 5 | 3 |
| Overall Ranking (Average ranking number) | | 1 (1.0) | 2 (2.57) | 3 (3.0) | 5 (4.57) | 4 (3.86) |

TABLE III COMPARISON BETWEEN DIFFERENT PSO METHODS FOR BENCHMARK TEST FUNCTIONS

(CATEGORY 2). ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|--|---------|-------------------|-------------------|-------------------|-------------------|-------------------|
| f_8 ($\times 10^0$) $T=100$ | Mean | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9980 |
| | Best | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9980 |
| | Std Dev | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | Rank | <u>1</u> | <u>1</u> | <u>1</u> | <u>1</u> | <u>1</u> |
| f_9 ($\times 10^{-3}$) $T=500$ | Mean | <u>1.0829</u> | 4.3140 | 3.0503 | 3.1956 | 1.4730 |
| | Best | 0.5844 | 0.6486 | 0.4209 | <u>0.3091</u> | 0.4041 |
| | Std Dev | <u>2.1739</u> | 9.8862 | 6.4610 | 9.8017 | 3.9022 |
| | Rank | <u>1</u> | 5 | 3 | 4 | 2 |
| f_{10} ($\times 10^{-5}$) $T=100$ | Mean | <u>245.5877</u> | 245.5961 | 245.5878 | 249.7721 | 245.5879 |
| | Best | 245.5858 | 245.5858 | 245.5858 | 245.5858 | 245.5858 |
| | Std Dev | <u>0.0039</u> | 0.0179 | 0.0060 | 25.2583 | 0.0070 |
| | Rank | <u>1</u> | 4 | 2 | 5 | 3 |
| f_{11} ($\times 10^{-4}$) $T=100$ | Mean | -10316.284 | -10316.266 | <u>-10316.285</u> | -10316.280 | <u>-10316.285</u> |
| | Best | -10316.285 | -10316.284 | -10316.285 | -10316.285 | -10316.285 |
| | Std Dev | 0.00008 | 0.02982 | <u>0.00000</u> | 0.01093 | <u>0.00000</u> |
| | Rank | 3 | 5 | <u>1</u> | 4 | <u>1</u> |
| f_{12} ($\times 10^{-4}$) $T=100$ | Mean | <u>-38627.822</u> | -38627.819 | <u>-38627.822</u> | -38609.629 | <u>-38627.822</u> |
| | Best | <u>-38627.822</u> | -38627.821 | <u>-38627.822</u> | <u>-38627.822</u> | <u>-38627.822</u> |
| | Std Dev | <u>0.00000</u> | 0.00259 | <u>0.00000</u> | 17.02938 | <u>0.00000</u> |
| | Rank | <u>1</u> | 4 | <u>1</u> | 5 | <u>1</u> |
| f_{13} ($\times 10^{-4}$) $T=100$ | Mean | <u>-32934.608</u> | -32577.928 | -32744.379 | -31597.339 | -32818.652 |
| | Best | <u>-33219.952</u> | <u>-33219.952</u> | <u>-33219.952</u> | <u>-33219.952</u> | <u>-33219.952</u> |
| | Std Dev | <u>512.9276</u> | 598.5759 | 588.3683 | 1563.1222 | 577.3916 |
| | Rank | <u>1</u> | 5 | 3 | 4 | 2 |
| Overall Ranking (Average ranking number) | | 1 (1.33) | 5 (4.00) | 3 (1.83) | 4 (3.83) | 2 (1.67) |

TABLE IV COMPARISON BETWEEN DIFFERENT PSO METHODS FOR BENCHMARK TEST FUNCTIONS

(CATEGORY 3). ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|--|---------|-----------------|----------------|----------------|----------------|----------------|
| f_{14} ($\times 10^{-5}$) $T=1000$ | Mean | <u>0.0001</u> | 0.0097 | 5717.5165 | 29541.424 | 73998389.7 |
| | Best | <u>0.0000</u> | 0.0033 | 70.3882 | 1448.468 | 242.9188 |
| | Std Dev | <u>0.0001</u> | 0.0089 | 6526.1476 | 18996.139 | 224284238.1 |
| | Rank | <u>1</u> | 2 | 3 | 4 | 5 |
| f_{15} ($\times 10^0$) $T=500$ | Mean | 10.2854 | 16.5169 | 19.3510 | 253.4083 | 20.7363 |
| | Best | 3.9828 | 5.9698 | 10.1299 | 174.0025 | 12.6420 |
| | Std Dev | 3.3522 | 5.8398 | 5.5421 | 26.1654 | 5.3895 |
| | Rank | 1 | 2 | 3 | 4 | 5 |
| f_{16} ($\times 10^{-5}$) $T=1000$ | Mean | <u>0.0001</u> | 0.0028 | 462.4819 | 13500.614 | 4404.184 |
| | Best | <u>0.0000</u> | 0.0004 | 15.6250 | 495.7651 | 0.8262 |
| | Std Dev | <u>0.0001</u> | 0.0032 | 482.2762 | 12956.985 | 10251.732 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_{17} ($\times 10^{-5}$) $T=1500$ | Mean | <u>1.0607</u> | 13.8681 | 22884.375 | 1784083.03 | 49893.949 |
| | Best | <u>0.4919</u> | 6.6542 | 1022.202 | 1637571.42 | 153.683 |
| | Std Dev | <u>0.3120</u> | 3.8260 | 39045.653 | 62021.418 | 62724.315 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_{18} ($\times 10^0$) $T=500$ | Mean | <u>-3928.83</u> | -3561.22 | -3495.37 | -3136.51 | -3586.37 |
| | Best | <u>-4189.83</u> | -4071.39 | -3854.25 | -3951.43 | -4071.39 |
| | Std Dev | <u>176.65</u> | 231.14 | 239.78 | 368.55 | 240.56 |
| | Rank | <u>1</u> | 3 | 4 | 5 | 2 |
| Overall Ranking (Average ranking number) | | 1 (1) | 2 (2.2) | 3 (3.2) | 5 (4.6) | 4 (4.0) |

TABLE V T-VALUE BETWEEN HPSOWM AND THE OTHER PSO METHODS

| Function s | t-value between HPSOWM and HPSOM | t-value between HPSOWM and HGAPSO | t-value between HPSOWM and HGPSO | t-value between HPSOWM and SPSO |
|------------|----------------------------------|-----------------------------------|----------------------------------|---------------------------------|
| f_1 | 12.87 | 4.25 | 8.20 | 4.84 |
| f_2 | 8.41 | 15.98 | 5.16 | 30.88 |
| f_3 | 6.02 | 2.51 | 91.97 | 9.15 |
| f_4 | 7.01 | 1.21 | 18.19 | 4.81 |
| f_5 | 0.44 | 26.60 | 4.29 | 27.41 |
| f_6 | 10.87 | 8.24 | 77.78 | 7.01 |
| f_7 | 0.99 | 51.99 | 10.65 | 17.59 |
| f_8 | N/A | N/A | N/A | N/A |
| f_9 | 2.26 | 2.04 | 1.49 | 0.62 |
| f_{10} | N/A | N/A | 1.17 | N/A |
| f_{11} | N/A | N/A | N/A | N/A |
| f_{12} | N/A | N/A | 7.55 | N/A |
| f_{13} | 3.20 | 1.72 | 5.75 | 1.06 |
| f_{14} | 7.63 | 6.19 | 11.00 | 2.33 |
| f_{15} | 6.54 | 9.90 | 65.17 | 11.64 |
| f_{16} | 5.96 | 6.78 | 7.37 | 3.04 |
| f_{17} | 23.59 | 4.14 | 203.40 | 5.62 |
| f_{18} | 8.94 | 10.29 | 13.71 | 8.11 |

TABLE VI EQUATION OF BENCHMARK TEST FUNCTIONS WITH SHIFT

| Test functions with shift | Domain range |
|---|--------------------------|
| $f_{1-shift}(\mathbf{x}) = \sum_{i=1}^{30} x_i^2$ | $-100 \leq x_i \leq 100$ |
| $f_{2-shift}(\mathbf{x}) = \sum_{k=1}^{30} \sum_{i=1}^k x_i^2$ | $-100 \leq x_i \leq 100$ |
| $f_{3-shift}(\mathbf{x}) = \sum_{i=1}^{10} \left(A \binom{i-1}{9} (x_i^2) \right)$, where $A=1 \times 10^6$ | $-100 \leq x_i \leq 100$ |
| $f_{4-shift}(\mathbf{x}) = \sum_{i=1}^{30} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ | $-100 \leq x_i \leq 100$ |
| $f_{5-shift}(\mathbf{x}) = \sum_{i=1}^{30} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$ | $-5 \leq x_i \leq 5$ |

TABLE VII COMPARISON BETWEEN DIFFERENT PSO METHODS FOR BENCHMARK TEST FUNCTIONS

WITH SHIFT. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|--|------------------------------|----------------|----------------|----------------|---------------------|---------------------|
| f_1 -shift | Mean ($\times 10^{-5}$) | <u>0.0009</u> | 0.2461 | 4.7214 | 89099×10^5 | 3400×10^5 |
| | Best ($\times 10^{-5}$) | <u>0.0003</u> | 0.0644 | 1.7769 | 21440×10^5 | 103.3×10^5 |
| | Std Dev ($\times 10^{-5}$) | <u>0.0008</u> | 0.1572 | 2.1647 | 23878×10^5 | 2958×10^5 |
| | t -value | N/A | 11.03 | 15.42 | 26.39 | 8.13 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_2 -shift | Mean | <u>0.8682</u> | 4.5085 | 7.9399 | 174402.9 | 495.75 |
| | Best | <u>0.1281</u> | 0.9981 | 2.9045 | 84910.9 | 57.91 |
| | Std Dev | <u>0.5374</u> | 2.4111 | 3.5604 | 55237.5 | 1142.94 |
| | t -value | N/A | 10.42 | 13.89 | 22.33 | 3.06 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_3 -shift | Mean ($\times 10^5$) | <u>1.0099</u> | 1.8167 | 1.2154 | 3.4240 | 1.9047 |
| | Best ($\times 10^5$) | 0.0814 | <u>0.0805</u> | 0.1793 | 0.8244 | 0.3108 |
| | Std Dev ($\times 10^5$) | <u>0.7752</u> | 1.4424 | 0.8864 | 1.5210 | 1.4428 |
| | t -value | N/A | 3.48 | 1.23 | 10.00 | 3.86 |
| | Rank | <u>1</u> | 3 | 2 | 5 | 4 |
| f_4 -shift | Mean | <u>40.9443</u> | 185.1716 | 467.6383 | 21.38×10^9 | 54.32×10^7 |
| | Best | <u>6.5265</u> | 12.7814 | 20.2429 | 9.42×10^6 | 2007.13 |
| | Std Dev | <u>28.9749</u> | 295.5604 | 1087.14 | 20.19×10^9 | 60.77×10^7 |
| | t -value | N/A | 3.43 | 2.77 | 7.49 | 6.32 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| f_5 -shift | Mean | <u>20.5177</u> | 98.3374 | 145.7935 | 493.3936 | 125.3497 |
| | Best | <u>15.9857</u> | 17.8212 | 23.4791 | 408.0382 | 34.7785 |
| | Std Dev | <u>2.8576</u> | 59.2000 | 47.7815 | 52.0081 | 45.3456 |
| | t -value | N/A | 9.28 | 18.51 | 64.20 | 16.31 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |
| Overall Ranking (Average ranking number) | | 1 (1.0) | 2 (2.2) | 3 (2.8) | 5 (5.0) | 4 (4.0) |

TABLE VIII SENSITIVITY OF THE SHAPE PARAMTER FOR WAVELET MUTATION ζ_{wm}

| Functions | $\zeta_{wm} = 0.2$ | $\zeta_{wm} = 0.5$ | $\zeta_{wm} = 1.0$ | $\zeta_{wm} = 2.0$ | $\zeta_{wm} = 5.0$ |
|---------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| $f_1 (\times 10^{-5})$ | 0.9124 | 0.0752 | 0.0098 | 0.0027 | <u>0.0015</u> |
| $f_2 (\times 10^0)$ | 4.7345 | 4.2675 | 3.6765 | 3.0992 | <u>1.0728</u> |
| $f_3 (\times 10^0)$ | 0.8400 | 5.6400 | 17.22 | 30.30 | 31.88 |
| $f_4 (\times 10^{-3})$ | 5.2763 | 5.5626 | 5.5987 | 5.7988 | <u>5.1260</u> |
| $f_5 (\times 10^0)$ | 0.3560 | 0.2768 | <u>0.2587</u> | 0.7753 | 1.1901 |
| $f_6 (\times 10^{-5})$ | 2.3139 | 0.8078 | 0.3044 | 0.1290 | <u>0.0872</u> |
| $f_7 (\times 10^0)$ | -0.9999 | -1.0000 | -1.0000 | -0.9600 | -0.9949 |
| $f_8 (\times 10^0)$ | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9980 |
| $f_9 (\times 10^{-3})$ | <u>1.0829</u> | 1.1684 | 2.1727 | 1.4457 | 2.3767 |
| $f_{10} (\times 10^{-3})$ | 2.4562 | 2.4559 | 2.4559 | <u>2.4559</u> | 2.4559 |
| $f_{11} (\times 10^{-4})$ | -10316.1667 | -10316.2804 | -10316.2839 | -10316.2844 | <u>-10316.2845</u> |
| $f_{12} (\times 10^{-4})$ | -38627.8173 | -38627.8211 | -38627.8215 | -38627.8215 | <u>-38627.8215</u> |
| $f_{13} (\times 10^0)$ | -3.2863 | -3.2863 | -3.2744 | -3.2720 | <u>-3.2935</u> |
| $f_{14} (\times 10^{-5})$ | 0.0397 | 0.0025 | 0.0003 | <u>0.0001</u> | 0.0001 |
| $f_{15} (\times 10^0)$ | <u>10.2854</u> | 16.8148 | 17.1133 | 18.1878 | 18.0486 |
| $f_{16} (\times 10^{-5})$ | 0.0078 | 0.0009 | <u>0.0001</u> | 0.1166 | 0.1257 |
| $f_{17} (\times 10^{-4})$ | 2.3264 | 0.6486 | 0.2504 | 0.1347 | <u>0.1061</u> |
| $f_{18} (\times 10^0)$ | <u>-3928.83</u> | -3578.69 | -3574.68 | -3582.24 | -3572.67 |

TABLE IX SENSITIVITY OF THE PARAMTER G IN WAVELET MUTATION OPERATION

| Functions | $g = 100$ | $g = 1000$ | $g = 10000$ | $g = 100000$ |
|---------------------------|-------------|---------------|--------------------|---------------|
| $f_1 (\times 10^{-5})$ | 0.0019 | 0.0017 | <u>0.0015</u> | 0.0016 |
| $f_2 (\times 10^0)$ | 1.1677 | <u>0.9651</u> | 1.0728 | 0.9844 |
| $f_3 (\times 10^0)$ | 35.70 | 34.30 | <u>31.88</u> | 35.00 |
| $f_4 (\times 10^{-3})$ | 8.4967 | 7.5472 | <u>5.1260</u> | 8.4817 |
| $f_5 (\times 10^0)$ | 1.3872 | 1.7286 | <u>1.1901</u> | 1.2979 |
| $f_6 (\times 10^{-5})$ | 0.1056 | 0.1270 | <u>0.0872</u> | 0.0983 |
| $f_7 (\times 10^0)$ | -0.9999 | -0.9999 | -0.9949 | -0.9999 |
| $f_8 (\times 10^0)$ | 0.9980 | 0.9980 | 0.9980 | 0.9980 |
| $f_9 (\times 10^{-3})$ | 5.1478 | 2.6208 | 2.3767 | <u>1.3275</u> |
| $f_{10} (\times 10^{-3})$ | 2.4559 | 2.4559 | 2.4559 | 2.4558 |
| $f_{11} (\times 10^{-4})$ | -10316.2844 | -10316.2844 | <u>-10316.2845</u> | -10316.2844 |
| $f_{12} (\times 10^{-4})$ | -38627.8215 | -38627.8215 | -38627.8215 | -38627.8215 |
| $f_{13} (\times 10^0)$ | -3.2625 | -3.2507 | <u>-3.2935</u> | -3.2744 |
| $f_{14} (\times 10^{-6})$ | 0.0015 | 0.0005 | 0.0010 | <u>0.0003</u> |
| $f_{15} (\times 10^0)$ | 24.2803 | 23.3822 | <u>18.0486</u> | 26.8642 |
| $f_{16} (\times 10^{-5})$ | 0.1332 | 0.1301 | 0.1257 | <u>0.1183</u> |
| $f_{17} (\times 10^{-4})$ | 0.1130 | 0.1086 | <u>0.1061</u> | 0.9867 |
| $f_{18} (\times 10^0)$ | -3532.81 | -3479.04 | <u>-3572.67</u> | -3589.74 |

TABLE X COMPARISON BETWEEN DIFFERENT PSO METHODS FOR MC-TSCOPF. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|---------|--------------|-----------------|----------|----------|----------|----------|
| $T=150$ | Mean | <u>36452.97</u> | 36464.20 | 36474.90 | 36877.66 | 36675.48 |
| | Best | <u>36435.61</u> | 36439.66 | 36442.08 | 36512.71 | 36459.85 |
| | Std Dev | <u>8.91</u> | 42.81 | 33.60 | 245.17 | 241.33 |
| | t -value | N/A | 4.46 | 12.24 | 6.52 | 8.73 |
| | Run Time (s) | 4814.4 | 4885.47 | 4878.11 | 9766.48 | 4878.36 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |

TABLE XI: COMPARISON BETWEEN DIFFERENT PSO METHODS FOR 40-generator system. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|----------|--------------|-----------------|-----------|----------|-----------|----------|
| $T=1000$ | Mean | <u>122844.4</u> | 124350.87 | 124575.7 | 126855.7 | 126074.4 |
| | Best | <u>121915.3</u> | 122112.4 | 122780.0 | 124797.13 | 124350.4 |
| | Std Dev | <u>497.44</u> | 978.75 | 906.04 | 1160.91 | 1153.11 |
| | t -value | N/A | 9.70 | 11.84 | 22.46 | 18.19 |
| | Run Time (s) | 25.39 | 23.91 | 24.22 | 48.47 | 23.92 |
| | Rank | <u>1</u> | 2 | 3 | 5 | 4 |

TABLE XII COMPARISON BETWEEN DIFFERENT PSO METHODS FOR MFD-EP (TRAINING): (A) ENCAPSULATION WEIGHT, (B) ENCAPSULATION THICKNESS. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

(a)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|----------------|------------------------------|---------------|--------------|--------|--------|--------|
| <i>T</i> =2000 | Mean ($\times 10^{-2}$) | <u>0.1026</u> | 0.3521 | 0.3142 | 1.6621 | 0.3724 |
| | Best ($\times 10^{-2}$) | <u>0.0426</u> | 0.0946 | 0.0994 | 0.1462 | 0.0777 |
| | Std Dev ($\times 10^{-2}$) | <u>0.0665</u> | 0.2000 | 0.2152 | 0.8827 | 0.3556 |
| | <i>t</i> -value | N/A | 8.37 | 6.64 | 12.46 | 5.27 |
| | Run Time (s) | 99.43 | <u>96.19</u> | 97.02 | 191.36 | 96.20 |
| | Rank | <u>1</u> | 3 | 2 | 5 | 4 |

(b)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|----------------|------------------------------|---------------|--------|--------|---------|---------------|
| <i>T</i> =2000 | Mean ($\times 10^{-3}$) | <u>0.8499</u> | 2.7997 | 2.3234 | 10.0607 | 2.2238 |
| | Best ($\times 10^{-3}$) | <u>0.5109</u> | 1.2551 | 1.3271 | 2.6584 | 0.6843 |
| | Std Dev ($\times 10^{-3}$) | <u>0.3068</u> | 0.7156 | 0.5593 | 6.5832 | 0.9099 |
| | <i>t</i> -value | N/A | 17.71 | 16.33 | 9.88 | 10.12 |
| | Run Time (s) | 122.23 | 115.77 | 116.20 | 241.80 | <u>114.63</u> |
| | Rank | <u>1</u> | 4 | 3 | 5 | 2 |

TABLE XIII COMPARISON BETWEEN DIFFERENT PSO METHODS FOR MFD-EP (TESTING): (A) ENCAPSULATION WEIGHT, (B) ENCAPSULATION THICKNESS. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS.

(a)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|---------------------------------|-----------------|----------------------|--------|--------|--------|--------|
| Order | True value (mg) | Predicted value (mg) | | | | |
| 1 | 72.3 | 76.84 | 74.96 | 76.68 | 71.75 | 75.62 |
| 2 | 43.2 | 44.87 | 47.74 | 44.44 | 46.47 | 46.57 |
| 3 | 87.4 | 85.36 | 81.76 | 84.36 | 77.56 | 81.74 |
| 4 | 37.2 | 36.57 | 37.64 | 36.83 | 39.91 | 37.10 |
| 5 | 75.1 | 76.21 | 75.09 | 75.49 | 72.85 | 75.09 |
| 6 | 59.3 | 62.16 | 63.13 | 62.94 | 63.74 | 63.55 |
| 7 | 115 | 113.83 | 111.00 | 112.67 | 96.30 | 111.01 |
| 8 | 62.4 | 66.63 | 68.19 | 67.11 | 67.57 | 68.20 |
| 9 | 53.1 | 51.91 | 53.18 | 52.45 | 55.68 | 54.03 |
| Mean error ($\times 10^{-2}$) | | <u>0.1029</u> | 0.3035 | 0.2713 | 1.7036 | 0.3909 |
| Std Dev ($\times 10^{-2}$) | | <u>0.0553</u> | 0.2265 | 0.2060 | 0.9546 | 0.4679 |

(b)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|--------------------------------------|-----------------|----------------------|--------|--------|--------|--------|
| Order | True value (mm) | Predicted value (mm) | | | | |
| 1 | 0.58 | 0.6146 | 0.5834 | 0.5975 | 0.5638 | 0.5925 |
| 2 | 0.48 | 0.4999 | 0.5196 | 0.5104 | 0.5394 | 0.5076 |
| 3 | 0.67 | 0.6462 | 0.6068 | 0.6299 | 0.5937 | 0.6198 |
| 4 | 0.46 | 0.4686 | 0.4968 | 0.4901 | 0.5244 | 0.4907 |
| 5 | 0.62 | 0.6067 | 0.5978 | 0.6038 | 0.5875 | 0.6001 |
| 6 | 0.57 | 0.5714 | 0.5768 | 0.5614 | 0.5665 | 0.5642 |
| 7 | 0.71 | 0.7372 | 0.7234 | 0.7452 | 0.6213 | 0.7233 |
| 8 | 0.53 | 0.5679 | 0.5916 | 0.5840 | 0.5860 | 0.5885 |
| 9 | 0.53 | 0.5423 | 0.5595 | 0.5451 | 0.5743 | 0.5538 |
| Mean error ($\times 10^{-2}$) | | <u>0.7323</u> | 2.5255 | 2.0302 | 7.0928 | 1.9249 |
| Variance of mse ($\times 10^{-2}$) | | <u>0.3030</u> | 0.8276 | 0.7395 | 4.4626 | 0.7893 |

TABLE XIV COMPARISON BETWEEN DIFFERENT PSO METHODS FOR NN-BC. ALL RESULTS ARE AVERAGED ONES OVER 50 RUNS (RANK: 1-BEST, 5-WORST)

| | | HPSOWM | HPSOM | HGAPSO | HGPSO | SPSO |
|-------------|-----------------|-----------------|----------|----------|----------|----------|
| <i>T=50</i> | Mean | <u>186.7021</u> | 187.8136 | 188.1021 | 191.2301 | 187.9021 |
| | Best | <u>186.2866</u> | 186.4617 | 186.4721 | 186.9712 | 186.4704 |
| | Std Dev | <u>0.4781</u> | 0.9722 | 0.8652 | 2.3138 | 1.1023 |
| | <i>t</i> -value | N/A | 7.25 | 10.01 | 13.55 | 7.06 |
| | Run Time (s) | 663.14 | 665.12 | 657.82 | 1198.71 | 651.33 |
| | Rank | <u>1</u> | 2 | 4 | 5 | 3 |