Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications

S. H. Ling, *Member, IEEE*, H. H. C. Iu, *Senior Member, IEEE*, K. Y. Chan, H. K. Lam, *Member, IEEE*, Benny C. W. Yeung, and Frank H. Leung, *Senior Member, IEEE*

***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 the PSO in exploring the solution space more effectively for a better solution. A suite of benchmark test functions and three industrial applications (solving the load flow problems, modeling 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 empirically show that the proposed method significantly outperforms the existing methods in terms of convergence speed, solution quality, and solution stability.**

***Index Terms*—Load flow problem, modeling, mutation operation, neural network control, particle swarm optimization, wavelet theory.**

I. Introduction

PARTICLE swarm optimization (PSO) is a recently proposed population-based stochastic optimization algorithm which is inspired by the social behaviors of animals like fish schooling and bird flocking [6]. Comparing with other population-based stochastic optimization methods, such as evolutionary algorithms, the PSO has comparable or even superior search performance for many hard optimization problems with faster and more stable convergence rates [7]. The PSO has been used in different industrial areas, such as power systems [1], [18]–[21], parameter learning of neural networks (NNs) [16], [22], control [23], [24], prediction [25], and modeling [26], [27]. However, observations reveal that the PSO sharply converges 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 the local optima. As a result, it is hard to obtain any significant improvements by examining neighboring 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 neighborhood search.

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

For 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 a better solution in the later stage of the search by setting a smaller mutating space based on the properties of the wavelet [2]. This idea can be applied to introduce the hybrid PSO with the GA's mutation. In this paper, a mutation with a dynamic mutating space by incorporating a wavelet function is proposed. The wavelet is a tool to model seismic signals by combining the dilations and the translations of a simple oscillatory function (the mother wavelet) of finite duration. The PSO's mutating space is dynamically varying 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 faster convergence than the PSO with constriction and inertia weight factors [9] and other hybrid PSOs [1], [16], [17], [28] in solving a suite of benchmark test functions. In addition, it achieves better and more stable solution quality. Application examples on solving some load flow problems [the multicontingency transient stability constrained optimal power flow (MC-TSCOPF) problem and the economic load dispatch with valve-points loading (ELD-VPL) problem], modeling the development of fluid dispensing for electronic packaging, and

Manuscript received June 28, 2007; revised December 3, 2007. This work was supported in part by The University of Western Australia, Perth, W.A., Australia, and in part by a Grant (Project code G-U414) from The Hong Kong Polytechnic University, Kowloon, Hong Kong. This paper was recommended by Associate Editor H. Ishibuchi.

S. H. Ling is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576.

H. H. C. Iu is with School of Electrical, Electronic and Computer Engineering, The University of Western Australia, Perth, W.A. 6009, Australia.

K. Y. Chan is with the Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

H. K. Lam is with the Department of Electronic Engineering, Division of Engineering, The King's College London, WC2R 2LS London, U.K.

B. C. W. Yeung and F. H. Leung are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.921005

designing a NN-based controller (NN-BC) 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 a wavelet mutation (WM). 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 the parameter g for the WM is discussed in this section. Application examples on the load flow problems, modeling of fluid dispensing for electronic packaging, and the NN-BC are given in Section IV. A conclusion will be drawn in Section V.

II. Hybrid PSO With the WM

The PSO is a novel optimization method developed by Kennedy and Eberhart [6]. It models the processes of the sociological behavior 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 (SPSO) with constriction and inertia weight factors is shown in Fig. 1. In this paper, a hybrid PSO with the WM (HPSOWM) is proposed and shown in Fig. 2. The details of the SPSO and the HPSOWM will be discussed as follows.

A. SPSO With Constriction and Inertia Weight Factors

In Fig. 1(a), $X(t)$ denotes a swarm at the tth iteration. Each particle $\mathbf{x}^p(t) \in X(t)$ contains κ elements $x_j^p(t) \in \mathbf{x}^p(t)$ at the tth iteration, where $p = 1, 2, \ldots, \gamma$ and $j = 1, 2, \ldots, \kappa$; 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 the PSO is to iteratively minimize the fitness values (cost values) of particles. The swarm evolves from iteration t to $t+1$ by repeating the procedure as shown in Fig. 1. The SPSO [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 jth element of the pth particle at the tth iteration can be calculated using the following formulae:

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

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

where

$$pbest^p = [pbest_1^p \quad pbest_2^p \quad \cdots \quad pbest_\kappa^p]$$
$$gbest = [gbest_1 \quad gbest_2 \quad \cdots \quad gbest_\kappa]$$
$$j = 1, 2, \ldots, \kappa.$$

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

```
begin
    t→0                 // iteration number
    Initialize X(t)     // X(t): Swarm for iteration t
    Evaluate f(X(t)) // f(·): fitness function
while (not termination condition) do
    begin
        t→t+1
        // Process of SPSO //
          Update velocity v(t) and position of each particle x(t)
                based on (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→0                 // iteration number
    Initialize X(t)     // X(t): Swarm for iteration t
    Evaluate f(X(t)) // f(·): fitness function
while (not termination condition) do
    begin
        t→t+1
        Perform the process of 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. Pseudocode for (a) SPSO and (b) HPSOM.

```
begin
    t→0                 // iteration number
    Initialize X(t)     // X(t): Swarm for iteration t
    Evaluate f(X(t)) // f(·): fitness function
while (not termination condition) do
    begin
        t→t+1
        Perform the process of SPSO (shown in Fig. 1)
        // Process of Wavelet mutation operation
        Perform mutation operation with p_m
        Update x̄_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. Pseudocode for HPSOWM.

[0, 1]. In [5], an improved version of the PSO is presented, where the constriction and inertia weight factors are introduced. Here, when the SPSO with the constriction and inertia weight factors is used, (1) will be changed to

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

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 (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 $\varphi > 4$.

The SPSO utilizes $pbest^p$ and $gbest$ to modify the current search point to avoid the particles moving in the same direction, but to converge gradually toward $pbest^p$ 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 iterations, and $w_{\max}$ 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 sufficiently explore beyond local solutions. From experience, $v_{\max}$ is often set to 10%–20% of the dynamic range of the element on each dimension.

B. Hybrid PSO

We observe that the SPSO [5], [9] works well in the early stage, but usually presents problems on reaching the near-optimal solution. The behavior of the 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 premature convergence, and no further improvement can be obtained. This phenomenon is known as *stagnation* [4].

Ahmed *et al.* [1] proposed to integrate the GAs' mutation operation into the 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 the 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, and $para_{\max}^j$ and $para_{\min}^j$ are the upper and lower boundaries of each particle element, respectively. The pseudocode of the hybrid PSO with the 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 (b) that the pseudocodes of both PSO methods are identical except that the mutation operation is introduced.

However, it can be noticed from (6) that the mutating space in the 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 dynamically contracts along the search. We propose a WM that varies the mutating space based on the wavelet theory. The resulting HPSOWM (Fig. 2) is identical to the HPSOM except for the mutation operation used. The proposed WM is discussed in Section II-C.

C. WM

1) Wavelet Theory: Certain seismic signals can be modeled by combining translations and dilations of an oscillatory function within finite duration called a "wavelet." A continuous-time function $\psi(x)$ is called a "mother wavelet" or a "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

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

Property 2:

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

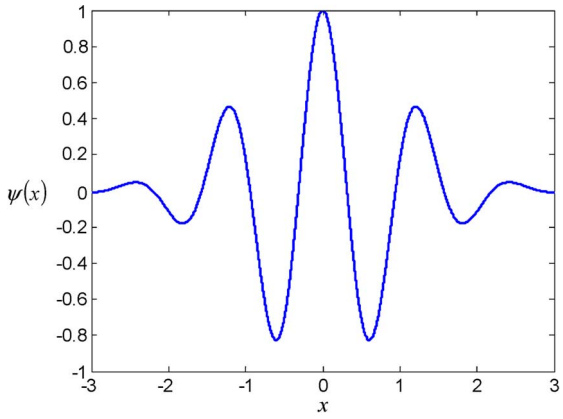


Fig. 3. Morlet wavelet.

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

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

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

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

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

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

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

As

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

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 to enhance the searching performance.

There are two reasons why the 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 Section III 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 iterations increases. Thus, the solution quality can be improved.

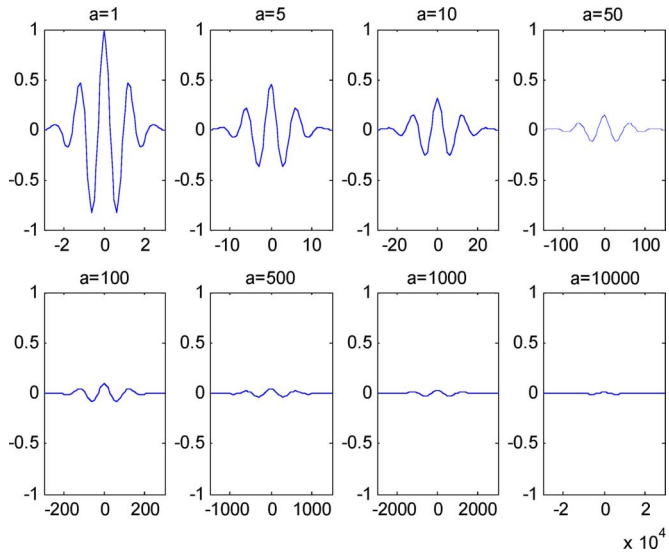


Fig. 4. Morlet wavelet dilated by different values of parameter a [x-axis: x, y-axis: $\psi_{a,0}(x)$].

2) Operation of the WM: The mutation operation is used to mutate the elements of particles. In general, various methods like uniform mutations or nonuniform mutations [8], [10] can be employed to realize the mutation operation. The proposed 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 that is 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, a mutation will take place on that element. For instance, if $\mathbf{x}^p(t) = [x_1^p(t), x_2^p(t), \ldots, x_\kappa^p(t)]$ is the selected pth particle, and the element of particle $x_j^p(t)$ is randomly selected for the 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 $\overline{\mathbf{x}}^p(t) = [\overline{x}_1^p(t), \overline{x}_2^p(t), \ldots, \overline{x}_\kappa^p(t)]$, i.e.,

$$\overline{x}_j^p(t) = \begin{cases} x_j^p(t) + \sigma \times \left(para_{\max}^j - x_j^p(t)\right) & \text{if } \sigma > 0 \\ x_j^p(t) + \sigma \times \left(x_j^p(t) - para_{\min}^j\right) & \text{if } \sigma \le 0 \end{cases} \tag{15}$$

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

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

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

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

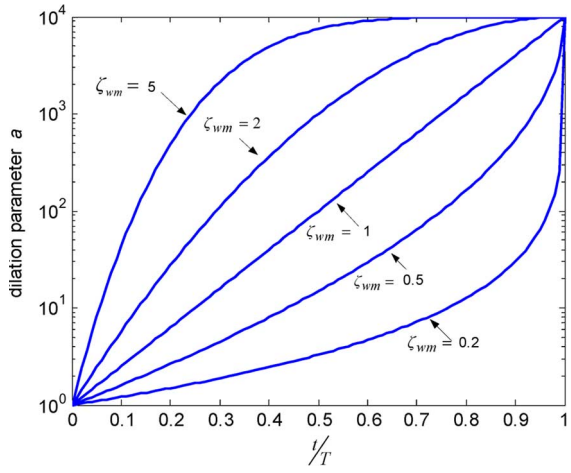


Fig. 5. Effect of the shape parameter ζ_{wm} to a with respect to t/T.

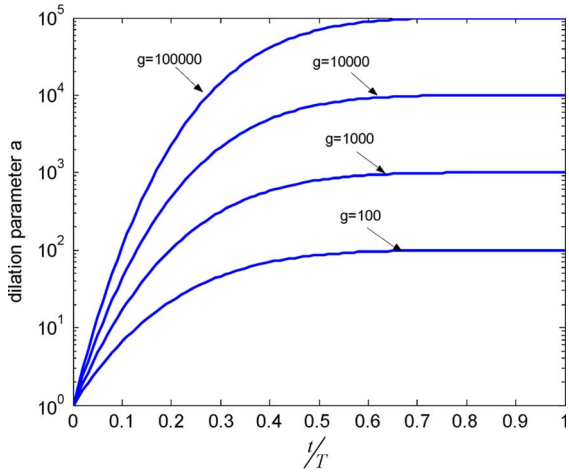


Fig. 6. Effect of the parameter g to a with respect to t/T.

operation because the selected wavelet function offers the best performance.

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

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

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, i.e.,

$$\frac{1}{N}\sum_{N} \sigma = 0 \quad \text{for} \quad N \to \infty \tag{19}$$

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 (a 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 randomly generated from $[-2.5a, 2.5a]$. The value of the dilation parameter a is set to vary with the value of t/T to meet the fine-tuning purpose, where T is the total number of iterations, and t is the current number of iterations. To perform a local search when t is large, the value of a should increase as t/T increases to reduce the significance of the mutation. Hence, a monotonic increasing function governing a and t/T is proposed as follows:

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

where ζ_{wm} is the shape parameter of the monotonic increasing function, and 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 Figs. 5 and 6, respectively. In this figure, g is set as 10 000. Thus, the value of a is between 1 and 10 000. 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 is given by $\overline{x}_j^p(t) = x_j^p(t) + 1 \times (para_{\max}^j - x_j^p(t)) = para_{\max}^j$. 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 $\overline{x}_j^p(t) = x_j^p(t) + 0.0158 \times (para_{\max}^j - x_j^p(t))$, 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 WM. The dilation parameter a will take a value 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 solution can be found easier by the searching algorithm, and 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 WM will be discussed in the next section.

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

3) Choosing the HPSOWM Parameters: The 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 the HPSOWM, is governed by the right choices of the control parameters: the swarm size γ, the probability of mutation p_m, the shape parameter ζ_{wm}, and the parameter g of the WM. Some views about these parameters are given as follows.

1) Increasing swarm size γ will increase the diversity of the search space and reduce the probability that the 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.

2) Increasing the probability of mutation p_m tends to transform the search into a random search such that when $p_m = 1$, all the elements of the particles will mutate. This probability gives us an expected number $(p_m \times \gamma \times \kappa)$ of elements of particles that undergo the mutation operation. In other words, the value of p_m depends on the desired number of elements of particles that undergo the mutation operation. Normally, when the dimension is very low (the number of elements in a particle is less than 5), p_m is set to 0.5–0.8. When the dimension is around 5–10, p_m is set to 0.3–0.4. When the dimension is in the range of 11–100, p_m is set to 0.1–0.2. When the dimension is in the range of 101–1000, normally, p_m is set to 0.05–0.1. Last, when the dimension is very high (the number of elements in a particle is larger than 1000), p_m is set to $<$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 an 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 the 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 are set by trial and error through experiments for good performance for all functions.

3) 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 WM. The dilation parameter a will take a value 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 fine-tuning can be done in an early iteration. Thus, a larger value of ζ_{wm} can be used to increase the step size of the early mutation. The 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 are more convenient to find a good setting. In Sections III-D and E, the sensitivity of ζ_{wm} and g to the WM with experimental results will be discussed. Based on the results, we suggest fixing the parameter g to 10 000 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 18 standard benchmark test functions [8], [13] is used to test the performance of the 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 the unimodal function, which is a symmetric model with a single minimum; f_1-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; $f_{14}-f_{18}$ belong to this type. The expressions of these functions are tabulated in Table I. (The details about parameters a, b, and c and function $u(\cdot)$ for functions f_8 and f_9 and $f_{12}-f_{14}$ are given in [13].)

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

The following simulation conditions are used.

- The shape parameter of the WM (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.)
- The parameter g of the WM: 10 000. (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.
- The probability of mutation for the HPSOWM, the HPSOM, and the 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, f_{11}, and f_{12}; $p_m = 0.8$ for f_{10}.)
- The probability of crossover for the HGAPSO (p_c): 0.8.
- The initial population: it is uniformly generated at random.
- The learning rate of the HGPSO is chosen by trial and error through experiments for good performance for all functions.

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

TABLE I
BENCHMARK TEST FUNCTIONS

Test function	Domain range	Optimal point
$f_1(\mathbf{x})=\sum_{i=1}^{30} x_i^2$	$-100 \le x_i \le 100$	$f_1(\mathbf{0})=0$
$f_2(\mathbf{x})=\sum_{i=1}^{9}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	$-2.048 \le x_i \le 2.048$	$f_2(\mathbf{1})=0$
$f_3(\mathbf{x})=\sum_{i=1}^{100}(\lfloor x_i+0.5\rfloor)^2$	$-10 \le x_i \le 10$	$f_3(\mathbf{0})=0$
$f_4(\mathbf{x})=\sum_{i=1}^{10} ix_i^4+random[0,1)$	$-2.56 \le x_i \le 2.56$	$f_4(\mathbf{0})=0$
$f_5(\mathbf{x})=\max_i\{\lvert x_i\rvert, 1\le i\le 30\}$	$-100 \le x_i \le 100$	$f_5(\mathbf{0})=0$
$f_6(\mathbf{x})=\sum_{i=1}^{30}\lvert x_i\rvert+\prod_{i=1}^{30}\lvert x_i\rvert$	$-10 \le x_i \le 10$	$f_6(\mathbf{0})=0$
$f_7(\mathbf{x})=-\cos(x_1)\cdot\cos(x_2)\cdot\exp\left(-\left((x_1-\pi)^2+(x_2-\pi)^2\right)\right)$	$-300 \le x_1, x_2 \le 300$	$f_7([\pi,\pi])=-1$
$f_8(\mathbf{x})=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$,	$-65.536 \le x_i \le 65.536$	$f_8([-32,-32])\approx 1$
$f_9(\mathbf{x})=\sum_{i=1}^{9}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	$-5 \le x_i \le 5$	$f_9(0.1928, 0.1928, 0.1231, 0.1358)\approx 0.0003075$
$f_{10}(\mathbf{x})=-\frac{\sin(x_1)\sin(x_2)}{x_1x_2}$	$-10 \le x_1, x_2 \le 10$	$\lim_{\mathbf{x}\to[0,0]} f_{10}(\mathbf{x})=-1$
$f_{11}(\mathbf{x})=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$-5 \le x_1, x_2 \le 5$	$f_{11}([0.08983,-0.7126])=$ $f_{11}([-0.08983,0.7126])\approx -1.0316$
$f_{12}(\mathbf{x})=-\sum_{i=1}^{30} c_i\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	$0 \le x_i \le 1$	$f_{12}(0.114, 0.556, 0.852)\approx -3.8628$
$f_{13}(\mathbf{x})=-\sum_{i=1}^{4} c_i\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right]$	$0 \le x_i \le 1$	$f_{13}([0.201, 0.15, 0.477, 0.275, 0.311, 0.627])\approx -3.32$
$f_{14}(\mathbf{x})=0.1\left\{\begin{array}{l}\sin^2(\pi 3x_1)\\ +\sum_{i=1}^{29}(x_i-1)^2\cdot\left[1+\sin^2(3\pi x_{i+1})\right]\\ +(x_{30}-1)^2\left[1+\sin^2(2\pi x_{30})\right]\end{array}\right\}+\sum_{i=1}^{30}u(x_i,5,100,4)$	$-50 \le x_i \le 50$	$f_{14}(\mathbf{1})=0$
$f_{15}(\mathbf{x})=\sum_{i=1}^{30}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	$-50 \le x_i \le 50$	$f_{15}(\mathbf{0})=0$
$f_{16}(\mathbf{x})=\frac{1}{4000}\sum_{i=1}^{30}x_i^2-\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1,$	$-600 \le x_i \le 600$	$f_{16}(\mathbf{0})=0$
$f_{17}(\mathbf{x})=-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30}x_i^2}\right)-\exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right)+20+e$	$-32 \le x_i \le 32$	$f_{17}(\mathbf{0})=0$
$f_{18}(\mathbf{x})=-\sum_{i=1}^{10}\left(x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	$-500 \le x_i \le 500$	$f_{18}([420.9687, \cdots 420.9687])$ $=-10\times 418.9829=-4189.829$

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 the HPSOWM are much better than those of the other methods; the mean cost value of the HPSOWM is about 180 to 1.2×10^7 times better. Also, the standard deviation is much better, which means that the searched solutions are more stable. In Fig. 7(a), the 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 iterations, whereas other optimization methods offer about 1×10^{-1}. Function f_2 is a generalized Rosenbrock's function, which is strongly nonseparable, and the optimum is located in a very narrow ridge. The tip of the ridge is very sharp, and it runs around a parabola. The HPSOWM performs better than the other methods in terms of the mean value and the standard deviation. Also, a good convergence rate for the 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

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

		HPSOWM	HPSOM	HGAPSO	HGPSO	SPSO
f_1 ($\times 10^{-5}$) T=1000	Mean	0.0015	0.2732	10834.8084	1586.8234	18217.2796
	Best	0.0004	0.0724	121.1258	59.7069	15.8949
	Std Dev	0.0010	0.1493	18020.5712	1368.0663	26642.2440
	Rank	1	2	4	3	5
f_2 ($\times 10^{0}$) T=1000	Mean	1.0030	1.4095	1.8935	3.0879	3.3426
	Best	0.5441	0.8952	1.1761	0.1428	2.1794
	Std Dev	0.2155	0.2653	0.3300	2.8497	0.4905
	Rank	1	2	3	4	5
f_3 ($\times 10^{0}$) T=500	Mean	0.8400	3.2000	5.0200	1140.22	32.0800
	Best	0.0000	0.0000	0.0000	933.000	13.0000
	Std Dev	0.9116	2.6186	11.7603	87.6000	24.1313
	Rank	1	2	3	5	4
f_4 ($\times 10^{-2}$) T=1000	Mean	0.5126	0.7845	0.5496	44.9282	0.7065
	Best	0.1783	0.2405	0.2934	16.0263	0.2758
	Std Dev	0.1687	0.2162	0.1341	17.2645	0.2295
	Rank	1	4	2	5	3
f_5 ($\times 10^{0}$) T=1000	Mean	0.2587	0.2684	1.9689	9.1558	2.6777
	Best	0.1057	0.0504	1.0284	0.5074	1.5338
	Std Dev	0.1070	0.1129	0.4418	14.6778	0.6149
	Rank	1	2	3	5	4
f_6 ($\times 10^{-5}$) T=1000	Mean	0.0872	0.9139	19560.3257	121.66×10^{5}	18189.8149
	Best	0.0224	0.2745	2067.5365	91.52×10^{5}	775.5302
	Std Dev	0.0440	0.5361	16787.6184	11.21×10^{5}	18356.8947
	Rank	1	2	4	5	3
f_7 ($\times 10^{0}$) T=100	Mean	−1.0000	−0.9949	−0.9999	−0.3060	−0.9999
	Best	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	Std Dev	1.00×10^{-9}	0.0364	1.36×10^{-5}	0.4609	4.02×10^{-5}
	Rank	1	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 ONCE 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	1	1	1	1	1
f_9 ($\times 10^{-3}$) T=500	Mean	1.0829	4.3140	3.0503	3.1956	1.4730
	Best	0.5844	0.6486	0.4209	0.3091	0.4041
	Std Dev	2.1739	9.8862	6.4610	9.8017	3.9022
	Rank	1	5	3	4	2
f_{10} ($\times 10^{-5}$) T=100	Mean	245.5877	245.5961	245.5878	249.7721	245.5879
	Best	245.5858	245.5858	245.5858	245.5858	245.5858
	Std Dev	0.0039	0.0179	0.0060	25.2583	0.0070
	Rank	1	4	2	5	3
f_{11} ($\times 10^{-4}$) T=100	Mean	−10316.284	−10316.266	−10316.285	−10316.280	−10316.285
	Best	−10316.285	−10316.284	−10316.285	−10316.285	−10316.285
	Std Dev	0.00008	0.02982	0.00000	0.01093	0.00000
	Rank	3	5	1	4	1
f_{12} ($\times 10^{-4}$) T=100	Mean	−38627.822	−38627.819	−38627.822	−38609.629	−38627.822
	Best	−38627.822	−38627.821	−38627.822	−38627.822	−38627.822
	Std Dev	0.00000	0.00259	0.00000	17.02938	0.00000
	Rank	1	4	1	5	1
f_{13} ($\times 10^{-4}$) T=100	Mean	−32934.608	−32577.928	−32744.379	−31597.339	−32818.652
	Best	−33219.952	−33219.952	−33219.952	−33219.952	−33219.952
	Std Dev	512.9276	598.5759	588.3683	1563.1222	577.3916
	Rank	1	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 ONCE OVER 50 RUNS (RANK: 1—BEST, 5—WORST)

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

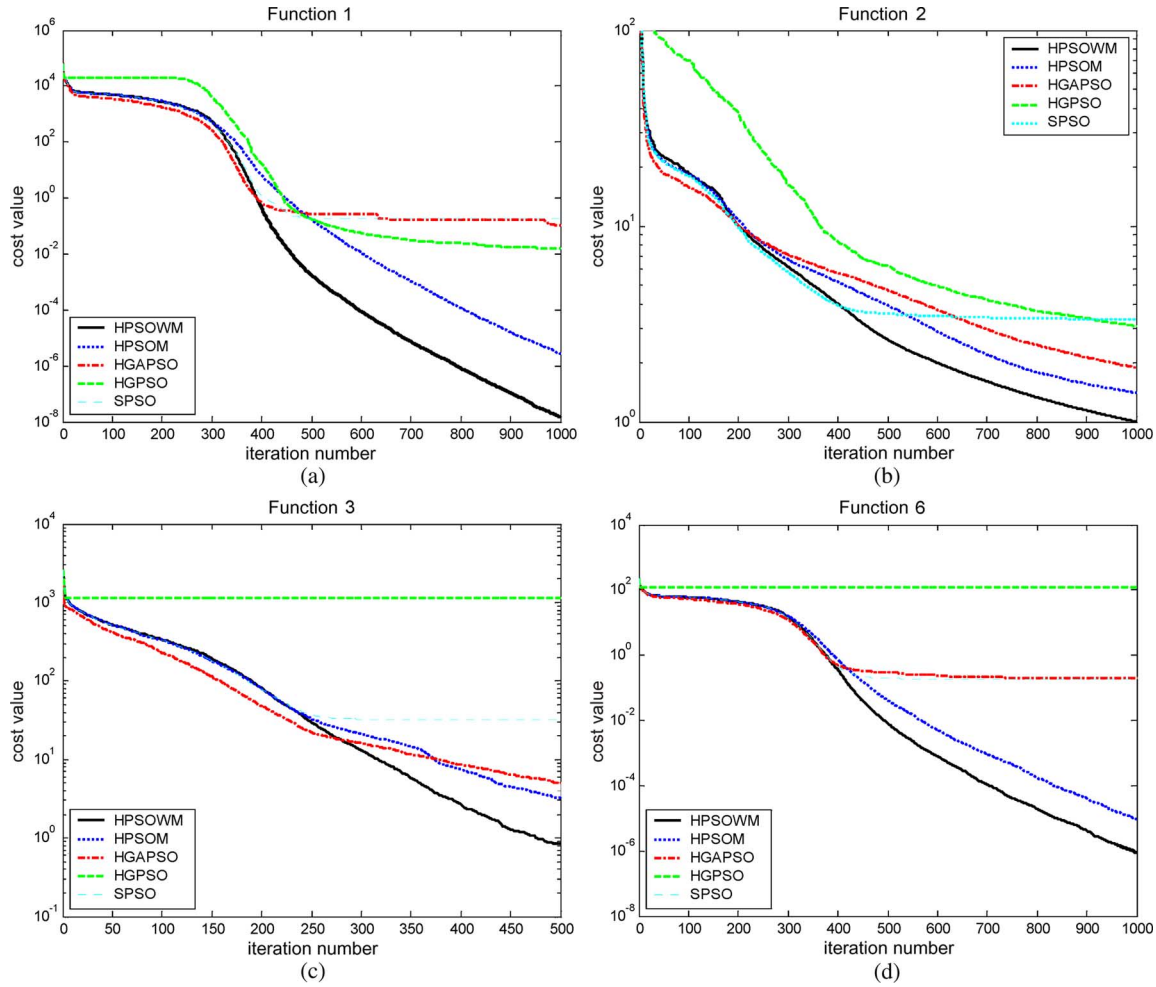


Fig. 7. Comparisons between different PSO methods for unimodal functions.

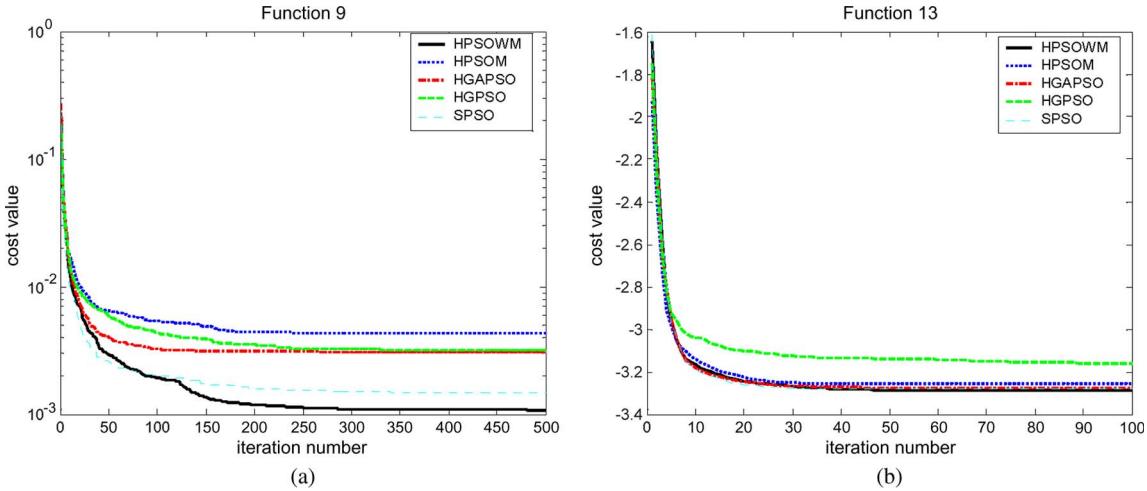


Fig. 8. Comparisons between different PSO methods for multimodal functions with a few local minima.

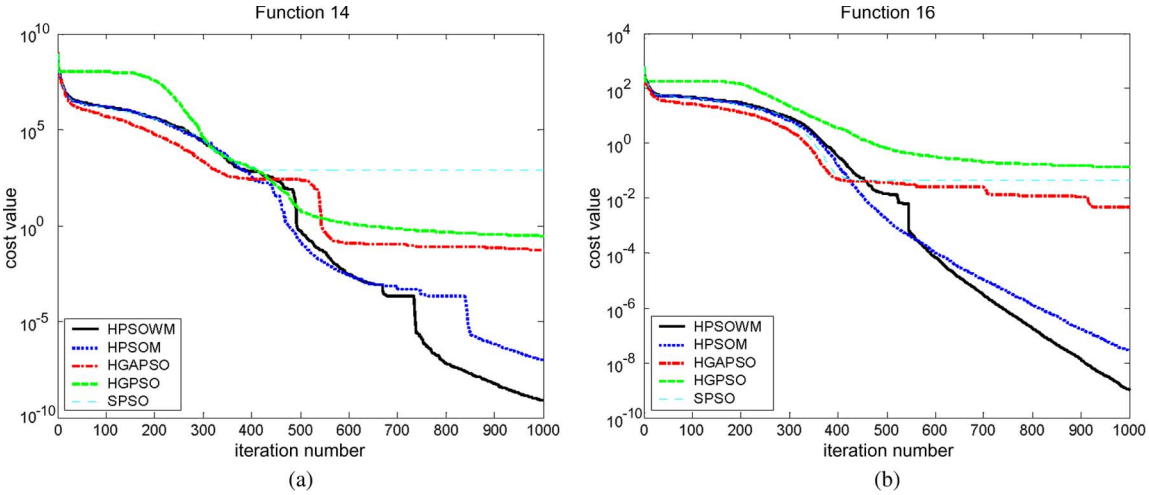


Fig. 9. Comparisons between different PSO methods for multimodal functions with many local minima.

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 that is 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 other optimization methods, the HPSOWM gives the best mean cost value. Function f_5 is Schwefel's problem 2.21. From Table II, although the best cost value of the HPSOWM is a little bit worse than that of the HPSOM, the mean cost value and the standard derivation of the HPSOWM are the best. Thus, the HPSOWM gives better solution quality and stability. Function f_6 is Schwefel's problem 2.22, and function f_7 is Eason's function. For these problems, the performance of the HPSOWM is better than that of the other methods. The rapid convergence of the HPSOWM, as shown in Fig. 7(c) and (d), supports our argument. In short, the 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 and $f_{10}-f_{12}$) do not show significant differences among the different optimization methods. They all reach or get near to the global optima; however, the HPSOWM still provides the smallest standard deviation in most cases. For functions f_9 and f_{13}, different results from the HPSOWM and the other methods are obtained. The HPSOWM gives better results in terms of the mean cost value and the standard deviation. Thus, the solution's stability and quality are good. According to Fig. 8(a) and (b), the convergence rate of the HPSOWM is faster than those of the others.

Category 3—Multimodal function with many local minima: Functions $f_{14}-f_{18}$ 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 clearly seen from Fig. 9(a) and (b) that if the PSO does not involve any mutation operation (the HGPSO and the 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 the HPSOWM

TABLE V
t-VALUE BETWEEN HPSOWM AND OTHER PSO METHODS

Functions	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

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 Ackley's and Schwefel's functions, respectively. From Table IV, we can see that the HPSOWM gives better results than the others. In general, the HPSOWM is good for handling multimodal functions with many local minima.

In conclusion, the HPSOWM gives the best performance for all kinds of optimization problems, particularly unimodal functions and multimodal functions with many local minima. It generally outperforms other hybrid PSOs and the 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 it is negative if it is poorer. The t-value is defined as follows:

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

where $\overline{\alpha}_1$ and $\overline{\alpha}_2$ are the mean values of the first and second methods, respectively; σ_1 and σ_2 are the standard deviations of the first and second methods, respectively; and ξ is the value of the degrees 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 the HPSOWM and 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 the HPSOWM is significantly better than that of 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 is used. 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 that there is no linking among these variables), we shift the global optimum points and rotate the test functions. That means, to make the variables 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 to avoid different variables to have the same numerical value after normalization at the global optimum point. We test five additional functions with shift: the shifted sphere function $(f_{1-\text{shift}})$, shifted Schwefel's problem 1.2 $(f_{2-\text{shift}})$, the 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 multimodal functions. The equations, the dimension, and the range of the variables are given in Table VI. The experimental results in terms of the mean cost value, the best cost value, the standard deviation, the t-value, and the convergence rate are summarized in Table VII and Fig. 10. The basic experimental setup is the same as that mentioned in Section III-A1. The shape parameters of the WM for $f_{1-\text{shift}} - f_{4-\text{shift}}$ are set to 5, and that for $f_{5-\text{shift}}$ is set to 2. The probability of mutation for the HPSOWM, the HPSOM, and the HGAPSO is set to 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 the HPSOWM, the HPSOM, and the HGAPSO show better performance in terms of the mean cost value and the standard deviation than the HSPSO and the SPSO. Based on this observation, we can see that when the PSO is without the mutation operation (the HSPSO and the PSO), it is hard to solve the optimization problems with the global optimum points shifted and rotated. Comparing with the PSO with the mutation operation (the HPSOWM, the HPSOM, and the HGAPSO), the performance of the HPSOWM is the best in terms of the mean value, the standard deviation, and the convergence rate.

TABLE VI
EQUATION OF BENCHMARK TEST FUNCTIONS WITH SHIFT

Test functions with shift	Domain range
$f_{1-shift}(\mathbf{x}) = \sum_{i=1}^{30} x_i^2$	$-100 \le x_i \le 100$
$f_{2-shift}(\mathbf{x}) = \sum_{k=1}^{30} \sum_{i=1}^{k} x_i^2$	$-100 \le x_i \le 100$
$f_{3-shift}(\mathbf{x}) = \sum_{i=1}^{10} \left(A^{\left(\frac{i-1}{9}\right)} \left(x_i^2\right) \right)$, where $A=1\times10^6$	$-100 \le x_i \le 100$
$f_{4-shift}(\mathbf{x}) = \sum_{i=1}^{30} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	$-100 \le x_i \le 100$
$f_{5-shift}(\mathbf{x}) = \sum_{i=1}^{30} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	$-5 \le x_i \le 5$

D. Sensitivity of the Shape Parameter for the WM

The mean cost values offered by the HPSOWM with different values of the WM's shape parameter ζ_{wm} for all test

TABLE VII
COMPARISON BETWEEN DIFFERENT PSO METHODS FOR BENCHMARK TEST FUNCTIONS WITH SHIFT. ALL RESULTS ARE AVERAGED ONCE OVER 50 RUNS (RANK: 1—BEST, 5—WORST)

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

functions in Section III-A are tabulated in Table VIII. The functions are tested by using $\zeta_{wm} = 0.2$, 0.5, 1, 2, and 5. In this experiment, the parameter g is fixed at 10 000. If the optimization problem needs a more significant mutation to reach the optimal point, a smaller ζ_{wm} should be used. Conversely, if the 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 that 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 the HPSOWM will perform the fine-tuning faster. On the other hand, ζ_{wm} can be set as 0.2 for f_3 (the 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 values for different values of ζ_{wm} are 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 the WM

The mean cost values offered by the HPSOWM with different values of the WM's parameter g for all test functions are tabulated in Table IX. The functions are tested by using $g = 100$, 1000, 10 000, and 100 000. In this experiment, the parameter ζ_{wm} is fixed at 5. If we want a smaller value of the upper limit (the searching limit) of the particle σ's mutated element, a larger value of g should be used. In some cases, 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 values of g have no significant difference. However, in some cases, such as f_9, the value of the parameter g is sensitive to the performance of the 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 $g = 100\,000$. Their difference is around four 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 $g = 10\,000$ gives better performance in general. As mentioned 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 10 000 and adjust the shape parameter ζ_{wm} to obtain an optimal monotonic increasing function for the WM operation.

IV. INDUSTRIAL APPLICATIONS OF THE HYBRID PSO

In this section, three industrial application examples on solving some load flow problems (the MC-TSCOPF problem and the ELD-VPL problem), modeling the development of the fluid dispensing process for electronic packaging, and designing of an NN-BC are used to illustrate the performance and the applicability of the proposed hybrid PSO.

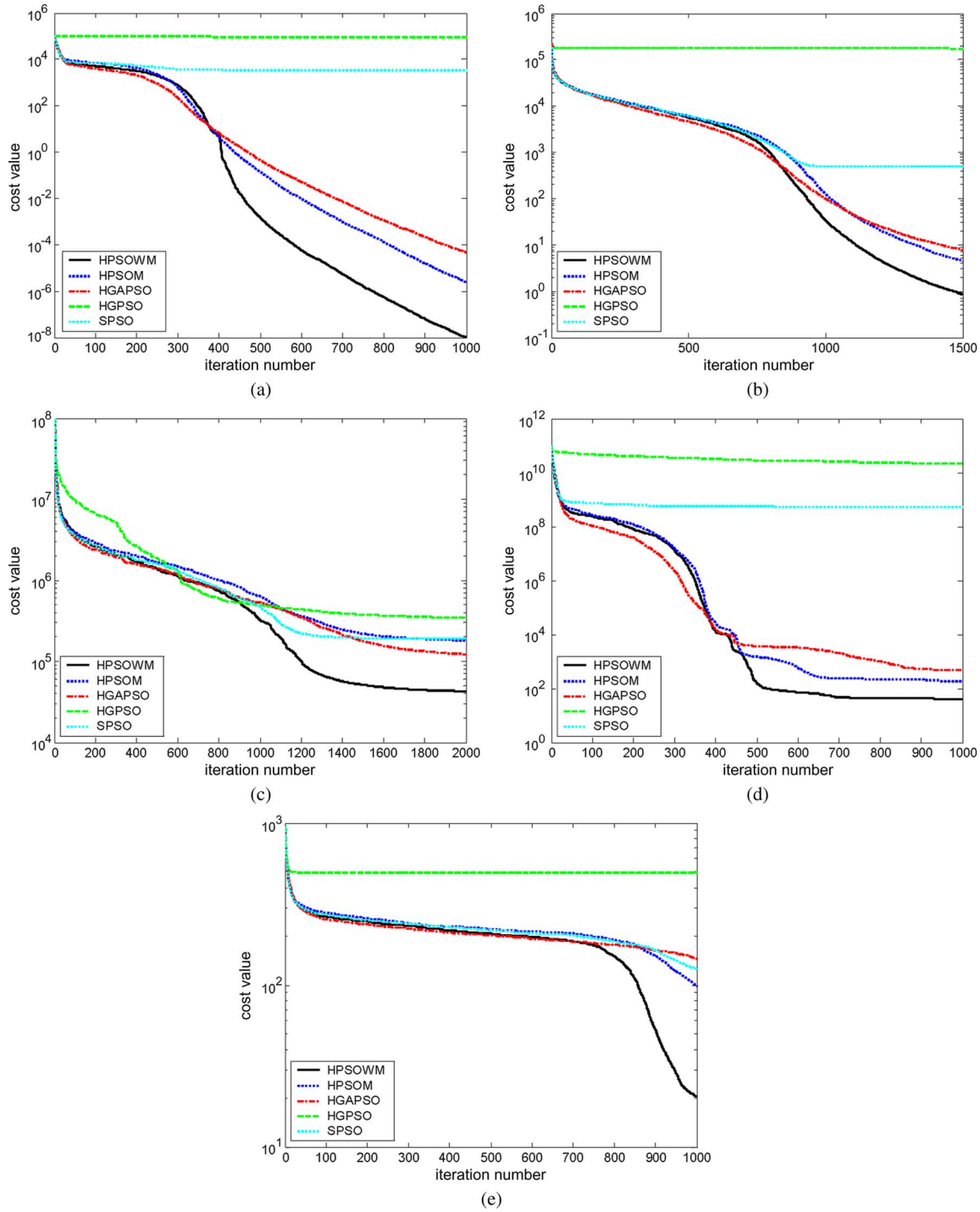


Fig. 10. Comparisons between different PSO methods for benchmark functions with shift.

A. Application I: Load Flow Problems

In this section, two application examples on load flow problems are given to show the performance of the HPSOWM. The problems are the MC-TSCOPF and the ELD-VPL. The load flow problem is a multimodal problem, which is suitable to be solved by the PSO.

1) MC-TSCOPF: It aims to achieve an optimal solution of a specific objective function, such as fuel cost and network loss, 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 nonconvexity nature of the OPF and the nonlinear differential algebraic equations that describe the transient stability constraints of the power system. Nonlinear and semidefinite 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 to ensure

TABLE VIII
SENSITIVITY OF THE SHAPE PARAMETER 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	0.0015
$f_2(\times 10^{0})$	4.7345	4.2675	3.6765	3.0992	1.0728
$f_3(\times 10^{0})$	0.8400	5.6400	17.22	30.30	31.88
$f_4(\times 10^{-3})$	5.2763	5.5626	5.5987	5.7988	5.1260
$f_5(\times 10^{0})$	0.3560	0.2768	0.2587	0.7753	1.1901
$f_6(\times 10^{-5})$	2.3139	0.8078	0.3044	0.1290	0.0872
$f_7(\times 10^{0})$	−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
$f_9(\times 10^{-3})$	1.0829	1.1684	2.1727	1.4457	2.3767
$f_{10}(\times 10^{-3})$	2.4562	2.4559	2.4559	2.4559	2.4559
$f_{11}(\times 10^{-4})$	−10316.1667	−10316.2804	−10316.2839	−10316.2844	−10316.2845
$f_{12}(\times 10^{-4})$	−38627.8173	−38627.8211	−38627.8215	−38627.8215	−38627.8215
$f_{13}(\times 10^{0})$	−3.2863	−3.2863	−3.2744	−3.2720	−3.2935
$f_{14}(\times 10^{-5})$	0.0397	0.0025	0.0003	0.0001	0.0001
$f_{15}(\times 10^{0})$	10.2854	16.8148	17.1133	18.1878	18.0486
$f_{16}(\times 10^{-5})$	0.0078	0.0009	0.0001	0.1166	0.1257
$f_{17}(\times 10^{-4})$	2.3264	0.6486	0.2504	0.1347	0.1061
$f_{18}(\times 10^{0})$	−3928.83	−3578.69	−3574.68	−3582.24	−3572.67

TABLE IX
SENSITIVITY OF THE PARAMETER g IN THE WAVELET MUTATION OPERATION

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

convexity [32]. Similar to the MC-TSCOPF problem, reactive power and voltage control problems, which are also mixed-integer nonlinear optimization problems, can be solved by the MC-TSCOPF with more promising results than the tested methods [47]. Here, a global optimization method, such as the PSO, is a good tool for handling the MC-TSCOPF problem.

a) Mathematical model for the MC-TSCOPF: The problem of the MC-TSCOPF is mathematically defined as follows:

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

such that

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

$$\mathbf{H}(\mathbf{x},\mathbf{y})\le 0 \tag{23b}$$

$$\mathbf{U}\left(\mathbf{x}(t),\mathbf{y}\right)\le 0,\quad t\in T \tag{23c}$$

where $\mathbf{x}(t)$ is a dependent vector that includes the active and reactive power of the swing bus, the voltage angle and the reactive power of the generator buses, and the 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 that includes the active power and the 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, the total network loss, the corridor transfer power, the total cost of compensation, etc. $\mathbf{g}$ is the set of equality constraints that are usually the power flow constraints for a specified operating condition. $\mathbf{H}$ is the set inequality constraints for the steady-state security limits such as bus voltage magnitude limits, generator power limits, and thermal limits for transmission lines. 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 implicitly imposed by the power flow calculation incorporated within the algorithm, and the inequality constraints $\mathbf{H}$ are directly satisfied by the PSO, the MC-TSCOPF can be formulated as a penalty function problem, i.e.,

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

Generally, transient stability constraints can be considered as hard constraints that should not be violated, whereas 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 large violation and discrimination, although the same penalty factor is used for all types of violations. Typically, $\beta=1000$ works very well in most power systems [33].

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

		HPSOWM	HPSOM	HGAPSO	HGPSO	SPSO
T=150	Mean	36452.97	36464.20	36474.90	36877.66	36675.48
	Best	36435.61	36439.66	36442.08	36512.71	36459.85
	Std Dev	8.91	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	1	2	3	5	4

b) 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 the robustness of the proposed hybrid PSO-based approach for solving the MC-TSCOPF problems. For comparison purposes, the HPSOM [1], the HGAPSO [17], the HGPSO [16], and the SPSO [9] are also used in this case study. The system data of the power system are collected in [36] and [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 0.01 s, and the integration time interval is fixed at 1.5 s. The total loads for the operating condition considered are 6.098 MW and 1.409 MVar. There are three LTCs 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.

1) 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.
2) 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 contingencies 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 iterations is set to 150. The dimension of this case is 22. The probability of mutation p_m and the shape parameter of the WM ζ_{wm} are set to 0.1 and 0.5, respectively. The shape parameter of the WM ζ_{wm} is chosen by trial and error through experiments for good performance. In this case, $\zeta_{wm} = 0.2, 0.5, 1, 2,$ 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 the HPSOWM are the smallest. The small standard deviation of the HPSOWM implies that it provides a stable and quality solution (the solution is robust). Also, the t-values between the HPSOWM and other optimization methods are higher than 2.06, and, thus, the 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 MC-TSCOPF problem.

2) ELD-VPL: The ELD is a method to schedule power generator outputs with respect to the load demands and to economically operate a power system 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. The PSO has been employed to solve the ELD-VPL problem.

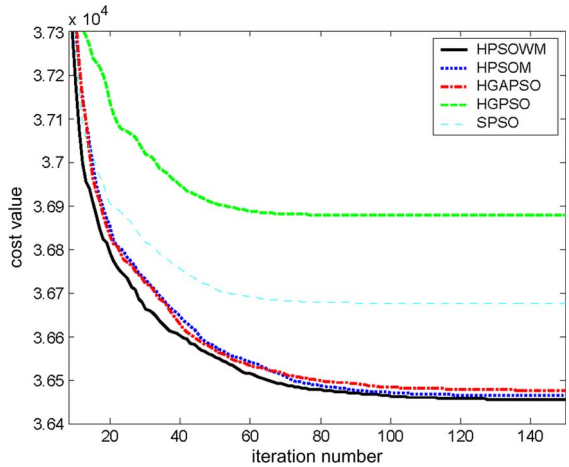


Fig. 11. Comparisons between different PSO methods for the MC-TSCOPF.

a) Mathematical model for the ELD-VPL: The ELD-VPL problem can be formulated into the following objective function:

$$\min \sum_{i=1}^{n} C_i(P_{L_i}) \tag{25}$$

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 constraints and generating capacity constraints as follows:

$$D = \sum_{i=1}^{n} P_{L_i} - P_{\text{Loss}} \tag{26}$$

$$P_{L_{i,\min}} \le P_{L_i} \le P_{L_{i,\max}}, \qquad i = 1, 2, \ldots, n \tag{27}$$

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

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

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

TABLE XI
COMPARISON BETWEEN DIFFERENT PSO METHODS FOR A 40-GENERATOR SYSTEM. ALL RESULTS ARE AVERAGED ONCE OVER 50 RUNS (RANK: 1—BEST, 5—WORST)

		HPSOWM	HPSOM	HGAPSO	HGPSO	SPSO
T=1000	Mean	122844.4	124350.87	124575.7	126855.7	126074.4
	Best	121915.3	122112.4	122780.0	124797.13	124350.4
	Std Dev	497.44	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	1	2	3	5	4

where a_i, b_i, and c_i are the coefficients of the cost curve of the ith generator, and e_i and f_i are the 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.)

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

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

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

$$P_{L_n} = D - \sum_{i=1}^{n-1} P_{L_i} + P_{\text{Loss}}. \tag{30}$$

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

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

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

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

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

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 to make sure that the output power of all generators is within the safety range. The objective is to minimize the cost function of (28) by using the PSO.

c) 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 demand (D) is 10 500 MW. The HPSOWM, the HPSOM [1], the HGAPSO [17], the HGPSO [16], and the 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 once after 50 runs. The dimension of this case is 39. The probability of mutation p_m and the shape parameter of the WM ζ_{wm} are set to 0.1 and 0.5, respectively. Similar to the example MC-TSCOPF,

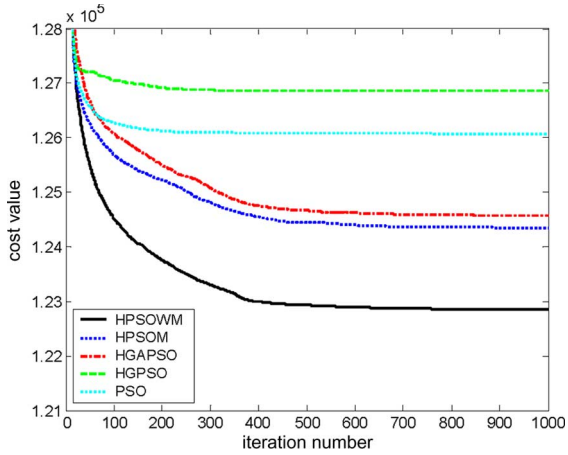


Fig. 12. Comparisons between different PSO methods for the ELD-VPL.

the shape parameter of the WM ζ_{wm} is chosen by trial and error through experiments for good performance. In this case, $\zeta_{wm} = 0.2,\ 0.5,\ 1,\ 2,$ and 5 are tried. For all approaches, the number of iterations is 2000. The statistical results in terms of the mean cost value, the best cost value, the standard deviation, 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 the HPSOWM is the best in terms of cost, t-values, and standard deviations. The average cost for the 40-generator system is \$122 844.40, and the best (minimum) cost is \$121 915.30. All t-values are higher than 2.06, implying that the HPSOWM is significantly better, with a 98% confidence level, than other hybrid PSOs. Due to the wavelet properties, the stability of the optimization is improved, and the smallest standard deviation is obtained by using the HPSOWM. To conclude, the solution quality and the stability of the HPSOWM are better.

B. Application II: MFD-EP

Fluid dispensing is a manufacturing process by which fluid materials are delivered to substrates, boards, or workpieces in a controllable manner. This process is widely used in various packaging processes in the electronics and semiconductor manufacturing industries, 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

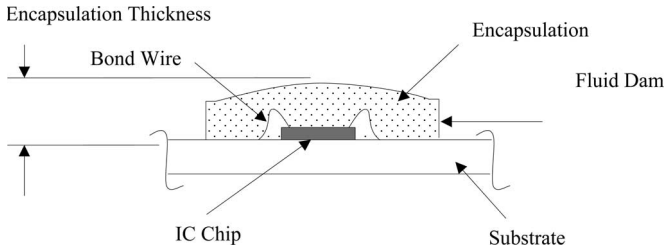


Fig. 13. Encapsulation of the microchip.

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 advantage of using the NN approach to process modeling is that it can provide learning and generalization abilities for nonlinearities. In this paper, a feedforward NN trained by the hybrid PSOs for modeling fluid dispensing for electronic packaging is given to illustrate the merits of the proposed PSO.

1) Fluid Dispensing Process for Electronic Packaging: Fluid dispensing is an important and popular process for electronic packaging. In this paper, modeling the fluid dispensing for microchip encapsulation is studied. Normally, silicon chips are covered using an $X-Y$ numerically controlled dispensing system that delivers a fluid encapsulant through a needle. The material is commonly dispensed in a pattern, working from the center 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.

Modeling the fluid dispensing process is critical for understanding the process behavior 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 (1–4 bar), x_1;
- the pump motor speed (400–1000 r/min), x_2;
- the height between the substrate and the needle (250–2000 steps of stepping motor), x_3.

Two quality characteristics were studied, which are the encapsulation weight (in milligrams) y and the encapsulation thickness (in millimeters) z.

2) Modeling With the Neural Network: A three-layer feedforward NN 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 that 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 allows for complex nonlinear interactions among the inputs to produce the desired output.

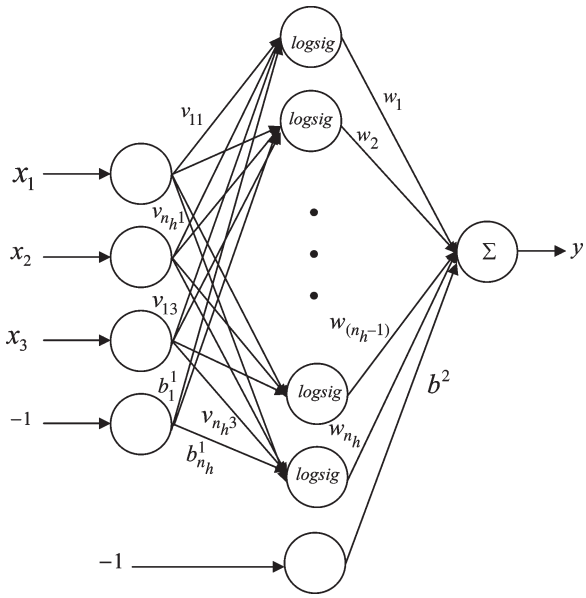


Fig. 14. Structure of the feedforward neural network.

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

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

$$z = \sum_{j=1}^{n_h'} w_j' \text{logsig}\left[\sum_{i=1}^{3}\left(v_{ji}'x_i - b_j^{1'}\right)\right] - b^{2'} \tag{35}$$

where n_h (or n_h') denotes the number of the hidden nodes; w_j (or w_j'), $j = 1, 2, \ldots, n_h$ (or n_h'), denotes the weight of the link between the jth hidden node and the output node; v_{ji} (or v_{ji}'), $i = 1, 2, 3$ and $j = 1, 2, \ldots, n_h$ (or n_h'), denotes the weight between the ith input node and the jth hidden node; b_j^1 (or $b_j^{1'}$) and b^2 (or $b^{2'}$) denote the biases for the jth hidden node and the output node, respectively; $\text{logsig}(\cdot)$ denotes the logarithmic sigmoid function, i.e.,

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

To develop the NN-based model for the fluid dispensing process, the values of the neural network parameters (i.e., w_j, v_{ji}, b_j^1, and b^2 with $i = 1, 2, 3$ and $j = 1, 2, \ldots, n_h$) 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 NN-based process model.

To tune the parameter values of the network, we use the hybrid PSO to minimize the mse by setting the swarm particle to be $[v_{ji} \quad w_j \quad b_j^1 \quad b^2]$ for all i and j. The mse for the

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

(a)

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

(b)

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

encapsulation weight y and for the encapsulation thickness z are defined as follows:

$$\text{mse}_y = \frac{\sum_{k=1}^{n_{\text{pat}}} \left(d_k^y - y_k\right)^2}{n_{\text{pat}}} \tag{37}$$

$$\text{mse}_z = \frac{\sum_{k=1}^{n_{\text{pat}}} \left(d_k^z - z_k\right)^2}{n_{\text{pat}}} \tag{38}$$

where d_k^y and d_k^z denote 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

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

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

3) Case Study: Modeling the fluid dispensing for electronic packaging (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. To test the learning ability of the NN trained by the proposed HPSOWM, a set of nine testing patterns is used. For comparison purposes, the NN models are also trained by the HPSOM [1], the HGAPSO [17], the HGPSO [16], and the 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 iterations is set to 2000. The probability of mutation p_m and the shape parameter of the WM ζ_{wm} are set to 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 to 5 and 7, respectively. In other words, the total numbers of parameters (the dimension) are 26 and 36, respectively. The training results are tabulated in Table XII, and the comparisons between different PSOs are shown in Figs. 15 and 16. The table shows

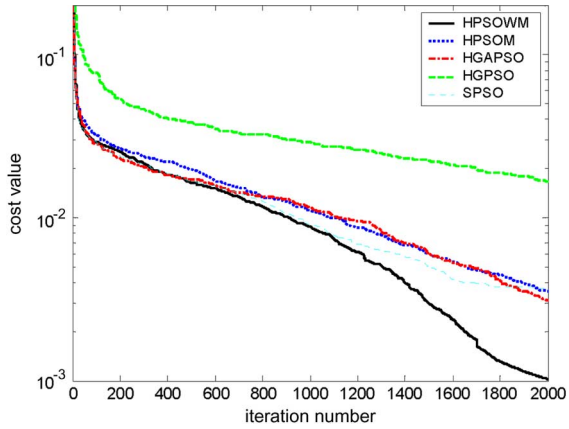


Fig. 15. Comparisons between different PSO methods for the MFD-EP (encapsulation weight).

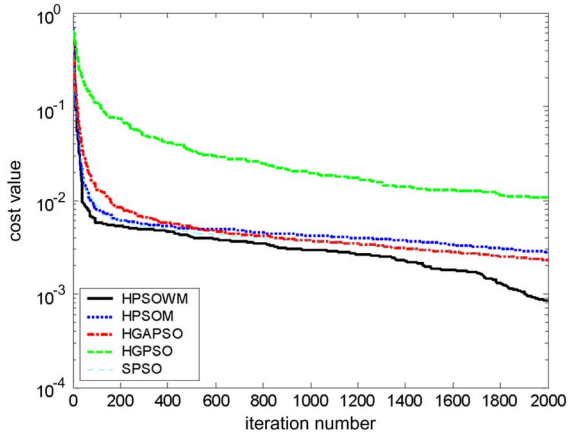


Fig. 16. Comparisons between different PSO methods for the MFD-EP (encapsulation thickness).

TABLE XIII
COMPARISON BETWEEN DIFFERENT PSO METHODS FOR MFD-EP (TESTING). (a) ENCAPSULATION WEIGHT. (b) ENCAPSULATION THICKNESS. ALL RESULTS ARE AVERAGED ONCE 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 ($\times 10^{-2}$)		0.1029	0.3035	0.2713	1.7036	0.3909
Std Dev ($\times 10^{-2}$)		0.0553	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 ($\times 10^{-2}$)		0.7323	2.5255	2.0302	7.0928	1.9249
Variance of mse ($\times 10^{-2}$)		0.3030	0.8276	0.7395	4.4626	0.7893

that the mean value, the best cost value, and the standard deviation offered by the HPSOWM are the smallest. Also, the t-values between the HPSOWM and other optimization methods are higher than 2.06, and, thus, the HPSOWM is significantly better than other methods with a 98% confidence level. The computational time of the HPSOWM is near to that of the other PSOs. (The HGPSO needs much more time than others.) Nine validation tests are carried out to evaluate the generalization ability of the NNs with different PSO methods. Table XIII shows the validation results yielded by the NN models for the encapsulation weight and the encapsulation thickness. From the table, the 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: NN-BC

In this application, an NN-BC realized by a three-layer feedforward fully connected neural network is proposed to stabilize a mass-spring-damper system [41]. The open-loop system can be described as follows:

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

where u is the force, and $x(t)$ and $\dot{x}(t)$ are the displacement and the velocity of the mass, respectively. This problem is considered as a multimodal optimization problem. A two-input two-output NN with four hidden nodes is employed to close the feedback loop. The total number of network parameters is 22. Denoting the outputs of the NN as $y_1(t)$ and $y_2(t)$, the NN-BC 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 NN-BC is defined as follows:

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

The control objective is to stabilize the mass-spring-damper system of (40), i.e., $x(t) \to 0$ and $\dot{x}(t) \to 0$ as $t \to \infty$. 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 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 a 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 $\mathbf{x}(0) = \begin{bmatrix} \frac{22\pi}{45} & 0 \end{bmatrix}^T$.

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

		HPSOWM	HPSOM	HGAPSO	HGPSO	SPSO
T=50	Mean	186.7021	187.8136	188.1021	191.2301	187.9021
	Best	186.2866	186.4617	186.4721	186.9712	186.4704
	Std Dev	0.4781	0.9722	0.8652	2.3138	1.1023
	t-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	1	2	4	5	3

For comparison purposes, the NN-BC is also trained by the HPSOM [1], the HGAPSO [17], the HGPSO [16], and the 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 randomly generated in the ranges of -1 to 1, and -200 to 200, respectively. The number of iterations is set to 50. The probability of mutation p_m and the shape parameter of the WM ζ_{wm} are set to 0.05 and 0.2, respectively. Fifty 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, the best cost values, and the standard deviations offered by various learning methods. It can be seen that the HPSOWM offers the best performance. Also, the t-values between the HPSOWM and other optimization methods are higher than 2.06, and, thus, the 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 the WM (the HPSOWM). Our objective is to apply the properties of the wavelet theory to enhance the PSO so that it can explore the solution space more effectively on reaching the solution. Simulation results have shown that the proposed WM-based hybrid PSO is a useful tool to solve optimization problems. Due 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, the HPSOWM gives better results than the methods of the HPSOM, the HGAPSO, the HGPSO, and the SPSO. Also, a faster convergence speed can be achieved by the HPSOWM. Comparing their runtime (computation time), the HGPSO consumes more time because of the process of the gradient descent. The other methods, including the 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, the HPSOWM shows better performance than other existing PSO methods.

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 Syst.*, vol. 20, no. 2, pp. 859–866, May 2005.

[2] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[3] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, Oct. 1995, pp. 39–43.

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

[5] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Jul. 2000, vol. 1, pp. 84–88.

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

[7] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[8] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, 2nd ed. New York: 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," *Proc. IET—Gene. Transm. Distrib.*, 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. Evol. Comput.*, Indianapolis, IN, 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 Nature*, 1992, vol. 2, 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 Syst.*, vol. 20, no. 2, pp. 1070–1078, May 2005.

[13] X. Yao and Y. Liu, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[14] C. R. Reeves, "Genetic algorithms and neighborhood search," in *Proc. Evol. Comput.—AISB Workshop*, 1994, pp. 115–130.

[15] J. M. Zurada, *Introduction to Artificial Neural Systems*. Singapore: 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 Symp. Syst. Theory*, 2004, pp. 150–153.

[17] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.

[18] T. O. Ting, 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 Syst.*, 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," *IEEE Trans. Power Syst.*, 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 Syst.*, 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 Syst.*, 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. Syst., Man, Cybern.*, Oct. 2000, vol. 4, 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 Netw.*, vol. 18, no. 2, pp. 595–601, Mar. 2007.

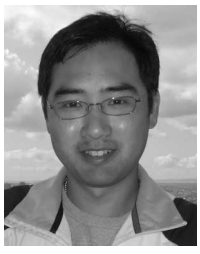
[24] Y. S. Wang, K. J. Wang, J. S. Qu, and Y. R. Yang, "Adaptive inverse control based on particle swarm optimization algorithm," in *Proc. IEEE Int. Conf. Mechatronics, Autom.*, Jul. 2005, pp. 2169–2172.

[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, Autom.*, Jun. 2006, pp. 2454–2458.

[26] R. Marinke, E. Araujo, L. S. Coelho, and I. Matiko, "Particle swarm optimization (PSO) applied to fuzzy modeling in a thermal-vacuum system," in *Proc. 5th Int. Conf. Hybrid Intell. Syst.*, Nov. 2005, pp. 67–72.

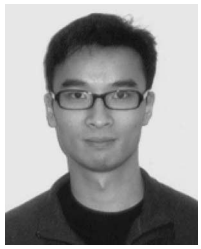
[27] Y. Liu and X. He, "Modeling identification of power plant thermal process based on PSO algorithm," in *Proc. Amer. Control Conf.*, Jun. 2005, vol. 7, pp. 4484–4489.

[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 Congr. Evol. Comput.*, 2003, vol. 4, pp. 2393–2399.
[29] J. Carpentier, "Optimal power flows," *Int. J. Electr. Power Energy Syst.*, vol. 1, no. 1, pp. 3–15, Apr. 1979.
[30] J. A. Momoh, R. J. Koessler, and M. S. Bond, "Challenges to optimal power flow," *IEEE Trans. Power Syst.*, 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. Electr. Power Energy Syst.*, 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 Syst.*, 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. Adv. Power Syst. Control, Oper. Manag.*, 2006. CD-ROM.
[34] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 7, no. 5, pp. 445–455, Oct. 2003.
[35] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000.
[36] J. H. Chow, *Power System Toolbox Version 2.0*. Colborne, ON, Canada: Cherry Tree Scientific Software, 2000.
[37] R. Zimmerman and D. Gan, *MATPOWER: A Matlab Power System Simulation Package*. Ithaca, NY: Power Syst. Eng. Res. Center (PSERC), School Electr. Eng., Cornell Univ., 2005.
[38] K. L. Petri, R. E. Billo, and B. Bidanda, "A neural network process model for abrasive flow machining operations," *J. Manuf. Syst.*, vol. 17, no. 1, pp. 52–65, 1998.
[39] E. Brinksmeier, H. K. Tonshoff, C. Czenkusch, and C. Heinzel, "Modelling and optimization of grinding processes," *J. Intell. Manuf.*, vol. 9, no. 4, pp. 303–314, Sep. 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. Prod. Res.*, vol. 38, no. 1, pp. 119–139, Jan. 2000.
[41] B. C. Kuo, *Automatic Control Systems*, 8th ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.
[42] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[43] P. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, May 1998, pp. 84–89.
[44] J. J. Lang, S. Baskar, P. Suganthan, and A. Qin, "Performance evaluation of multi agent genetic algorithm," *Natural Comput.*, 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 Int. Swarm Intell. Symp.*, 2005, pp. 68–75.
[46] P. H. Chen and H. C. Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Trans. Power Syst.*, 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 Syst.*, vol. 15, no. 4, pp. 1232–1239, Nov. 2000.



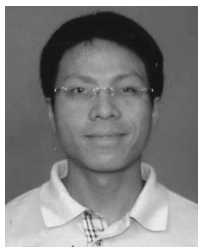
S. H. Ling (S'03–M'06) received the B.Eng., M.Phil., and Ph.D. degrees from the Hong Kong Polytechnic University, Kowloon, in 1999, 2002, and 2006, respectively.

From 2006 to 2007, he was with the School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth, as Postdoctoral Research Associate. He is currently a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Kent Ridge. He is the author or coauthor of more than 50 international journal and conference proceedings papers on computational intelligence and its industrial applications. His current research interests include evolution computations, fuzzy logics, neural networks, hybrid systems, and job shop scheduling.



H. H. C. Iu (S'98–M'00–SM'06) received the B.Eng. (Hons.) degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, in 1997 and the Ph.D. degree from the Hong Kong Polytechnic University, Kowloon, in 2000.

He is currently a Senior Lecturer with the School of Electrical, Electronic and Computer Engineering, University of Western Australia, Perth. He has published more than 60 papers. He currently serves as a Guest Editor for the *Australian Journal of Electrical and Electronics Engineering and Circuits, Systems and Signal Processing*. His research interests include nonlinear circuits and systems, power electronics, and TCP network dynamics.

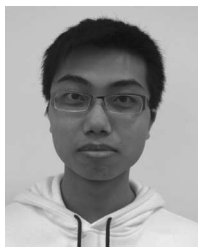


K. Y. Chan received the M.Phil. degree in electronic engineering from the City University of Hong Kong, Kowloon, and the Ph.D. degree in computing from London South Bank University, London, U.K.

He is currently a Postdoctoral Research Fellow with the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, Kowloon. His research interests include computational intelligence and its applications in product design, signal processing, power systems, and operation research.

H. K. Lam (M'98) received the B.Eng. (Hons.) and Ph.D. degrees from the Hong Kong Polytechnic University, Kowloon, in 1995 and 2000, respectively.

From 2000 to 2005, he was with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, as a Postdoctoral Fellow and a Research Fellow. He joined King's College London in 2005 as a Lecturer. His current research interests include intelligent control systems and computational intelligence.



Benny C. W. Yeung received the B.Eng. (Hons.) degree in electronic and information engineering from the Hong Kong Polytechnic University, Kowloon, in 2006.

He is currently a Research Student with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University. His current research interests include computational intelligence and computer vision.

Frank H. Leung (M'92–SM'03) was born in Hong Kong in 1964. He received the B.Eng. degree and the Ph.D. degree in electronic engineering from the Hong Kong Polytechnic University, Kowloon, in 1988 and 1992, respectively.

He joined the Hong Kong Polytechnic University in 1992 and is currently an Associate Professor with the Department of Electronic and Information Engineering. He is active in research and has published more than 150 research papers on computational intelligence and control and power electronics and is currently involved in the R&D on intelligent home automation. He also took his industry-based training on PABX and telephone systems.

Dr. Leung is currently an Executive Committee Member of the IEEE Hong Kong Chapter of Signal Processing. He is a Chartered Engineer and a Corporate Member of the Institution of Electrical Engineers, U.K. He was a recipient of the Sir Edward Youde Memorial Fellowship in 1989 and 1990. He has been serving as a Reviewer for many international journals and helping organize many international conferences.