Journal of Intelligent Learning Systems and Applications, 2012, 4, 81-97
http://dx.doi.org/10.4236/jilsa.2012.42008 Published Online May 2012 (http://www.SciRP.org/journal/jilsa)

An Improved Differential Evolution and Its Industrial Application

Johnny Chung Yee Lai[1], Frank Hung Fat Leung[1], Sai Ho Ling[2], Edwin Chao Shi[1]

[1]Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, China; [2]Centre for Health Technologies, Faculty of Engineering and IT, University of Technology Sydney, Sydney, Australia.
Email: 08900438r@polyu.edu.hk

Received April 9th, 2012; revised May 9th, 2012; accepted May 16th, 2012

ABSTRACT

In this paper, an improved Differential Evolution (DE) that incorporates double wavelet-based operations is proposed to solve the Economic Load Dispatch (ELD) problem. The double wavelet mutations are applied in order to enhance DE in exploring the solution space more effectively for better solution quality and stability. The first stage of wavelet operation is embedded in the DE mutation operation, in which the scaling factor is governed by a wavelet function. In the second stage, a wavelet-based mutation operation is embedded in the DE crossover operation. The trial population vectors are modified by the wavelet function. A suite of benchmark test functions is employed to evaluate the performance of the proposed DE in different problems. The result shows empirically that the proposed method out-performs signifycantly the conventional methods in terms of convergence speed, solution quality and solution stability. Then the proposed method is applied to the Economic Load Dispatch with Valve-Point Loading (ELD-VPL) problem, which is a process to share the power demand among the online generators in a power system for minimum fuel cost. Two different conditions of the ELD problem have been tested in this paper. It is observed that the proposed method gives satisfactory optimal costs when compared with the other techniques in the literature.

Keywords: Differential Evolution; Evolutionary Algorithm; Economic Load Dispatch

1. Introduction

Economic Load Dispatch (ELD) is one of the key considerations when operating a power generation system. Electric power utilities are expected to maximize the profit by minimizing the operating cost on generating the power. The loading demand and transmission losses must be met on providing a stable power supply for the users. For secure operation, the demand of power should be dispatched to different generators, so that the generation capacity limits are not exceeded. Therefore, the ELD problem can be formulated as an optimization problem [1]. Its objective is to reduce the total power generation cost of a group of power generators while satisfying different constraints. Because of the valve-point loadings and rate limits, the input-output characteristics of modern generators are nonlinear by nature. As a result, the characteristics of ELD with Valve-Point Loading (ELD-VPL) problem are multimodal, discontinuous, highly nonlinear, large-dimensional and highly constrained; which is difficult to solve. The classical gradient techniques failed to address the ELD-VPL problem satisfactorily. As a result, Evolutionary Algorithms (EAs) have been introduced to handle it. For example, the PSO methods were applied to solve the ELD-VPL problem in [2,3], and an algorithm called DEC-SQ [4] was used to tackle the problem.

Differential Evolution (DE) has been well accepted as a powerful algorithm for handling optimization problems during the last decade. Proposed by Storn and Price [5], DE is a population based stochastic optimization algorithm that searches the solution space based on the weighted difference between two population vectors. No separate probability distribution has to be used so that the scheme is completely self-organizing. It is a new member of the class of Evolutionary Algorithms (EAs) that imitate the process of biological evolution. Owing to the population based strategy, EAs are less possibly getting trapped in a locally optimal solution. Apart from DE, other EAs include Genetic Algorithm (GA) [6] and Evolutionary Programming (EP) [7,8].

Similar to GA, DE uses evolutionary operations of mutation and crossover to make the population evolving towards the global solution within the given solution space. Comparing with other optimization algorithms, DE is easy to implement, requires fewer parameters for tuning, and has a relatively fast convergence speed. A simple vector subtraction is able to generate a random direction of exploration over the solution space. DE can also offer a high degree of variations for the population to search the solution. It has been successfully applied in a number of optimization benchmark functions [9] and in a wide range of optimization problems such as data clustering [10], power plant control [11], optimization of nonlinear functions [12], electromagnetic inverse scattering problems [13], etc. However, for maintaining the diversity from one generation of population to the next, mutation takes an important role in the evolution process. The presence of mutation can help assuring the reached solution is a global optimum; but a too vigorous mutation in every iteration step may slow down or even destroy the convergence of the algorithm [14,15].

On doing the mutation and crossover operation, we can have the solution space to be more widely explored in the early stage of the search; and it is more likely to obtain a fine-tuned global solution in the later stage of the search by setting a smaller searching space. This can be realized by considering the properties of a wavelet function [16]. The wavelet is a tool to model seismic signals by combining dilations and translations of a simple, oscillatory function (mother wavelet) of a finite domain. In this paper, mutation and crossover operations within a searching space that take advantage of some wavelet functions are proposed. The double mutation operations in the proposed method aid the DE to perform more efficiently and provide a faster convergence than the standard DE in finding the solution for the ELD-VPL problem. In addition, it can achieve better solution quality and stability.

This paper is organized as follows: Section 2 presents the details of the DE with double wavelet mutations. Experimental study and analysis for a suite of benchmark functions are given in Section 3. These functions serve as good platforms to evaluate the performance of the proposed method. The problem formulation of ELD-VPL is given in Section 4. Experimental study of applying the proposed method to the ELD-VPL problem is given in Section 5. A conclusion will be drawn in Section 6.

2. DE with Double Wavelet Mutations

To realize DE, a randomly generated population over the solution space is first obtained. The population of solution vectors is then successively updated until the population converges to the optimum. The pseudo code for the standard DE (SDE) process is shown in **Figure 1**. In this paper, an algorithm called DE with double wavelet mutation (DWM-DE) is proposed and the pseudo code of it is shown in **Figure 2**. The details of both the SDE and the DWM-DE are discussed as follows.

2.1. Standard Differential Evolution (SDE)

DE attempts to maintain a population of N_p vectors for each generation of evolution, with each vector contains D elements. Let $P_{x,g}$ be the population of the current generation g, and $\mathrm{x}_{i,g}$ be the i-th vector in this population:

```
begin
Initialize the population
  While (not termination condition) do
  begin
    Mutation operation by Equation (2)
    Crossover operation by Equation (3)
    Evaluation of the fitness function
    Select the best vector by Equation (4)
  end
end
```

Figure 1. Pseudo code for SDE.

```
begin
Initialize the population
  While (not termination condition) do
  begin
    Mutation operation by Equation (2)
    Update the new value of F by Equation (12)
    Crossover operation by Equation (3)
    Modifying the trail population vectors
    based on Equation (15)
    Evaluation of the fitness function
    Select the best vector by
    Equation (4)
  end
end
```

Figure 2. Pseudo code for the proposed DE.

$$\begin{aligned} P_{x,g} &= \left(\mathrm{x}_{i,g}\right), i = 0,1,\cdots,N_p - 1; g = 0,1,\cdots,g_{\max} \\ \mathrm{x}_{i,g} &= \left(x_{j,i,g}\right), j = 0,1,\cdots,D-1. \end{aligned} \quad (1)$$

where g_{max} is the maximum generation number. Before the population can be initialized over the solution space, the boundary of the searching space should be specified. The population should be uniformly and randomly distributed in the searching space. Once initialized, DE creates a mutated vector, $\mathbf{v}_{i,g}$ for each target vector $\mathbf{x}_{i,g}$ by using the mutation operation. This operation adds a scaled, randomly sampled, vector difference to $\mathbf{x}_{i,g}$ to form a third vector. The mutated vector is therefore realized by the following equation:

$$\mathrm{v}_{i,g} = \mathrm{x}_{i,g} + F \cdot \left(\mathrm{x}_{r_1,g} - \mathrm{x}_{r_2,g}\right) \quad (2)$$

where F is the scaling factor; r_1 and r_2 are two different integers which are randomly generated from {0, 1,…, N_p − 1}. To complement the differential mutation search strategy and increase the diversity of the perturbed vectors, DE employs a method called uniform crossover for all the mutated vectors. Each vector element pair $x_{j,i,g}$ and $v_{j,i,g}$ generates a new trial vector element $u_{j,i,g}$, which is realized by the following equation:

$$\mathrm{u}_{i,g} = \left(u_{j,i,g}\right) = \begin{cases} \left(v_{j,i,g}\right) & \text{if } (rand_j(0,1) \le C_r \\ \left(x_{j,i,g}\right) & \text{otherwise.} \end{cases} \quad (3)$$

where $C_r \in [0, 1]$ is called the crossover rate, which is a user-defined value that controls the fraction of elements copied from the mutant. $rand_j$ (0, 1) generates a random value between 0 and 1 for the j-th element. The algorithm also ensures $u_{j,i,g}$ gets at least one element value as $x_{j,i,g}$. Then the population is updated by comparing each trial vector $\mathrm{u}_{i,g}$ to the corresponding target vector $\mathbf{x}_{i,g}$. If the fitness function value of the trial vector is lower than that of the target vector, replace the target vector with the trial vector in the next generation; otherwise the target vector retains its place in the population. The selection operation is therefore realized by the following equation:

$$\mathrm{x}_{i,g+1} = \begin{cases} \mathrm{u}_{i,g} & \text{if } f\left(\mathrm{u}_{i,g}\right) \le f\left(\mathrm{x}_{i,g}\right) \\ \mathrm{x}_{i,g} & \text{otherwise.} \end{cases} \quad (4)$$

where $f(\cdot)$ is the fitness function. Because of this selection operation, DE is expected to have high optimization ability. When the condition to stop further evolution is satisfied; for example, a preset maximum number of iteration has been reached, the algorithm ends with the best solution as the final solution.

2.2. Differential Evolution with Double Wavelet Mutation (DWM-DE)

In the SDE mutation operation, the value of F in (2) is a fixed value within the range of [0, 1] determined based on the kind of application. The choice of this value relies very much on experience or expert knowledge. Yet, a fixed value of F takes no advantage of the benefit brought by the evolution. We propose the value of F to diminish with the increase of the number of iteration. Moreover, for some complex optimization problem such as finding the minimum point of a multimodal function with many local minima, a large number of iteration for solving the problem is required. It reduces the efficiency of the SDE. This leads to the proposed DWM-DE in which the value F is determined by a wavelet function. The degree of different movements for the trial vectors will then be increased. More "random" directions for the exploration would be generated during the mutation operation. Moreover, in the crossover operation, we proposed a second wavelet mutation that varies the searching space based on the wavelet function. As the wavelet function output is inversely proportional to the number of iteration; when the searching population is approaching the optimal solution, the effect of the double wavelet mutations will be decreasing until the DE ends eventually. By adopting this method, the effort on searching and evaluating those local optima, which could be far away from the global optimum, in the later iteration is reduced. The total number of iteration should also decrease. Thanks to the property of the wavelet function, the solution stability is enhanced in a statistical sense, *i.e*. the performance of the DE on converging to the optimal point is relatively stable despite the presence of many random factors during the evolution.

2.3. Double Wavelet Mutation

2.3.1. Wavelet Theory

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

Property 1:

$$\int_{-\infty}^{+\infty} \psi(x)\, \mathrm{d}x = 0 \quad (5)$$

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

Property 2:

$$\int_{-\infty}^{+\infty} \left|\psi(x)\right|^2 \mathrm{d}x < \infty \quad (6)$$

which means most of the energy in $\psi(x)$ is confined to a finite duration and bounded. The Morlet wavelet, as shown in **Figure 3**, is an example mother wavelet:

$$\psi\left(x\right) = e^{-x^2/2} \cos\left(5x\right) \quad (7)$$

The Morlet wavelet integrates to zero (Property 1). Over 99% of the total energy of the function is contained in the interval of $-2.5 < x < 2.5$ (Property 2). In order to control the magnitude of $\psi(x)$, a function $\psi_a(x)$ is defined as follows.

$$\psi_a(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x}{a}\right) \quad (8)$$

where a is the dilation parameter.

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

2.3.2. Operation of DE with Wavelet Mutation

The vectors in the population are mutated based on a proposed wavelet mutation (WM) operation, which exhibits a fine-tuning property. First, modify the mutation operation (2) as follows.

$$\mathrm{v}_{i,g} = \mathrm{x}_{i,g} + F \cdot \left(\left|\mathrm{x}_{r_1,g} - \mathrm{x}_{r_2,g}\right|\right) \quad (9)$$

where

$$F = \psi_a(\varphi) \quad (10)$$

$$F = \frac{1}{\sqrt{a}} \psi\left(\frac{\varphi}{a}\right) \quad (11)$$

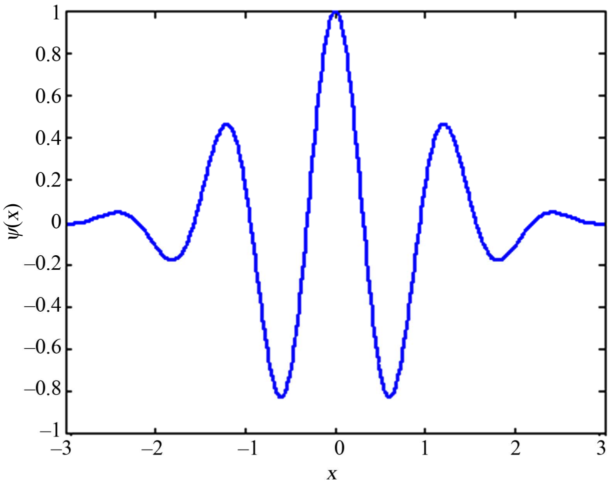

Figure 3. Morlet wavelet.

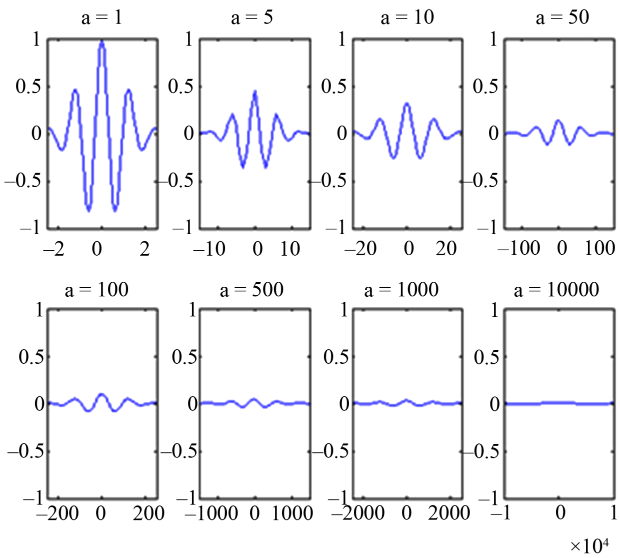

Figure 4. Morlet wavelet dilated by different values of *a* (*x*-axis: *a*, *y*-axis: $\psi_{a,0}(x)$).

By using the Morlet wavelet in (7) as the mother wavelet,

$$F = \frac{1}{\sqrt{a}} e^{-\left(\frac{\varphi}{a}\right)^2 / 2} \cos\left(5\left(\frac{\varphi}{a}\right)\right) \quad (12)$$

Referring to the Property 1 of (5), the total positive momentum of the mother wavelet is equal to the total negative momentum of the mother wavelet. Then, the sum of the positive F is equal to the sum of the negative F when the number of samples is large and φ is randomly generated, *i.e.*

$$\frac{1}{N}\sum_{N} F = 0 \quad \text{for} \quad N \to \infty \quad (13)$$

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

$$a = e^{-\ln(\lambda)\times\left(1-\frac{t}{T}\right)^{\zeta_{wm}} + \ln(\lambda)} \quad (14)$$

where ζ_{wm} is the shape parameter of the monotonic increasing function, λ is the upper limit of the parameter a.

The effects of the various values of the shape parameter ζ_{wm} to a with respect to t/T are shown in **Figure 5**. In this figure, λ is set as 10,000. Thus, the value of a is between 1 and 10,000. Referring to (12), the maximum value of F is 1 when the random number of $\varphi = 0$ and $a = 1$ (at $t/T = 0$). Then referring to (9), the vector $v_{i,g}$ has a large degree of mutation. It ensures that a large search space for the mutated vector is given at the early stage of evolution. When the value t/T is near to 1, the value of a is so large that the maximum value of F will become very small. For example, at $t/T = 0.9$ and $\zeta_{wm} = 1$, $a = 400$; if the random value of φ is zero, the value of F will be equal to 0.0158. A smaller searching space for the mutated vector is then given for fine-tuning.

2.3.3. Operation of DE Crossover with Wavelet Mutation

The crossover operation of (3) is done with respect to the

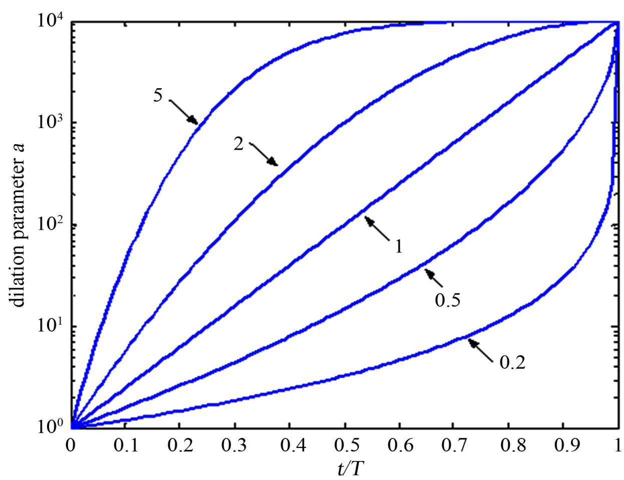

Figure 5. Effect of the shape parameter ξ_{wm} to *a* with respect to t/T.

elements of the trial vector (after mutation) in DE. In DWM-DE, the second-stage wavelet mutation is embedded in the crossover operation. In general, various methods like uniform mutation or non-uniform mutation [17,18] can be employed to realize the mutation operation. In this paper, the second-stage wavelet operation, which exhibits a fine-tuning ability, is realized by adding a second wavelet mutation following the original crossover operation. The crossover after the first mutation takes place according to (3). Let $\mathrm{u}_{i,g} = \left(u_{0,i,g}, u_{1,i,g}, \cdots, u_{D-1,i,g}\right)$ (where g is the current generation number and D is the number of elements in the vector) be the i-th vector after crossover for the second wavelet mutation. The value of the element $u_{j,i,g}$ is inside the vector element's boundary $\left[para_{\min}^{j}, para_{\max}^{j}\right]$. The mutated crossover vector is given by $\bar{\mathrm{u}}_{i,g} = \left(\bar{u}_{0,i,g}, \bar{u}_{1,i,g}, \cdots, \bar{u}_{D-1,i,g}\right)$, and

$$\bar{u}_{j,i,g} = \begin{cases} u_{j,i,g} + \sigma \times \left(para_{\max}^{j} - u_{j,i,g}\right) & \text{if } \sigma > 0 \\ u_{j,i,g} + \sigma \times \left(u_{j,i,g} - para_{\min}^{j}\right) & \text{if } \sigma \le 0 \end{cases} \tag{15}$$

$$\sigma = \psi_a(\varphi) = \frac{1}{\sqrt{a}} \psi\left(\frac{\varphi}{a}\right) \tag{16}$$

where the same Morlet wavelet in (7) is used as the mother wavelet and the value of a is governed by (14). Similar to F of (12), a larger value of $|\sigma|$ at the early stage of evolution gives a larger searching space for the solution; when $|\sigma|$ is small at the later stage of evolution, the algorithm gives a smaller searching space for fine-tuning.

After the operations of the double wavelet mutation, the population is updated by comparing each trial vector $\bar{\mathrm{u}}_{i,g}$ to the corresponding target vector $\mathbf{x}_{i,g}$ using the method of standard DE as given by (4). A new population is generated and the same evolution process is repeated. Such an iterative process will be terminated when a defined number of iteration has been met.

3. Benchmark Test Functions and Results

3.1. Benchmark Test Functions

A suite of eighteen benchmark test functions [19-22] are used to test the performance of the proposed DWM-DE. Many different kinds of optimization problems are covered by these benchmark test functions. They can be divided into three categories. The first one is the category of unimodal functions, which involves a symmetric model with a single minimum; functions f_1-f_7 are unimodal functions. The second one is the category of multimodal functions with a few local minima; functions f_8-f_{13} belong to this category. The last one is the category of multimodal functions with many local minima; functions f_{14}-f_{18} belong to this category. The expressions of these functions are listed in **Table 1**. (The details about the parameters a, b, and c and the function $u(\cdot)$ for the functions f_8, f_9 and f_{12}-f_{14} are given in [22].)

3.2. Experimental Setup

We evaluate the performance of SDE [23], DE with single wavelet mutation (first stage only, SWM-DE), DE/local-to-best/1 [20], DE/rand/1 with per-vector-dither [24] and the proposed DWM-DE by finding the minimum values of the benchmark test functions. The following simulation conditions are used:

- The shape parameter of the wavelet mutation (ζ_{wm}): It is chosen by trial and error for each function through experiments for good performance. $\zeta_{wm}=1$ is used for all functions (A discussion for the value of ζ_{wm} will be given in Section 3.4).
- The parameter λ for the monotonic increasing function: 10,000.
- Initial population: It is generated uniformly at random.
- Crossover probability constant: $C_r = 0.5$.
- The mutation weight factor (for SDE, DE/local-to-best/1 and DE/rand/1): $F = 0.5$.
- The population size: 30.
- The numbers of iteration for all algorithms are listed in **Table 2**.

3.3. Results and Analysis

The simulation results for the 18 benchmark test functions are given to show the merits of the proposed DWM-DE. All results shown are averaged data out of 50 trials.

3.3.1. Unimodal Functions

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 DWM-DE are much better than those of the other methods as shown in **Table 3**. Also, the standard deviation is small, which means that the searched solutions are stable. In **Figure 6(a)**, the DWM-DE returns a faster convergence rate than other methods thanks to its better searching ability.

Function f_2 is the Generalized Rosenbrock's function, which is also called the Banana function. The global minimum of this function is inside a long, narrow, parabolic shaped flat valley. Owing to the smooth and symmetric characteristic of f_2, the main purpose of testing is to measure the convergence rate of the searching algorithms. The result is shown in **Figure 6(b)**. The convergence rate of the proposed DWM-DE is the highest. When using the proposed DWM-DE, the solution quality is increased when the number of iteration increases. As there is only one minimum within the solution space,

Table 1. Benchmark test functions.

Test function	Domain	Optimal point
$f_1(\mathrm{x})=\sum_{i=1}^{30}x_i^2$	$-50\le x_i\le 150$	$f_1(0)=0$
$f_2(\mathrm{x})=\sum_{i=1}^{29}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	$-2.048\le x_i\le 2.048$	$f_2(1)=0$
$f_3(\mathrm{x})=\sum_{i=1}^{30}\left(\lfloor x_i+0.5\rfloor\right)^2$	$-5\le x_i\le 10$	$f_3(0)=0$
$f_4(\mathrm{x})=\sum_{i=1}^{30}x_i^4+random[0,\ 1]$	$-1.28\le x_i\le 2.56$	$f_4(0)=0$
$f_5(\mathrm{x})=\max_i\left\{\lvert x_i\rvert,\ 1\le i\le 30\right\}$	$-150\le x_i\le 50$	$f_5(0)=0$
$f_6(\mathrm{x})=\sum_{i=1}^{30}\lvert x_i\rvert+\prod_{i=1}^{30}\lvert x_i\rvert$	$-5\le x_i\le 15$	$f_6(0)=0$
$f_7(\mathrm{x})=-\cos(x_1)\cdot\cos(x_2)\cdot\exp\left(-\left((x_1-\pi)^2+(x_2-\pi)^2\right)\right)$	$-300\le x_1,x_2\le 300$	$f_7([\pi,\pi])=-1$
$f_8(\mathrm{x})=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]$	$-65.536\le x_i\le 65536$	$f_8([-32,-32])\approx 1$
$f_9(\mathrm{x})=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	$-5\le x_i\le 5$	$f_9([0.1928,\ 0.1928,\ 0.1231,\ 0.1358])\approx 0.0003075$
$f_{10}(\mathrm{x})=-\frac{\sin(x_1)\sin(x_2)}{x_1x_2}$	$-5\le x_1,x_2\le 15$	$\lim_{\mathrm{x}\to[0,0]}f_{10}(\mathrm{x})=-1$
$f_{11}(\mathrm{x})=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^7$	$-5\le x_1,x_2\le 5$	$f_{11}([-0.08983,0.7126])\approx -1.0316$
$f_{12}(\mathrm{x})=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right]$	$0\le x_i\le 1$	$f_{12}([0.114,0.556,0.853])\approx -3.8628$
$f_{13}(\mathrm{x})=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right]$	$0\le x_i\le 1$	$f_{13}([0.201,0.15,0.477,0.275,0.311,0.627])\approx -3.32$
$f_{14}(\mathrm{x})=0.1\left\{\begin{array}{l}\sin^2(\pi 3x_1)\\+\sum_{i=1}^{29}(x_i-1)^2\cdot\left[1+\sin^2(3\pi x_{i+1})\right]\\+(x_{30}-1)^2\left[1+\sin^2(2\pi x_{30})\right]\end{array}\right\}+\sum_{i=1}^{30}u(x_i,5,100,4)$	$-50\le x_i\le 50$	$f_{14}(1)=0$
$f_{15}(\mathrm{x})=\sum_{i=1}^{30}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	$-5.12\le x_i\le 10.24$	$f_{15}(0)=0$
$f_{16}(\mathrm{x})=\frac{1}{4000}\sum_{i=1}^{30}x_i^2-\prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$-1200\le x_i\le 600$	$f_{16}(0)=0$
$f_{17}(\mathrm{x})=-20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30}x_i^2}\right)-\exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos 2\pi x_i\right)+20+e$	$-64\le x_i\le 32$	$f_{17}(0)=0$
$f_{18}(\mathrm{x})=\sum_{i=1}^{30}\left(x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	$-500\le x_i\le 500$	$f_{18}([420.9687,\cdots,420.9687])=-12569.5$

Table 2. Number of iteration in the experiments.

Test Function	No. of Iteration
f_1. Sphere model	300
f_2. Generalized Rosenbrock's function	500
f_3. Step function	100
f_4. Quartic function	200
f_5. Schwefel's problem 2.21	500
f_6. Schwefel's problem 2.22	200
f_7. Easom's function	200
f_8. Shekel's foxholes function	50
f_9. Kowalik's function	100
f_{10}. Maxican hat function	50
f_{11}. Six-hump camel back function	50
f_{12}. Hartman's family 1	50
f_{13}. Hartman's family 2	100
f_{14}. Generalized penalized's function	200
f_{15}. Generalized Rastrigin's function	1000
f_{16}. Generalized Griewank's function	200
f_{17}. Ackley's function	500
f_{18}. Schwefel's function	500

Table 3. Comparison between different de methods for benchmark test functions (category 1). All results are averaged ones over 50 runs.

		DWM-DE	SW-DE	SDE	DE/local-to-best/1	DE/rand/1with per-vector-dither
f_1	Mean	**0.5902**	33.9672	0.9937	228.8271	411.8185
	Best	**0.0605**	0.668	0.4317	17.3954	206.3942
	Std Dev	**0.5712**	75.9703	0.308	213.2461	99.2895
f_2	Mean	**0.0961**	51.5008	25.3632	40.3851	30.2437
	Best	**0.0068**	22.9713	23.7534	26.9933	27.3151
	Std Dev	**0.0867**	30.3582	0.6442	15.6161	3.9049
f_3	Mean	**0**	0.78	11.24	4.7	51.54
	Best	**0**	0	7	1	35
	Std Dev	**0**	0.9957	1.9119	3.4241	8.1346
f_4	Mean	**0.0385**	0.2103	0.2307	0.5176	4.2939
	Best	**0.0172**	0.0366	0.1033	0.0798	1.7894
	Std Dev	**0.0111**	0.2053	0.0684	0.3172	1.3556
f_5	Mean	**1.4127**	33.9335	5.5862	44.3662	22.7523
	Best	**0.7089**	16.0324	3.7069	21.1518	19.1513
	Std Dev	**0.3512**	9.5589	2.9702	9.0817	2.05
f_6	Mean	**0.388**	0.7726	3.3391	4.3577	27.5979
	Best	**0.1151**	0.182	2.3578	0.1878	20.7384
	Std Dev	**0.187**	0.7565	0.5352	3.6601	3.2958
f_7	Mean	**–1**	–0.9455	–0.9284	–0.66	–0.4641
	Best	–1	–1	–1	–1	–1
	Std Dev	**0**	0.2131	0.2488	0.4785	0.4454

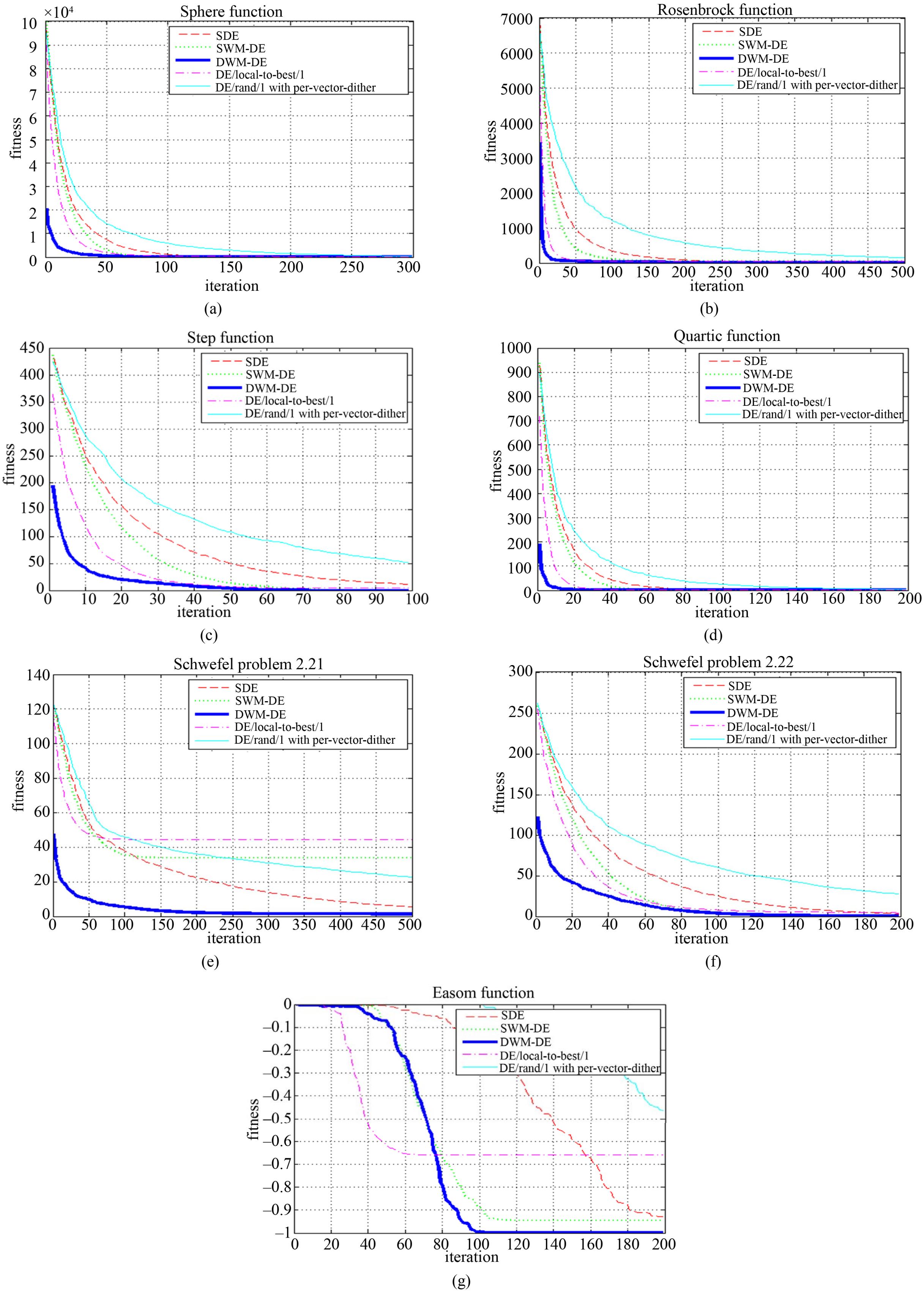

Figure 6. Unimodal functions.

nearly all the population will move towards that minimum. Yet, The DWM-DE performs better than the other methods in terms of the mean value and the standard deviation as shown in **Table 3**.

Function f_3 is the Step function with many flat surfaces. Flat surfaces are obstacles for optimization algorithms because they do not give any information about the search direction. Unless the algorithm has a variable step size, it can get stuck in one of the flat surfaces. All hybrid DEs that involve the mutation operation are good for this function as shown in **Figure 6(c)** because it can generate a long jump by using mutation operations during the evolution.

Function f_4 is the Quartic function. Since it is a polynomial of even degree, it approaches the same limit when the argument goes to positive or negative infinity. Thus the function has a global minimum. The results are shown in **Figure 6(d)** and **Table 3**. We can see that the convergence rate of the proposed DWM-DE is much greater than that of SDE. After around 10 times of iteration, the proposed method is able to reach the minimum.

Functions f_5 and f_6 are the Schwefel's problem 2.21 and Schwefel's problem 2.22. The best value, mean cost value and the standard derivation of the DWM-DE are the best as shown in **Table 3**. Thus, the proposed algorithm gives better solution quality and stability.

Function f_7 is the Easom function where the global minimum is near a small area relative to the search space. The function was inverted for minimization. The result is shown in **Figure 6(g)**. For this function, the convergence rate of the proposed DWM-DE is high. While the convergence rate of DWM-DE is nearly the same as DE with single wavelet mutation only, the solution quality on using DWM-DE is better than the other algorithms during the early evolution. The proposed algorithm gives better performance in terms of convergence rate, solution quality and stability as shown in **Table 3**.

For unimodal functions, the proposed DWM-DE can offer a higher rate of convergence. By adopting the Morlet wavelet on controlling the scaling factor F, the degree of freedom of the trial vectors can be increased. More directions of exploration would be generated during the mutation operation. Moreover, based on the fine-tuning ability of the wavelet operations, the population can easily reach the small region around the global minimum. In short, the DWM-DE is the best to tackle unimodal functions when compared with the other methods.

3.3.2. Multimodal Functions with a Few Local Minima

Six multimodal functions with a few local minima are used to evaluate the five algorithms. Those functions are Shekel's foxholes function, Kowalik's function, Maxican hat function, Six-hump camel back function, Hartman's family 1 and Hartman's family 2. All of them contain some local minima within the searching space. The experimental results for these functions are listed in **Table 4** and shown in **Figure 7**. For all the functions, it is found that all the searching methods perform similarly in reaching the optimal point. While the functions contain a few local minima, all the searching methods do not get trapped in the local minima easily. The advantage brought by the double wavelet mutation operations to the searching is not obvious for these functions. Yet, the solution quality and stability offered by DWM-DE are good.

Table 4. Comparison between different de methods for benchmark test functions (category 2). All results are averaged ones over 50 runs.

		DWM-DE	SW-DE	SDE	DE/local-to-best/1	DE/rand/1with per-vector-dither
f_8	Mean	0.998	0.998	0.998	0.998	0.998
	Best	0.998	0.998	0.998	0.998	0.998
	Std Dev	0	0	0	0	0
f_9	Mean	**0.0009**	0.0012	0.0011	0.0015	0.0016
	Best	0.0005	0.0004	0.0007	<u>0.0003</u>	0.0007
	Std Dev	**0.0003**	0.0011	0.0008	0.0039	0.0018
f_{10}	Mean	–1	–1	–1	–1	–1
	Best	–1	–1	–1	–1	–1
	Std Dev	0	0	0	0	0
f_{11}	Mean	–1.0316	–1.0316	–1.0316	–1.0316	–1.0315
	Best	–1.0316	–1.0316	–1.0316	–1.0316	–1.0316
	Std Dev	0	0	0.0001	0	0.0002
f_{12}	Mean	–3.8628	–3.8628	–3.8628	–3.8628	–3.8628
	Best	–3.8628	–3.8628	–3.8628	–3.8628	–3.8628
	Std Dev	0	0	0	0	0
f_{13}	Mean	**–3.3124**	–3.3036	–3.2909	–3.2911	–3.2777
	Best	–3.322	–3.322	–3.322	–3.322	–3.322
	Std Dev	**0.0303**	0.0408	0.0475	0.0527	0.0448

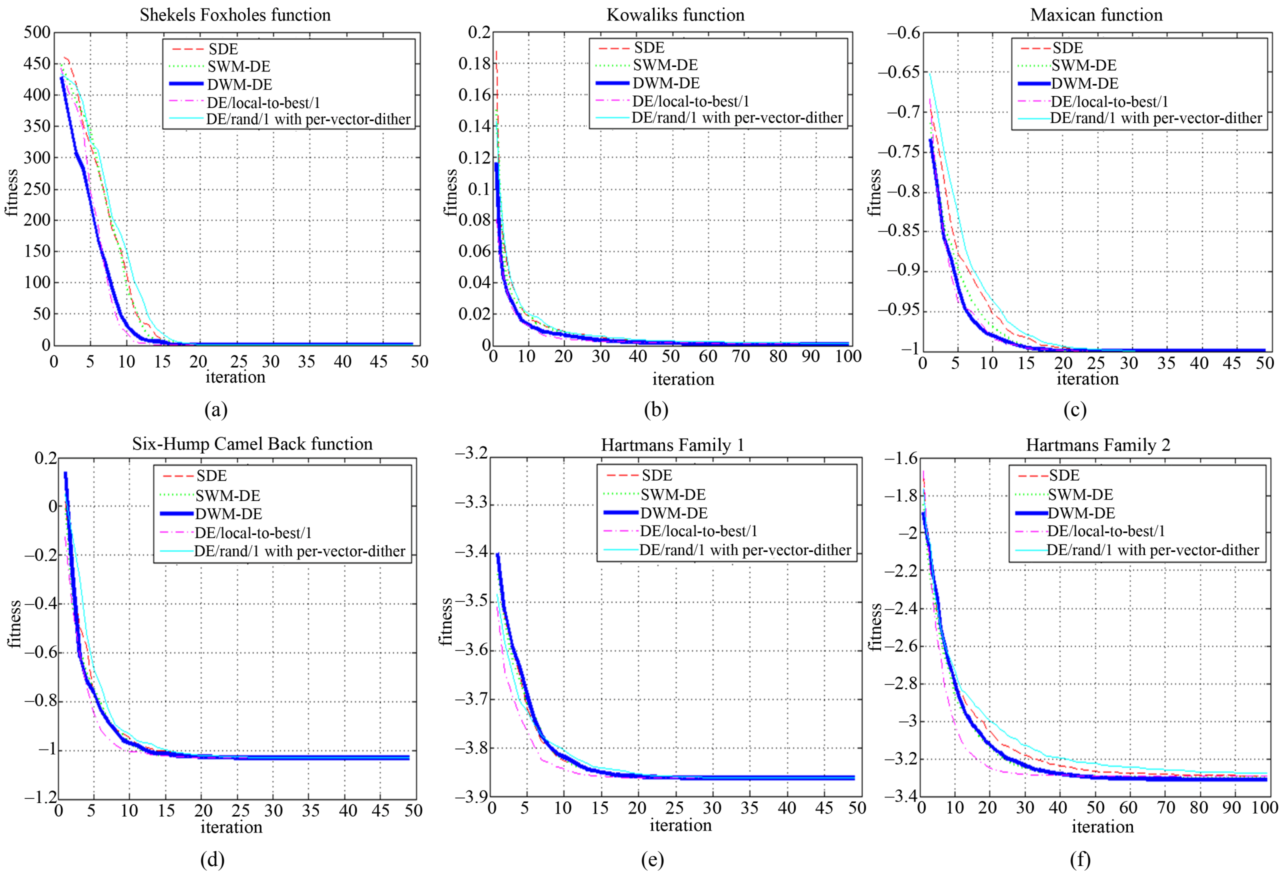

Figure 7. Multimodal functions with a few local minma.

3.3.3. Multimodal Functions with Many Local Minima

Functions f_{14}-f_{18} are multimodal functions with many local minima. The experimental results for these functions are listed in **Table 5** and shown in **Figure 8**. Function f_{14}, f_{15} and f_{16} are the Generalized penalized's function, Generalized Rastrigin's function and Generalized Griewank's function respectively. They are widely used as test functions for global optimization. Those functions have an exponentially increasing number of local minima as their dimension increases, and the locations of the minima are regularly distributed. In the experiment, the dimension is 30. As a result, the function contains plenty of local minima. From **Figure 8**, we can see that if the wavelet mutation is used, the rate of convergence is much improved. By adding the double wavelet mutation operations to the DE, we can reduce the chance that the searching process is trapped in some local minima. Moreover, by introducing the second wavelet mutation to DE, the searching process of DWM-DE is capable of moving closely to the global minimum in the early iteration stage. Thanks to the property of the second wavelet mutation, the effort on searching and evaluating those local minima that are far away from the global minimum is reduced.

Function f_{17} is the Generalized Ackley's function, which is a continuous multimodal function obtained by modulating an exponential function with a cosine wave of moderate amplitude. Its topology is characterized by an almost flat outer region and a central hole or peak where the modulation of the cosine wave becomes more and more influential. The result is shown in **Figure 8(d)**. It shows that if the double wavelet mutation operations are used with DE, the fitness of the function dropped rapidly. After 250 times of iteration, the fitness becomes near the global minimum. However, even DE with single wavelet mutation cannot satisfactorily reach the global minimum. It shows the advantage the double wavelet mutation operations on reducing the effort for searching and evaluating those local minima that are far away from the global minimum.

Function f_{18} is the Schwefel's function, which is deceptive in that the global minimum is geometrically distant from the next best local minima. Therefore, the search algorithms are potentially prone to converge to the wrong direction. The result is shown in **Figure 8(e)**. Similar to functions f_{16} and f_{17}, if the double wavelet mutation operations are used, the convergence rate is much improved. The DWM-DE can move closely to the global minimum at the early iteration stage.

Table 5. Comparison between different de methods for benchmark test functions (category 3). All results are averaged ones over 50 runs.

		DWM-DE	SW-DE	SDE	DE/local-to-best/1	DE/rand/1with per-vector-dither
	Mean	**0.4883**	67682.17	207.3875	147.6601	6835946
f_{14}	Best	**0.1582**	8.0881	22.4585	4.7027	149.9788
	Std Dev	**0.2989**	289798.5	291.9504	391.5459	4501213
	Mean	**8.3213**	9.6949	124.7772	34.9675	160.9108
f_{15}	Best	**0.0063**	3.3497	87.5478	16.9349	136.2236
	Std Dev	4.4707	**3.155**	10.5849	11.9889	10.8145
	Mean	**1.0243**	2.0471	2.2483	3.0293	54.1213
f_{16}	Best	**0.8751**	1.1207	1.5661	1.0394	33.3442
	Std Dev	**0.0576**	1.1589	0.4224	1.9004	10.991
	Mean	**0.3199**	3.983	0.7723	4.6762	17.728
f_{17}	Best	**0.0271**	0.2236	0.0261	2.2245	7.1453
	Std Dev	**0.3194**	3.5579	2.9323	1.5461	3.687
	Mean	**–12568.8**	–12254.1	–12475.8	–10328.1	–10128.1
f_{18}	Best	**–12569.5**	–12563.5	–12569	–11004.5	–11272.4
	Std Dev	**0.6883**	181.3254	145.1123	391.6365	560.2403

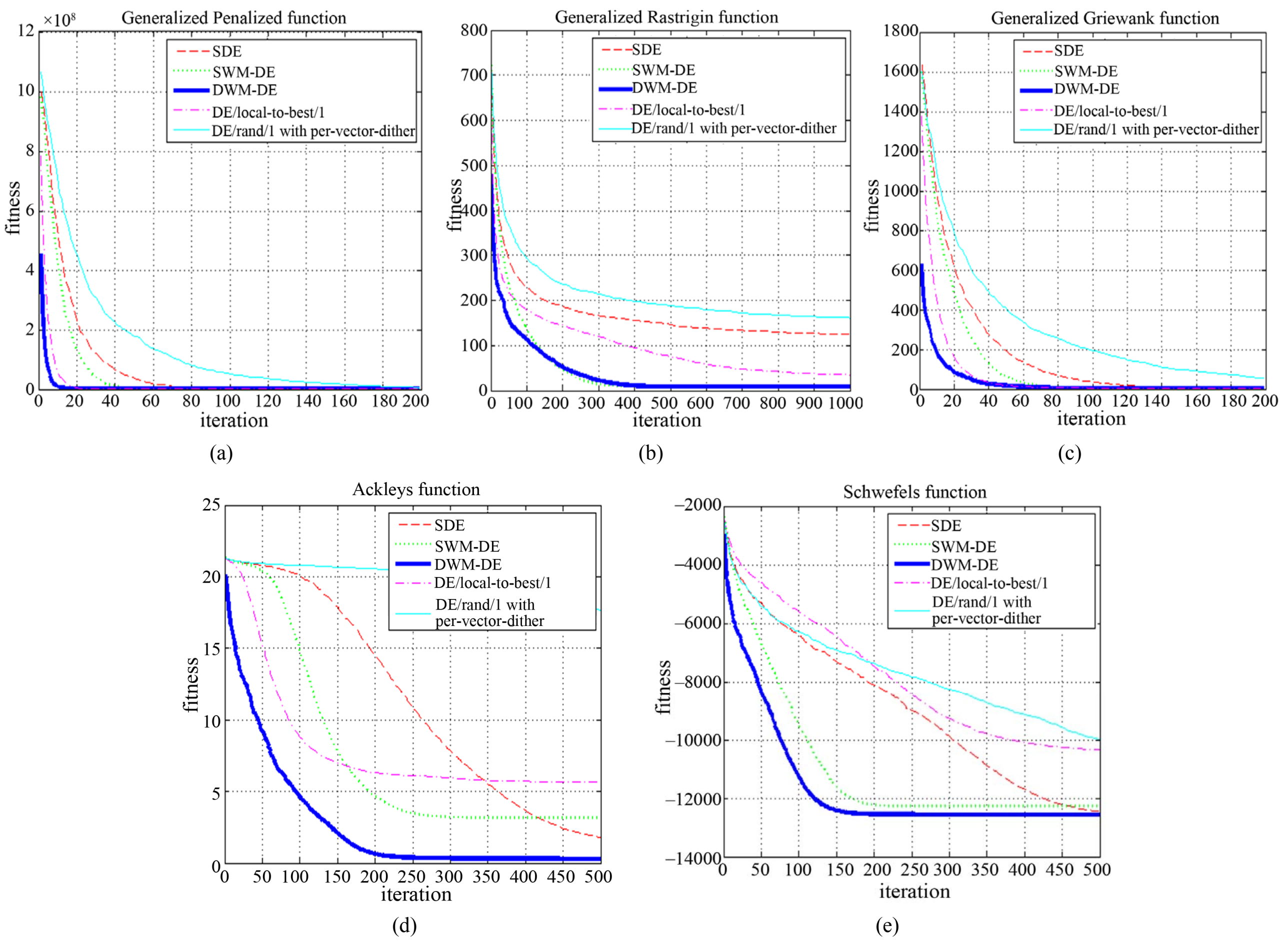

Figure 8. Multimodal functions with many local minima.

For multimodal functions with many local minima, the proposed DWM-DE can significantly improve the convergence rate and the chance of reaching the global optimum as compared to the other algorithms.

3.4. The *t*-Test

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

$$t = \frac{\overline{\alpha_2} - \overline{\alpha_1}}{\sqrt{\left(\frac{\sigma_2^2}{\xi + 1}\right) + \left(\frac{\sigma_1^2}{\xi + 1}\right)}} \quad (17)$$

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 degree of freedom. When the *t*-value is higher than 1.645 (with ξ = 49 for 50 runs), there is a significant difference between the two algorithms with a 95% confidence level. The *t*-values between the DWM-DE and other optimization algorithms are shown in **Table 6**. The N/A in the table means the result of *t*-test is undefined. We see that most of the *t*-values in this table are higher than 1.645. Therefore, the performance of the DWM-DE is significantly better than that of other optimization algorithms with a 95% confidence level.

3.5. Sensitivity of the Shape Parameter for the WM

The mean cost values offered by the DWM-DE under different values of the shape parameter ζ_{wm} for all test functions in Section 3.1 are listed in **Table 7**. The functions are tested by using ζ_{wm} = 0.2, 0.5, 1, 2, and 5. In this experiment, the parameter λ 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 DWM-DE needs to perform the fine-tuning faster, a larger ζ_{wm} should be used. In general, if the function is smooth and symmetric, the searching algorithms should be fast to jump to the area near the global optimum and then perform the fine-tuning. Therefore, a smaller ζ_{wm} can be set (ζ_{wm} = 0.2) so that the DWM-DE will perform more significant mutation in order to increase its speed of convergence. In some cases, the value of ζ_{wm} is not very critical, e.g. in f_7, f_8 and f_9. For f_7, the mean cost values under different values of ζ_{wm} are nearly the same. However, in some cases, the value of the parameter ζ_{wm} is sensitive to the performance of the searching, e.g. in f_2 and f_5. In conclusion, no formal method is available to choose the value of the parameter ζ_{wm}; it depends on the chara- cteristics of the optimization problems.

Table 6. *t*-value between dwm-de and other de methods.

Functions	*t*-value between DWM-DE and SWM-DE	*t*-value between DWM-DE and SDE	*t*-value between DWM-DE and DE/local-to-best/1	*t*-value between DWM-DE and DE/r and/1 with per-vector-dither
f_1	3.106535	4.396617	7.568123	29.28583
f_2	11.97319	274.8664	18.24283	54.5784
f_3	5.539252	41.57059	9.705914	44.80157
f_4	5.90861	19.61271	10.67363	22.19623
f_5	24.0406	9.866993	33.41888	72.54976
f_6	3.489848	36.80786	7.659204	58.28451
f_7	1.808415	2.034921	5.024374	8.507825
f_8	N/A	N/A	N/A	N/A
f_9	1.860521	1.655212	1.084652	2.712445
f_{10}	N/A	N/A	N/A	N/A
f_{11}	N/A	0	N/A	3.535534
f_{12}	N/A	N/A	N/A	N/A
f_{13}	1.224414	2.69834	2.477622	4.536719
f_{14}	1.651429	5.011117	2.657828	10.73876
f_{15}	4.521753	85.52812	39.58508	109.6375
f_{16}	6.232955	20.30214	7.456842	34.15952
f_{17}	7.250986	1.084521	19.51148	33.26134
f_{18}	11.0344	8.376944	38.8019	34.21467

Table 7. Sensitivity of the shape parameter ζ_{wm} for wavelet mutation.

Functions	0.2	0.5	1	2	5
f_1	**0.1022**	0.4705	0.5902	1.4617	0.6216
f_2	**0.0065**	0.0144	0.0961	0.3188	16.7615
f_3	**0**	**0**	**0**	0.12	1.76
f_4	**0.0329**	0.033	0.0385	0.044	0.0582
f_5	**0.4248**	0.7679	1.4127	2.771	4.5791
f_6	**0.3018**	0.5405	0.388	0.937	0.7229
f_7	–1	–1	–1	–0.9999	–0.9798
f_8	**0.998**	**0.998**	**0.998**	**0.998**	7.2744
f_9	0.0014	0.001	**0.0009**	0.0013	0.0016
f_{10}	**–1**	**–1**	**–1**	**–1**	**–1**
f_{11}	**–1.0316**	**–1.0316**	**–1.0316**	**–1.0316**	**–1.0316**
f_{12}	–3.8628	–3.8627	**–3.8628**	–3.8623	–3.862
f_{13}	–3.3121	–3.312	–3.3124	**–3.3186**	–3.3059
f_{14}	**0.1728**	0.4728	0.4883	1.308	7.9534
f_{15}	5.4248	**4.9128**	8.3213	10.4683	10.9009
f_{16}	0.9953	1.0694	1.0243	0.9733	**0.9356**
f_{17}	**0.0457**	0.2032	0.3199	1.2207	1.7944
f_{18}	**–12569.4**	–12569.1	–12568.8	–12567.5	–12554.9

3.6. Sensitivity of the Parameter λ for the WM

The mean cost values offered by the DWM-DE with different values of the WM's parameter λ for all test functions are tabulated in **Table 8**. The functions are tested by using λ = 100, 1000, 10,000, and 100,000. In this experiment, the parameter ζ_{wm} is fixed at 1. If we want a smaller value of the upper limit (the searching limit) of the particle's mutated element, a larger value of λ should be used. In some cases, the parameter λ is not very sensitive, such as f_3-f_4, f_7-f_{13}, f_{16} and f_{18}. The mean cost values under different values of λ have no significant difference. However, in some cases, such as f_1, the value of the parameter λ is sensitive to the performance of the DWM-DE. In f_1, the mean cost value is 0.3202 when λ = 100, and the mean cost value is 1.4945 when λ = 100,000. Their difference is around 5 times. In conclusion, similar to the parameter ζ_{wm}, no formal method is available to choose the value of the parameter λ. It depends on the characteristics of the optimization problem. Comparing with the sensitivity of the shape parameter ζ_{wm}, the parameter λ is less sensitive to the performance of the searching.

Table 8. Sensitivity of the parameter λ for wavelet mutation.

Functions	λ = 100	λ = 1000	λ = 10,000	λ = 100,000
f_1	**0.3202**	0.8152	0.5902	1.4945
f_2	**0.0131**	0.0357	0.0961	0.0359
f_3	**0**	0.02	**0**	0.02
f_4	0.0405	0.0395	**0.0385**	0.0395
f_5	**0.6381**	1.0508	1.4127	1.5449
f_6	0.4413	0.7524	**0.388**	0.8447
f_7	**–1**	–0.9997	**–1**	**–1**
f_8	0.998	0.998	0.998	0.998
f_9	0.0011	0.001	**0.0009**	0.0011
f_{10}	–1	–1	–1	–1
f_{11}	–1.0316	–1.0316	–1.0316	–1.0316
f_{12}	**–3.8628**	–3.8619	**–3.8628**	–3.8623
f_{13}	–3.3122	**–3.3145**	–3.3124	–3.3211
f_{14}	**0.3142**	0.6711	0.4883	0.9591
f_{15}	**6.7112**	7.7008	8.3213	8.3281
f_{16}	1.0369	1.0698	**1.0243**	1.0726
f_{17}	**0.2116**	0.3268	0.3199	0.4361
f_{18}	**–12569.1056**	–12568.6379	–12568.8411	–12568.2966

4. The Economic Load Dispatch with Valve-Point Loading Problem

The Economic Load Dispatch with Valve-Point Loading (ELD-VLP) is a method to control or schedule a group of power generator outputs with respect to the load demands, and operate a power system economically so as to minimize the operation cost of the power system. Because of the valve-point loadings and rate limits, the input-output characteristics of modern generators are nonlinear by nature. As a result, the characteristics of ELD-VPL problems are multimodal, discontinuous, and highly nonlinear. In this paper, the DWM-DE is employed to solve the ELD-VPL problem, which aims at minimizing the following objective function:

$$\sum_{i=1}^{n} C_i\left(P_{L_i}\right) \quad (18)$$

where $C_i(P_{Li})$ 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.

$$D_L = \sum_{i=1}^{n} P_{L_i} - P_{Loss} \quad (19)$$

$$P_{L_{i,min}} \le P_{L_i} \le P_{L_{i,\max}} \ ; i = 1, 2, \cdots, n \quad (20)$$

where D_L is the load demand, P_{Li} is the output power of the i-th 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 i-th generator, respectively. The operation fuel cost function is given by

$$C_i\left(P_{L_i}\right) = a_i P_{L_i}^2 + b_i P_{L_i} + c_i \quad (21)$$

where a_i, b_i, and c_i are the coefficients of the cost curve of the i-th generator.

To obtain the practical ELD solution, the operation of the ELD problem should be considered with the valve-point effects. Typically, to model the effects of valve points, a rectified sinusoidal term is added to the cost function:

$$C_i\left(P_{L_i}\right) = a_i P_{L_i}^2 + b_i P_{L_i} + c_i + \left| e_i \times \sin\left(f_i \times \left(P_{L_{i,\min}} - P_{L_i} \right)\right)\right| \quad (22)$$

where 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. In practice, the valve-point effects introduce ripples in the heat-rate curves.

To solve the ELD-VPL problem by using DWM-DE, the solution representation of elements in the population is defined as follows:

$$\mathrm{P} = \left[P_{L_1} P_{L_2} P_{L_3} \cdots P_{L_{n-1}} \right] \quad (23)$$

From (16), we have

$$P_{L_n} = D_L - \sum_{i=1}^{n-1} P_{L_i} + P_{Loss} \quad (24)$$

In this paper, the power loss is not considered; therefore $P_{Loss} = 0$, and

$$P_{L_n} = D_L - \sum_{i=1}^{n-1} P_{L_i} \quad (25)$$

Based on the problem defined above, the objective of this optimization problem is to minimize the total fuel cost based on (22) by using DWM-DE.

5. The Experiment and Result of the ELD-VPL Problem

In this paper, ELD-VPL problem with 13 and 40 generators system with a non-smooth fuel cost function with sinusoidal terms are used to test the performance of the proposed DWM-DE method. The results obtained are compared with those reported in the literature. Both the 13- and 40-generator systems have non-convex solution spaces with many local minima. As a result, the global minimum is difficult to determine. For the 13-generator system, the total load demand of 1800 MW is tested. For the 40-generator system, the total load demand of 10,500 MW is tested. The parameters for the 13- and 40-generator systems are shown in **Tables 9** and **10** respectively. By using the WDM-DE, the following simulation conditions are used:

- Shape parameter of the wavelet mutation (ζ_{wm}): 1.
- Parameter λ for the monotonic increasing function: 10,000.
- Initial population: It is generated uniformly at random.
- Crossover probability constant: $C_r = 0.5$.
- Mutation weight factor (For SDE): $F = 0.5$.
- Numbers of iteration: 500.
- Number of population: 50.

All results shown are averaged ones out of 100 trials. The statistical results in terms of the mean cost value, the best cost value, and the standard deviation are shown in **Table 11**. From the result obtained in this experiment, we find that the DWM-DE performs much better than the other DE methods. The DWM-DE can offer the best (minimum) cost. The average cost for the 13-generator system is $17,996.43, and the best (minimum) cost is

Table 9. Parameters for the 13 generators system.

Unit (i)	a_i	b_i	c_i	e_i	f_i	$P_{i,min}$	$P_{i,max}$
1	0.00028	8.10	550	300	0.035	0	680
2	0.00056	8.10	309	200	0.042	0	360
3	0.00056	8.10	307	150	0.042	0	360
4	0.00324	7.74	240	150	0.063	60	180
5	0.00324	7.74	240	150	0.063	60	180
6	0.00324	7.74	240	150	0.063	60	180
7	0.00324	7.74	240	150	0.063	60	180
8	0.00324	7.74	240	150	0.063	60	180
9	0.00324	7.74	240	150	0.063	60	180
10	0.00284	8.60	126	100	0.084	40	120
11	0.00284	8.60	126	100	0.084	40	120
12	0.00284	8.60	126	100	0.084	55	120
13	0.00284	8.60	126	100	0.084	55	120

Table 10. Parameters for the 40 generators system.

Unit (i)	a_i	b_i	c_i	e_i	f_i	$P_{i,min}$	$P_{i,max}$
1	0.00690	6.73	94.705	100	0.084	36	114
2	0.00690	6.73	94.705	100	0.084	36	114
3	0.02028	7.07	309.54	100	0.084	60	120
4	0.00942	8.18	369.03	150	0.063	80	190
5	0.01140	5.35	148.89	120	0.077	47	97
6	0.01142	8.05	222.33	100	0.084	68	140
7	0.00357	8.03	278.71	200	0.042	110	300
8	0.00492	6.99	391.98	200	0.042	135	300
9	0.00573	6.60	455.76	200	0.042	135	300
10	0.00605	12.9	722.82	200	0.042	130	300
11	0.00515	12.9	635.20	200	0.042	94	375
12	0.00569	12.8	654.69	200	0.042	94	375
13	0.00421	12.5	913.40	300	0.035	125	500
14	0.00752	8.84	1760.40	300	0.035	125	500
15	0.00708	9.15	1728.30	300	0.035	125	500
16	0.00708	9.15	1728.30	300	0.035	125	500
17	0.00313	7.97	647.85	300	0.035	220	500
18	0.00313	7.95	649.69	300	0.035	220	500
19	0.00313	7.97	647.83	300	0.035	242	550
20	0.00313	7.97	647.81	300	0.035	242	550
21	0.00298	6.63	785.96	300	0.035	254	550
22	0.00298	6.63	785.96	300	0.035	254	550
23	0.00284	6.66	794.53	300	0.035	254	550
24	0.00284	6.66	794.53	300	0.035	254	550
25	0.00277	7.10	801.32	300	0.035	254	550
26	0.00277	7.10	801.32	300	0.035	254	550
27	0.52124	3.33	1055.10	120	0.077	10	150
28	0.52124	3.33	1055.10	120	0.077	10	150
29	0.52124	3.33	1055.10	120	0.077	10	150
30	0.01140	5.35	148.89	120	0.077	47	97
31	0.00160	6.43	222.92	150	0.063	60	190
32	0.00160	6.43	222.92	150	0.063	60	190
33	0.00160	6.43	222.92	150	0.063	60	190
34	0.00010	8.95	107.87	200	0.042	90	200
35	0.00010	8.62	116.58	200	0.042	90	200
36	0.00010	8.62	116.58	200	0.042	90	200
37	0.01610	5.88	307.45	80	0.098	25	110
38	0.01610	5.88	307.45	80	0.098	25	110
39	0.01610	5.88	307.45	80	0.098	25	110
40	0.00313	7.97	647.83	300	0.035	242	550

Table 11. Result of the ELD-VPL problem.

Number of Generators	Load		DWM-DE	Standard DE	DE/local-to-best/1	DE/rand/1 with per-vector-dither
		Mean	**17996.43**	18185.27	18078.82	18213.61
13	1800 MW	Best	**17972.78**	18104.61	17982.91	18077.96
		Std Dev	**20.85**	51.61	47.95	45.34
		Mean	**121521.79**	121834.62	123363.297947	122490.90
40	10500 MW	Best	**121431.63**	121530.99	121971.298961	122188.14
		Std Dev	**53.27**	172.74	610.10	89.64

$17,972.78. For the 40-generator system, the average cost is $121,521.79, and the best (minimum) cost is $121,431.63. Moreover, the smallest standard deviation is also obtained by using DWM-DE. Thanks to the wavelet properties, the stability of the optimization solution is improved. In the ELD-VPL problem, the solution stability is very important. Since the load demand is changing with time. A stable optimization method can offer a better quality of the power generation service. To conclude, the convergence speed, solution quality and solution stability of the DWM-DE are good. DWM-DE is suitable to be applied to the ELD-VPL problem.

Table 12 summarized the best results obtained by DWM-DE, DEC-SQP [4] and IGA [25,26] for comparison. The result shows that the DEC-SQP performs the best for the 13-generator system. The best cost is $17,938.95, while the best cost of DWM-DE is $17,972.78. Although, the DWM-DE method cannot offer the best result, the result is already very near to the DEC-SQP method. As the dimension of the 13-generator system is relatively small, the wavelet based mutations of DWM-DE might not be able to enhance the searching process very effectively. For the 40-generator system, the best result of DWM-DE, MPSO [2], DEC-SQP [4] and NPSO-LRS [27] are summarized in **Table 13**. Among the 4 methods, DWM-DE offers the best result. The best cost for the 40-generator system is $121,431.63. Since the dimension of the 40-generator system is large, the wavelet based mutations of DWM-DE can enhance the searching process effectively and it does not trap into some local minimum easily. For the ELD-VPL problem, to obtain the best result, we suggest applying DWM-DE for the high dimensional cases; for example, a dimension higher than 30.

6. Conclusion

In this paper, we have proposed an improved Differential Evolution (DE) that incorporates double wavelet-based mutations to handle the ELD-VPL problem. In the first mutation operation, a scheme on tuning the scaling factor F of the DE algorithm that applies a wavelet function is proposed. In the crossover operation of DE, a second mutation operation for modifying the trial population vectors that applies a wavelet function is proposed. The resulting DWM-DE takes advantage of the properties of the wavelet function to improve the solution quality and stability. The proposed method can explore the solution space more effectively in reaching the global solution. Simulation results have shown that the proposed double-wavelet-mutation based DE is a useful algorithm to solve a suite of 18 benchmark test functions, and offers better results in terms of convergence rate, solution quality and stability. Moreover, the ELD-VPL problem is solved by the DWM-DE. It is shown empirically that the proposed method out-performs significantly the conventional methods in terms of convergence speed, solution quality and solution stability, especially when the dimension of the problem is high.

Table 12. Comparison with other published results for the 13 generators system (load = 1800 MW).

Number of Generators	Load	DWM-DE	DEC-SQP	IGA
13	1800 MW	17972.78	**17938.95**	18069.40

Table 13. Comparison with other published results for the 40 generators system (load = 10,500 MW).

Number of Generators	Load	DWM-DE	MPSO	DEC-SQP	NPSO-LRS
40	10,500 MW	**121431.63**	122252.26	121741.97	1216664.43

7. Acknowledgements

The work described in this paper was substantially supported by a grant from The Hong Kong Polytechnic University (Project No. RP9L).

REFERENCES

[1] R. J. M. Vaessens, E. H. L. Aarts and J. K. Lenstra, “A Local Search Template,” *Proceedings of Parallel Problem Solving Nature*, Brussels, 28-30 September 1992, pp. 65-74.

[2] J. B. Park, K. S. Lee and K. Y. Lee, “A Particle Swarm Optimization for Economic Dispatch with Non-Smooth Cost Functions,” *IEEE Transactions on Power Systems*, Vol. 20, No. 1, 2005, pp. 34-49. doi:10.1109/TPWRS.2004.831275

[3] T. A. A. Victoire and A. E. Jeyakumar, “Hybrid PSO-SQP for Economic Dispatch with Valve-Point Effect,” *Electrical Power Systems Research*, Vol. 71, No. 1, 2004, pp. 51-59. doi:10.1016/j.epsr.2003.12.017

[4] L. S. Coelho and V. C. Mariani, “Combining of Chaotic Differential Evolution and Quadratic Programming for Economic Dispatch Optimization with Valve-Point Effect,” *IEEE Transactions on Power Systems*, Vol. 21, No. 2, 2006, pp. 989-995.

[5] R. Storn and K. Price, “Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *Journal of Global Optimization*, Vol. 11, No. 4, 1997, pp. 341-359. doi:10.1023/A:1008202821328

[6] L. Fogel, “Evolutionary Programming in Perspective: The Top-Down View,” In: J. M. Zurada, R. J. Marks and C. J. Robinson, Eds., *Computational Intelligence*: *Imitating Life*, IEEE Press, Piscataway, 1994, pp. 135-146.

[7] D. Goldberg, “Genetic Algorithms in Search Optimization and Machine Learning,” Addison-Wesley, Boston, 1989.

[8] X. Yao and Y. Liu, “Evolutionary Programming Made Faster,” *IEEE Transactions Evolutionary Computation*, Vol. 3, No. 2, 1999, pp. 82-102. doi:10.1109/4235.771163

[9] Y. Ao and H. Chi, "Differential Evolution Using Opposite Point for Global Numerical Optimization," *Journal of Intelligent Learning Systems and Applications*, Vol. 4, No. 1, 2012, pp. 1-19. doi:10.4236/jilsa.2012.41001

[10] S. Paterlini and T. Krink, "High Performance Clustering with Differential Evolution," 2004. http://dsp.szu.edu.cn/dsp2006/research/areas/t08/papers/Clustering/03.High%20performance%20clustering%20with%20differential%20evolution.pdf

[11] J. H. van Sickel, K. Y. Lee and J. S. Heo, "Differential Evolution and Its Applications to Power Plant Control," *Proceedings of Intelligent Systems Applications to Power System*, Kaohsiung, 5-8 November 2007, pp. 1-6.

[12] B. Babu and R. Angira, "Optimization of Non-Linear Functions Using Evolutionary Computation," *Proceedings of* 12*th ISME International Conference on Mechanical Engineering*, India, 2001, pp. 153-157.

[13] A. Qing, "Dynamic Differential Evolution Strategy and Applications in Electromagnetic Inverse Scattering Problems," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 44, No. 1, 2006, pp. 116-125. doi:10.1109/TGRS.2005.859347

[14] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," *Proceedings of* 6*th International Symposium on Micro Machine and Human Science*, Nagoya, 4-6 October 1995, pp. 39-43. doi:10.1109/MHS.1995.494215

[15] J. Vesterstroem and R. Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," *Congress on Evolutionary Computation*, Portland, 19-23 June 2004, pp. 1980-1987.

[16] I. Daubechies, "Ten Lectures on Wavelets," Society for Industrial and Applied Mathematics, Philadelphia, 1992. doi:10.1137/1.9781611970104

[17] Z. Michalewicz, "Genetic Algorithm + Data Structures = Evolution Programs," 2nd Edition, Springer-Verlag, New York, 1994.

[18] A. Neubauer, "A Theoretical Analysis of the Non-Uniform Mutation Operator for the Modified Genetic Algorithm," *IEEE International Conference on Evolutionary Computation*, Indianapolis, 13-16 April 1997, pp. 93-96.

[19] J. Brest, S. Greiner, B. Bošković, M. Mernik and V. Žumer, "Self Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 6, 2006, pp. 646-657. doi:10.1109/TEVC.2006.872133

[20] Y. Ao and H. Chi, "Experimental Study on Differential Evolution Strategies," *Proceedings of Global Congress on Intelligent Systems*, Anqing, 19-24 May 2009, pp. 19-24. doi:10.1109/GCIS.2009.31

[21] H. Y. Fan and J. Lampinen, "A Trigonometric Mutation Operation to Differential Evolution," *Journal of Global Optimization*, Vol. 27, No. 1, 2003, pp. 105-129. doi:10.1023/A:1024653025686

[22] M. M. Ali, C. Khompatraporn and Z. B. Zabinsky, "A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems," *Journal of Global Optimization*, Vol. 31, No. 4, 2005, pp. 635-672. doi:10.1007/s10898-004-9972-2

[23] U. K. Chakraborty, "*Advances in Differential Evolution*," Springer, Heidelberg, 2008.

[24] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-Based Differential Evolution," *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 1, 2008, pp. 64-79. doi:10.1109/TEVC.2007.894200

[25] P. H. Chen and H. C. Chang, "Large-Scale Economic Dispatch by Genetic Algorithms," *IEEE Transactions on Power Systems*, Vol. 10, No. 1, 1995, pp. 117-124.

[26] S. H. Ling, H. C. C. Iu, K. Y. Chan, H. K. Lam, C. W. Yeung and F. H. F. Leung, "Hybrid Particle Swarm Optimization with Wavelet Mutation and Its Industrial Applications," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part B*, Vol. 38, No. 3, 2008, pp. 743-763.

[27] A. I. Selvakumar and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Non Convex Economic Dispatch Problems," *IEEE Transactions on Power Systems*, Vol. 22, No. 1, 2007, pp. 42-50. doi:10.1109/TPWRS.2006.889132

