SOLVING TRANSIENT-STABILITY CONSTRAINED OPTIMAL POWER FLOW PROBLEMS WITH WAVELET MUTATION BASED HYBRID PARTICLE SWARM OPTIMIZATION


Abstract The paper extends our previous work on solving multi-contingency transient stability constrained optimal power flow problems (MC-TSCOPF) with the approach of particles swarm optimization (PSO). A hybrid PSO method that incorporates with a new wavelet theory based mutation operation, intends to improve the searching strategies on previously used PSO methods, is proposed to solve MC-TSCOPF problems. It employs wavelet theory in enhancing PSO methods in exploring solution spaces more effectively and robustly in reaching better solutions. A case study on the New England 39-bus system indicates that the proposed hybrid PSO outperforms significantly existing PSO methods in terms of solution quality and stability. As a result, reasonable solutions can be reached with faster convergence speeds and smaller computational efforts.

Key Words, Particles swarm optimization, genetic algorithm, wavelet theory, mutation, multi-contingency transient stability constrained optimal power flow problems.

1. Introduction

It has been recently demonstrated that particles swarm optimization (PSO) can offer a more robust and efficient solver than classical optimization methods [1,2] and evolutionary algorithms [3] on hard MC-TSCOPF problems [4]. MC-TSCOPF aims to achieve an optimal solution of a specific objective function by setting the system control variables while satisfying the system to withstand specified contingencies and reach an acceptable steady-state operating condition [5]. However, based on our observation that the proposed PSO performs well only in early iterations and its improvement decreases gradually along iterations or even terminates in later iterations [4]. They behave like traditional local searching methods that drop into a local optima and cannot escape from it. It is hard to obtain any large improvements by examining neighbouring solutions in later iterations. In [6] and [7], this problem has been put into the context of local search or neighbourhood search.

In order to enhance the exploration on searching spaces, hybrid PSO algorithms have been proposed to incorporate with GAs’ evolutionary operations like crossover, mutation, selection [8-11]. It has been demonstrated in our recent research [12] that the hybrid PSO incorporated with GAs’ mutation with constant mutating space [8] is better than the other existing ones [9-11] in solving MC-TSCOPF problems. In the approach presented in [8], particles can follow different directions by themselves, and local positions of particles can be permuted. As a result, pre-mature convergence to unreasonable solutions is more likely to be avoided. However, in this approach, the mutating space is unchanged throughout the search, so the space of permutation of particles in PSO is also unchanged. The approach could be further improved by varying the mutating space along the search.

In GAs, solution spaces are more likely to be explored in early iterations by setting
larger mutating space, and it is more likely to be fine tuned to better solutions in late iterations by setting smaller mutating space, based on the properties of wavelet. This idea can be applied on the hybrid PSO with GAs’ mutation. In this paper, a dynamic mutating space is proposed by incorporating with the wavelet function [13], which is a tool to model seismic signals by combining dilations and translations of a simple, oscillatory function (mother wavelet) of a finite duration. It is varied dynamically based on the properties of the wavelet function. A wavelet theory based mutation operation is proposed to embed on PSO for solving MC-TSCOPF problems. Thanks to the properties of wavelet, the better solution quality and stability with faster convergence speed can be produced by the proposed hybrid PSO for solving the MC-TSCOPF problem in the New England 39 bus system than the other existing PSO based methods [4,8-12].

2. MC-TSCOPF Problem Formulation

MC-TSCOPF [4] is mathematically defined as

\[
\begin{align*}
\text{(1)} & \quad \min \ f(x, y) \\
\text{(2)} & \quad \text{s.t. } \ g(x, y) = 0 \\
\text{(3)} & \quad H(x, y) \leq 0 \\
\text{(4)} & \quad U(x(t), y) \leq 0, t \in T
\end{align*}
\]

where \( x(t) \) is a dependent vector which includes active and reactive power of the swing bus, voltage angle and reactive power of generator buses, and voltage angle and magnitude of 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 \( t_e \). \( x \) represents the initial value of \( x(t) \) at \( t = 0 \). \( y \) is a control which includes vector active power and voltage magnitude of generator buses, voltage angle and magnitude of the swing bus, and tap position of transformers. \( f \) can be expressed as the total generation cost, total network loss, corridor transfer power, total cost of compensation, etc. \( g \) is the set of equality constraints which are usually the power flow constraints for a specified operating condition. \( H \) is inequality constraints for the steady-state security limits like bus voltage magnitude limits, generator power limits, thermal limits for transmission lines, etc. The dynamic security constraints set \( U \) is infinite in the functional space. Further details of the formulation of MC-TSCOPF are available in [4].

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

\[
\bar{F}(x) = \min \left\{ f(x, y) + \beta \max \left[ U(x(t), y) \right] \right\}
\]

Generally, transient stability constraints can be considered as hard constraints that should not be violated whilst the static constraints are soft in nature that slight violation could be tolerant. Compared with other constraint handling approaches, penalty function offers a simple and flexible strategy to effectively deal with mixed hard and soft constraints [14]. In addition, there is no need to have separate penalty factors for each type of constraints. In (5), any transient instability would introduce a huge angle deviation and thus produce a large violation and thus discrimination even though the
same penalty factor is used for all type of violations. Typically, $\beta = 1000$ works very well in most power systems [3,4].

3. Wavelet Mutation Based PSO

3.1. PSO

PSO is a powerful optimization method developed by Eberhart et al. [15-17]. It is modelled on processes of the sociological behaviour associated with bird flocking, and is one of the evolutionary computation techniques essentially. It uses a number of particles that constitute a swarm. Each particle traverses the search space looking for the global optimum.

In PSO, the velocity $v(t)$ (corresponding flight speed in a search space) and position $x(t)$ (a particle coordinates) of each particle at the $t$-th generation can be calculated using the following formulas [18]:

$$v(t) = k \cdot (w \cdot v(t-1) + \varphi_1 \cdot \text{rand}() \cdot (pbest - x(t-1)) + \varphi_2 \cdot \text{rand()} \cdot (gbest - x(t-1))$$

(6)

$$x(t) = x(t+1) + v(t)$$

(7)

where the best previous position of a particle is recorded and represented as $pbest$; the position of best particle among all the particles is represented as $gbest$; $w$ is an inertia weight factor; $\varphi_1$ and $\varphi_2$ are acceleration constants; $\text{rand}()$ returns a uniform random number in the range of $[0,1]$; $k$ is a constriction factor derived from the stability analysis of equation (7) to ensure the system to be converged but not prematurely [17]. Mathematically, $k$ is a function of $\varphi_1$ and $\varphi_2$ as reflected in the following equation:

$$k = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}$$

(8)

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

PSO utilizes $pbest$ and $gbest$ to modify the current search point to avoid the particles moving in the same direction, but to converge gradually toward $pbest$ and $gbest$. Suitable selection of inertia weight $w$ provides a balance between global and local explorations. Generally, $w$ can be dynamically set with the following equation [19]:

$$w = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{T} \times t$$

(9)

where $t$ is the current number of iterations, $T$ is the total number of iteration, $w_{\text{max}}$ and $w_{\text{min}}$ are the upper and lower limits of the inertia weight.

In (6), the particle velocity is limited by a maximum value $v_{\text{max}}$. The parameter $w_{\text{min}}$ determines the resolution, or fitness, 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 $v_{\text{max}}$ is too high, particles might fly past good solutions. If $v_{\text{max}}$ is too small, particles may not explore sufficiently beyond local solutions. In many experiences with PSO, $v_{\text{max}}$ was often set at 10%-20% of the dynamic range of the variable on each dimension. Fig. 1a lists the pseudocode for the basic PSO.
begin
Create a set of particles.
while (not termination condition) do
    begin
    Evaluation the fitness of each particle.
    Update velocity and position of each particle based on (6) and (7) respectively.
    end
end

Fig. 1 (a) Pseudocode for PSO, (b) Pseudocode for hybrid PSO with GA’s mutation

In [4], the PSO have been developed for solving MS-TSCOPF problems with reasonable results in solution quality found and were shown to be better the genetic algorithm. This implementation is referred as the Standard PSO (SPSO) in this paper. The detailed procedures of SPSO are given in the Appendix.

However, based on our observation that SPSO works well in the early iterations, but it usually presents problems reaching a near-optimal solution. The behaviour of the SPSO in the model presents some important aspects related with the velocity update. If a particle’s current position coincides with the global best position, the particle will only move away from this point if its inertia weight and velocity are different from zero. If their velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to a premature convergence to the SPSO and no further improvement can be obtained. This phenomenon is known as stagnation by [17].

Ahmed et al. [8] proposed to integrate GAs’ mutation into PSO, which aids to break through stagnation. It starts with the random choice of a particle in the swarm and moves to different positions inside the search area by using the mutation. The mutation operation illustrated in (10) is used in Ahmed et al.’s hybrid PSO:

\[
\text{mut}(x_i(t)) = x_i(t) - \omega
\]

where \( x_i(t) \) is the coordinate of an randomly choice element from the particle \( x(t) = \{x_1(t), x_2(t), ..., x_n(t)\} \) with \( n \) elements at \( t \)-th iteration, and \( \omega \) is randomly generated within the range \([0, 0.1 \times (x_{\text{max}}^i - x_{\text{min}}^i)]\), representing 0.1 times the length of the search space of \( x_i(t) \). The pseudocode of hybrid PSO with GAs’ mutation is shown in Figure 1b, in which after updating velocities and positions of particles, mutation on particles is performed. It can also be found from Figure 1 that the pseudocodes of both PSO methods are identical except the mutation operation is integrated in the hybrid PSO with GAs’ mutation.

For solving MC-TSCOPF problems, a hybrid PSO integrated with mutation operation (10) named as APOS is shown in the Appendix. It can be noticed from (10), however, that the mutating space in APOS is limited by \( \omega \) in which 10% of the range of the \( i \)-th element of the particle (i.e. \([0, 0.1 \times (x_{\text{max}}^i - x_{\text{min}}^i)]\)) is mutated. It may not be the best approach in fixing the mutating space all the time along the search. Therefore the
approach can be further improved by modifying with a dynamic mutation operation in which the mutating space varies dynamically along the search.

Here a wavelet mutation, which will be discussed in Section 3.2, is proposed by varying the mutating space dynamically based on the wavelet theory [13]. A wavelet mutation based hybrid PSO named WPSO is proposed to solve the MC-TSCOPF problems. Its procedures are identical to APSO except the mutation operation used. In APSO, the mutation operation (10) is embedded, and in WPSO, the wavelet mutation discussed in Section 3.2 is embedded. The detailed procedures of WPSO are shown in the Appendix.

3.2. Wavelet Mutation

Before presenting the wavelet mutation operation, the basic wavelet theory is briefly reviewed.

A. Wavelet theory

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

\[ \int_{-\infty}^{\infty} \psi(x) dx = 0 \]

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

\[ \int_{-\infty}^{\infty} |\psi(x)|^2 dx < \infty \]

where most of the energy in \( \psi(x) \) is confined to a finite duration and bounded. The Morlet wavelet (as shown in Figure 2) [13] is an example mother wavelet:

\[ \psi(x) = e^{-x^2/2} \cos(5x) \]

The Morlet wavelet integrates to zero (Property 1). Over 99% of the total energy of the function is contained in the interval of \(-2.5 \leq x \leq 2.5\) (Property 2). In order to control the magnitude and the position of \( \psi(x) \), a function \( \psi_{a,b}(x) \) is defined as follows.
Fig. 2 Morlet wavelet.

\[
\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right)
\]

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

\[
\psi_{1,0}(x) = \psi(x)
\]

As

\[
\psi_{a,0}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x}{a}\right)
\]

it follows that \(\psi_{a,0}(x)\) is an amplitude-scaled version of \(\psi(x)\). Figure 3 shows different dilations of the Morlet wavelet. The amplitude of \(\psi_{a,0}(x)\) will be scaled down as the dilation parameter \(a\) increases. This property is used to do the mutation operation in order to enhance the searching performance.

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

**B. Operation of wavelet mutation**

The mutation operation is used to mutate the particles of the swarm. In general, various methods like uniform mutation or non-uniform mutation [20,21] can be employed to realize the mutation operation. The proposed Wavelet Mutation (WM) operation exhibits a fine-tuning ability. The details of the operation are as follows. Every particles of the swarms will have a chance to mutate governed by a probability of mutation, \(\mu_m \in [0, 1]\), which is defined by the user. For each particle, a random number between 0 and 1 will be generated such that if it is less than or equal to \(\mu_m\), the mutation will take place on that particle. For instance, if \(\mathbf{x}(t) = [x^1(t), x^2(t), \ldots, x^n(t)]\) is
the selected particle and the $j$-th element $x_j(t)$ is randomly selected for mutation (the value of $x_j(t)$ is within its boundary $[\text{para}_{\text{min}}^j, \text{para}_{\text{max}}^j]$), the resulting particle is given by

$$x_j(t+1) = \begin{cases} x_j(t) + \sigma \times (\text{para}_{\text{max}}^j - x_j(t)) & \text{if } \sigma > 0 \\ x_j(t) + \sigma \times (x_j(t) - \text{para}_{\text{min}}^j) & \text{if } \sigma \leq 0 \end{cases},$$

(17)

$$\sigma = \psi_{a,0}(\phi),$$

(18)

$$\sigma = \frac{1}{\sqrt{a}} \psi \left( \frac{\phi}{a} \right),$$

(19)

By using the Morlet wavelet shown in (12) as the mother wavelet,

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

(20)

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

$$\frac{1}{N} \sum_{N} \sigma = 0 \quad \text{for} \quad N \rightarrow \infty,$$

(21)

where $N$ is the number of samples.

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

$$a = e^{-\ln(g) \times \left(1 - \frac{t}{T}\right)^{\tau_{\text{sum}}} + \ln(g)}$$

(22)
where $\zeta_{wm}$ is the shape parameter of the monotonic increasing function, $g$ is the upper limit of the parameter $a$. The effects of the various values of the shape parameter $\zeta_{wm}$ to $a$ with respect to $\tau/T$ are shown in Figure 4. In this figure, $g$ is set as 10000. Thus, the value of $a$ is between 1 and 10000. Referring to (20), the maximum value of $\sigma$ is 1 when the random number of $\varphi=0$ and $a=1$ ($\tau/T=0$). Then referring to (17), the element $x'(t+1)=x'(t)+1\times(\text{para}_{\text{max}}'-x'(t))=\text{para}_{\text{max}}'$. It ensures that a large mutating space for the mutated element inside the particle is given. When the value $t/T$ is near to 1, the value of $a$ is so large that the maximum value of $\sigma$ 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 $\varphi$ is zero, the value of $\sigma$ will be equal to 0.0158. With $x'(t+1)=x'/t)+0.0158\times(\text{para}_{\text{max}}'-x'(t))$, a smaller mutating space for the mutated element inside the particle can be used for fine-tuning.

![Figure 4](image_url)  

**Fig. 4** Effect of the shape parameter $\zeta_{wm}$ to $a$ with respect to $\tau/T$.

4. **Result and Analysis**

A case study of solving the MC-TSCOPF problem for the New England 39-bus system is used to demonstrate the effectiveness and robustness of the proposed WPSO. Comparison is given with SPSO [4] and APSO [8].

All the PSO approaches were coded in Matlab and were run on a Pentium D 3.4GHz personal computer with 1GB RAM. The system data of the power system was collected in [22,23]. The New England 39-bus test system comprises 10-generator, 39-bus, and 46-line. Power System Toolbox [22] was employed to perform time-domain transient stability simulations for determining generator rotor trajectories. The time step adopted is 0.01s and the integration time interval is fixed to 1.5s. The total load for the operating condition considered is 6,098MW and 1,409MVAr. There are three on-load tap changers connected buses 11-12, 12-13 and 19-20.

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

Contingency 1: A three phase fault occurred at the end of line 26-27 near bus 26. The fault was cleared by tripping the line at bus 26 after 110 ms and at bus 27 after 120 ms.
Contingency 2: A three phase fault occurred at the end of line 16-17 near bus 16. The fault was cleared by tripping the line at bus 16 after 80 ms and at bus 17 after 100 ms.

With the above two contingencies, the following 4 cases were built.
Case 0: conventional OPF without any transient stability constraints
Case 1: transient stability constrained OPF with contingency 1 considered only
Case 2: transient stability constrained OPF with contingency 2 considered only
Case 3: transient stability constrained OPF with contingency 1 and 2 considered

The parameters used in the three PSO approaches (SPSO, APSO and WPSO), which are same as the parameters used in our previous study [4], are followings: swarm size = 30, acceleration constants $\phi_1 = \phi_2 = 2.05$, penalty factor $\beta = 1000$, pre-defined number of iterations = 50, maximum velocity $v_{max} = 0.2$, the upper and lower limits of the inertia weights $w_{min} = 0.1$ and $w_{max} = 1.2$. In both APSO and WPSO, the mutation probability is set at $\mu_m = 0.2$. In WPSO, its parameters is set at $g = 10000$ and $\zeta_{nm} = 5$.

<table>
<thead>
<tr>
<th>Case</th>
<th>SPSO</th>
<th>APSO</th>
<th>WPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>36279 (3)</td>
<td>36240 (2)</td>
<td>36220 (1)</td>
</tr>
<tr>
<td></td>
<td>36557 (3)</td>
<td>36405 (2)</td>
<td>36378 (1)</td>
</tr>
<tr>
<td></td>
<td>36395 (3)</td>
<td>36306 (2)</td>
<td>36256 (1)</td>
</tr>
<tr>
<td></td>
<td>36754 (3)</td>
<td>36629 (2)</td>
<td>36557 (1)</td>
</tr>
<tr>
<td>Variance</td>
<td>36279.9 (3)</td>
<td>20027 (3)</td>
<td>491.38 (1)</td>
</tr>
<tr>
<td></td>
<td>36557.7 (3)</td>
<td>24449 (3)</td>
<td>1214.4 (1)</td>
</tr>
<tr>
<td></td>
<td>36395.7 (3)</td>
<td>8939.1 (2)</td>
<td>3733.5 (1)</td>
</tr>
<tr>
<td></td>
<td>36754.7 (3)</td>
<td>20525 (2)</td>
<td>2203.8 (1)</td>
</tr>
</tbody>
</table>

Table 1 Means and variances of the best values in the 50 runs found by the methods

Since all PSO methods are stochastic algorithms, different solutions are obtained with runs. The better the PSO method is, the smaller mean and variance of solutions in runs can be obtained. Therefore 50 test runs were performed to collect the two statistics for the means and variances, which are detailed on Table 1 for the above four cases using the three PSO approaches (SPSO, APSO and WPSO). The ranks of the means and variances for the four cases using the three PSO approaches are shown in the blankets. It can be found from Table 1 that WPSO achieves the best mean cost among the three PSO algorithms. In fact, WPSO obtains the lowest cost values in all cases. Also the variances of WPSO are the smallest comparing with the other two PSO methods in all cases. The smaller the variance means the closer the values cluster around the mean. Since all the variances of WPSO are the smallest, it demonstrates that the WPSO is capable to approach and keep searching around the optimal mean closer. Therefore WPSO can produce better and more stable solution quality than the other two PSO methods in all four cases.
Table 2 \(t\)-values between WPSO to the other PSO methods

<table>
<thead>
<tr>
<th></th>
<th>Case 0</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-values between WPSO to SPSO</td>
<td>4.6454</td>
<td>8.6845</td>
<td>5.8548</td>
<td>8.2789</td>
</tr>
<tr>
<td>T-values between WPSO to APSO</td>
<td>2.9996</td>
<td>2.5639</td>
<td>3.1326</td>
<td>4.1351</td>
</tr>
</tbody>
</table>

The \(t\)-test is then used to evaluate how significance the WPSO better than the other PSO methods is, and the \(t\)-values are shown in Table 2. It shows that all \(t\)-values in Table 2 are higher than 2.15. Based on the normal distribution table, if the \(t\)-value is higher than 2.15, the significance is with 98% confident level. Therefore the performance of WPSO is significantly better than the other two PSO methods with 98% confident in New England 39-bus system for all cases.

For the New England 39-bus system, those convergence plots of APSO, SPSO and WPSO methods for case 0, 1, 2 and 4 are shown in Figs 5-8 respectively. 5\(^{th}\) to 50\(^{th}\) iterations among the 50 iterations are shown on the plots to illustration the different of the progresses of the PSO methods. It can be observed clearly from the figures that in general the convergence speed of WPSO with wavelet mutation operation are faster than the other two APSO and SPSO. Also WPSO can reach better solutions than the other PSO methods. Therefore WPSO is more likely to reach better solutions and pre-mature convergence is more unlikely to be happened in WPSO than the other PSO methods.

![Fig. 5 Convergence curves of the PSO methods for Case 0](image-url)
Fig. 6 Convergence curves of the PSO methods for Case 1

Fig. 7 Convergence curves of the PSO methods for Case 2

Fig. 8 Convergence curves of the PSO methods for Case 3
However, it is hard to count computational efforts used on the PSO methods to reach the reasonable solutions from the convergence curves. In [4], it has already been demonstrated that SPSO can reach the acceptable solutions for the New England 39-bus system. Its solutions found are supposed to be the acceptable solutions of the MC-TSCOPF problems. Table 3 shows the number of computational iterations used on the three PSO methods (i.e. SPSO, APSO and WPSO) to reach the solutions found by SPSO. In Table 3, computational iterations used on SPSO are all 50, since the pre-defined numbers of computational iterations 50 are all used in all cases. ‘Nil’ from the table means that the method cannot reach the acceptable solution found by SPSO. It can be found from Table 3 that WPSO can reach the acceptable solutions with smallest numbers of computational iterations than the other PSO methods.

<table>
<thead>
<tr>
<th></th>
<th>Case 0</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>APSO</td>
<td>30</td>
<td>21</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td>WPSO</td>
<td>21</td>
<td>16</td>
<td>18</td>
<td>22</td>
</tr>
</tbody>
</table>

**Table 3** Number of iterations performed in the PSO methods (i.e. SPSO, APSO and WPSO) until the acceptable solution reached

Furthermore the computational times used (in seconds) on all PSO methods, that can reach the acceptable solutions found by SPSO, were also recorded in Table 4. It can also be found from Table 4 that WPSO can reach the acceptable solutions with shortest computational times than the other PSO methods. Both Table 3 and 4 show that WPSO used less than half computational effort to reach the acceptable solutions than the SPSO used in all cases.

<table>
<thead>
<tr>
<th></th>
<th>Case 0</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
<td>13.74</td>
<td>360.54</td>
<td>336.32</td>
<td>680.18</td>
</tr>
<tr>
<td>APSO</td>
<td>8.04</td>
<td>137.82</td>
<td>169.44</td>
<td>371.14</td>
</tr>
<tr>
<td>WPSO</td>
<td>5.74</td>
<td>106.94</td>
<td>121.80</td>
<td>294.54</td>
</tr>
</tbody>
</table>

**Table 4** Computational time (in seconds) performed in the PSO methods (i.e. SPSO, APSO and WPSO) until the acceptable solution reached

The best solutions found by WPSO for New England-39 bus system are shown in the last columns of Table A1 in the appendix. In addition, the rotor angle curves for both contingency 1 and contingency 2 are shown in Fig. 9 and 10, respectively. They indicate that the rotor angles of all generators for the two contingencies are within the boundary. Therefore transient stability can be maintained after the faults.

In the tested hybrid PSO methods (APSO and WPSO), their steps are very similar except that different GAs’ evolutionary operations used. In WPSO, the proposed wavelet mutation operation is integrated to the PSO method. These results indicate that the proposed wavelet mutation operation aides to find the acceptable solutions with better solution quality and shorter computational time in the MC-TSCOPF problems than APSO, which is better than the other existing hybrid PSO methods [12].
5. Conclusion

In this paper, we proposed to incorporate PSO with the wavelet mutation operation to solve the challenging MS-TSCOPF problems. In the proposed wavelet mutation operation, the wavelet theory was applied. Our objective is to apply the properties of the wavelet theory to enhance PSO, so that it could explore solution spaces more effectively and robustly in reaching better solutions. The case study showed that the proposed wavelet mutation based hybrid PSO is useful as an optimization technique to solve the challenging MC-TSCOPF problem. Thanks to the properties of the wavelet, the feasibility and robustness of the proposed wavelet mutation based hybrid PSO for solving the MC-TSCOPF problem were demonstrated on the New England 39-bus system with better results in solution quality and stability than the other tested PSO.
methods. By observing the general statistical analyses in the tested PSO methods with the $t$-test, it confirmed that proposed wavelet mutation based hybrid PSO outperformed the other methods significantly. Also the proposed method can reach reasonable solutions with faster convergence speeds and smaller computational efforts than the other methods.

6. Acknowledgement

The authors gratefully acknowledge the support by The University of Western Australia, Australia, and The Hong Kong Polytechnic University under Project A-PA4Y and 4-ZV28.

REFERENCES


APPENDIX

<table>
<thead>
<tr>
<th>Generators</th>
<th>a</th>
<th>b</th>
<th>C</th>
<th>(P_{\text{min}}) (MW)</th>
<th>(P_{\text{max}}) (MW)</th>
<th>Optimal load found by WPSO (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>100</td>
<td>350</td>
<td>350.00</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>600</td>
<td>1145.5</td>
<td>659.36</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>750</td>
<td>601.30</td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>732</td>
<td>568.35</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>608</td>
<td>548.37</td>
</tr>
<tr>
<td>6</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>750</td>
<td>576.58</td>
</tr>
<tr>
<td>7</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>660</td>
<td>569.89</td>
</tr>
<tr>
<td>8</td>
<td>0.01</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>640</td>
<td>571.02</td>
</tr>
<tr>
<td>9</td>
<td>0.006</td>
<td>0.3</td>
<td>0.2</td>
<td>250</td>
<td>930</td>
<td>751.94</td>
</tr>
<tr>
<td>10</td>
<td>0.006</td>
<td>0.3</td>
<td>0.2</td>
<td>600</td>
<td>1100</td>
<td>10483</td>
</tr>
</tbody>
</table>

Table AI. Parameters of New England 39-bus system

Algorithm: SPSO

Step 1: Input system data, contingency set, SPSO parameters and specify the lower and upper boundaries of each variable. Control variables include the active power and terminal voltage of each generator, voltage angle and magnitude of the swing bus, and tap position of each transformer.

Step 2: Each particle in the swarm represents a feasible candidate solution to the optimization problem and is initialized randomly with all control variables satisfied their practical operation constraints.

Step 3: For each particle, an unconstrained Newton-Raphson power flow calculation
is used to determine the power flow solution, which includes all the dependent variables, for a given set of control variables.

Step 4: Evaluate the fitness of each particle using the evaluation function described in (5). Power flow solution obtained in Step 3 is used to evaluate the objective function (1) and the static violations (2)-(3). For transient stability violation evaluation (4), transient stability simulation is used to produce the generator rotor responses. The maximum rotor angle deviation from the COI, among all generators and contingencies, is then used to compute a transient stability penalty using (5).

Step 5: Find the best position of the swarm $g_{best}$ and the best position of each particle $p_{best}$ by comparing the evaluation value $F(x)$ of each particle with the one in $p_{best}$. If $F$ is better, then set $p_{best}$ to the corresponding $x$. The best among $p_{best}$ is denoted as $g_{best}$.

Step 6: If there is any stopping criteria being satisfied, go to Step 11; otherwise, increment the iteration number $i$.

Step 7: Update the inertia weight $w$ according to equation (9).

Step 8: Update the velocity $v$ of each particle according to equation (6).
   
   If $v > v_{max}$, $v = v_{max}$.
   
   If $v < -v_{max}$, $v = -v_{max}$.

Step 9: Update the position of each particle by (7). If a particle violates its position limits (i.e. limits of the control variables) in any dimension, set its position at the proper limit.

Step 10: Return to Step 4 to repeat the evaluation process with updated position, until the termination condition is reached.

Step 11: The particle that generates the latest $g_{best}$ is the optimal value.

Remarks:

Explicit handling of any static violations in the dependent variables or any transient stability violations is not required since their effects have been incorporated in the penalty function already. In other words, those violations would be dealt with implicitly through the optimization process by homing to the fittest solution which has little or no violations.

Algorithm: APSO

The procedures of APSO [1] are shown:

Step 1: Step 1 to Step 9 of SPSO

Step 2: Perform the mutation operation based on (10) discussed in Section 3.1

Step 3: Step 10 to Step 11 of SPSO

Algorithm: WPSO

The procedures of WPSO are shown:

Step 1: Step 1 to Step 9 of SPSO

Step 2: Perform the wavelet mutation operation based on (17) discussed in Section 3.2

Step 3: Step 10 to Step 11 of SPSO
BIOGRAPHIES

K.Y. Chan is currently a Postdoctoral Research Fellow in the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University, Hong Kong. Dr. Chan received his MPhil degree in Electronic Engineering from City University of Hong Kong, Hong Kong and his PhD degree in Computing from London South Bank University, United Kingdom. His research interests include computational intelligence and its applications in product design, signal processing, power systems and operation researches.

S.H. Ling was born in Hong Kong. He received the BEng degree from the Department of Electrical Engineering, MPhil and PhD degrees from the Department of Electronic and Information Engineering in the Hong Kong Polytechnic University in 1999, 2002 and 2006 respectively. He is currently a Postdoctoral Research Associate in the School of Electrical, Electronic and Computer Engineering in the University of Western Australia, Australia. He has published over 45 research papers on computational intelligence and its industrial applications. His current research interests include evolution computation, fuzzy logic, neural network, power system, recognition and biomedical engineering.

K.W. Chan received his BSc (Hons) and PhD Degrees in Electronic and Electrical Engineering from the University of Bath (UK). He currently is an Assistant Professor in the Department of Electrical Engineering of the Hong Kong Polytechnic University, Hong Kong SAR. His main interest is power system stability analysis and control, power system simulation and security assessment.

G.T.Y. Pong received his BSc from the University of Wisconsin, MSc from the University of Missouri, U.S.A. and PhD from the Loughborough University, U.K. He is a Senior Lecturer in the Department of Applied Mathematics of the Hong Kong Polytechnic University, Hong Kong SAR. His main research interests are computational mathematics and optimization.

H.H.C. Iu received the B.Eng. (Hons) degree in electrical and electronic engineering from the University of Hong Kong, Hong Kong, in 1997. He received the PhD degree from the Hong Kong Polytechnic University, Hong Kong, in 2000. He is currently a Senior Lecturer in the School of Electrical, Electronic and Computer Engineering at the University of Western Australia, Australia. His research interests include nonlinear circuits and systems, power electronics and TCP network dynamics. He has published over 60 papers in these areas. He currently serves as a Guest Editor for the Australian Journal of Electrical and Electronics Engineering and Circuits, Systems and Signal Processing.

1Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR
2School of Electrical, Electronic and Computer Engineering, The University of Western Australia, Australia
3Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong SAR
4Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR