An Investigation into Gaussian Mutation for Particle Swarm Optimization on Economic Dispatch Problems

Pichet Sriyanyong

1. Introduction

Particle swarm optimization (PSO) seems to be one of the most promising stochastic population-based algorithms for a wide range of engineering problems. The attractive characteristics of PSO include: ease of implementation, fast convergence compared with the traditional evolutionary computation techniques, and stable convergence characteristic. However, PSO tends to suffer from premature convergence for strongly multi-modal optimization problems. Aiming at overcoming this limitation, this scholarly article investigates a method of incorporating a Gaussian mutation (GM) operator into the PSO algorithm to enhance the global search capability, especially in Economic Dispatch Problem (EDP).

Generally, EDP is a significant optimization problem in power system operation where it is used to schedule the optimum scheduling of generation at a particular time that minimizes the total production cost while satisfying an equality constraint and inequality constraints, i.e. power balance constraint and operating limits [1], [2]. In recent years, there are many evolutionary computation techniques have been developed and proposed so as to solve a wide range of power system problems including EDP, for example Simulated Annealing (SA) [3], Genetic Algorithm (GA) [4], Evolutionary Programming (EP) [5], Tabu Search (TS) [6], and Particle Swarm Optimization (PSO) [7], [8] etc.

Until now, substantial efforts related to developing of the PSO performance in various areas of the problems have been carried out. There are several pieces of research that investigate and report on some testing on the optimum values, which the best solution is obtained for the particular problem and applied method. That can be classified into two main problem categories: (1) Mathematical problems [9]-[12], and (2) Real-world problems [7], [13]-[16]. However, only a few papers have focused on the diversity investigation where almost of them preferred to examine the mathematical cases. So, this study aims to focus directly on the searching ability analysis of the PSO algorithms that investigate the effect of diversity variation on EDP.

This scholarly article is organized as follows: Section 2 and 3 presents a brief introduction to PSO algorithm as well as Gaussian mutation. Section 4 illustrates the detail of Economic Dispatch Problem (EDP), whilst the Diversity Factor is introduced.

---

1 Lecturer, Department of Teacher Training in Electrical Engineering, Faculty of Technical Education, King Mongkut’s University of Technology North Bangkok. Tel. 0-2913-2500 Ext. 3343 E-mail: psyy@kmutnb.ac.th

Received February 27, 2009; Accepted April 8, 2009
in the section 5. Then, section 6 shows the development of the PSO-GM. Section 7 gives the discussion about diversity investigation. Finally, conclusion, discussion and suggestion are given in the last section.

2. Overview of Particle Swarm Optimization

Modern heuristic optimization algorithms are considered as promising for nonlinear optimization problems [17], [18] as they do not require that the objective function has to be differentiable and continuous. Particle swarm optimization (PSO) is one of the modern heuristic optimization techniques where it was originally introduced Kennedy and Eberhart in 1995 [19]. Similar to other evolutionary computation techniques, PSO employs the principle of a random initialized population and the concept of evaluation and modification of a population to find the optimal solution. Mathematically, the fundamental model of PSO can be expressed by the following [9]:

Let a swarm have \( n \) particles in a \( d \)-dimensional search space. At the \( t \)th iteration, \( x_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{id}^t) \) expresses the position of the \( i \)th particle and \( pbest_i^t = (pbest_{i1}^t, pbest_{i2}^t, \ldots, pbest_{id}^t) \) shows the best previous position of the \( i \)th particle. In addition, the best position among all the particles is represented by \( gbest_i = (gbest_{11}, gbest_{12}, \ldots, gbest_{id}) \). The velocity of the \( i \)th particle can be represented by \( v_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{id}^t) \). Each of the population, called a particle or agent, can be updated or changed to the new position according to the current velocity, the difference between the current position and the best value itself \((pbest)\) and its group \((gbest)\) [20]. There are a number of algorithms [21] used to update the velocity of the \( i \)th particle, and they are discussed below.

2.1 Original PSO Algorithm (OPSO)

In this case, the modified velocity can be calculated from:

\[
v_i^{t+1} = v_i^t + c_1 \times rand_i \times (pbest_{id} - x_i^t) + c_2 \times rand_2 \times (gbest_d - x_i^t)
\]

where the values of both \( c_1 \) and \( c_2 \) are set to a value of 2, while both \( rand_1 \) and \( rand_2 \) are random numbers between 0 and 1.

2.2 Basic PSO Algorithm (BPSO)

As the OPSO does not adapt the velocity step size, it may lead to a poor searching ability. Consequently, BPSO utilizes an inertia weight \((w)\) in order to balance the global and the local searches. The updated velocity in the BPSO is calculated by:

\[
v_i^{t+1} = w_i \cdot v_i^t + c_1 \times rand_i \times (pbest_{id} - x_i^t) + c_2 \times rand_2 \times (gbest_d - x_i^t)
\]

where \( w \) is 0.9 at the first iteration and linearly decreases to 0.4 at the final iteration [9].

2.3 Constriction Factor PSO Algorithm (CPSO)

CPSO has been proposed by Clerc [22]-[24] so as to ensure convergence of the PSO algorithm. The updated velocity in the CPSO can be expressed by:

\[
v_i^{t+1} = k \cdot [v_i^t + c_1 \times rand_i \times (pbest_{id} - x_i^t) + c_2 \times rand_2 \times (gbest_d - x_i^t)]
\]

\[
k = \frac{2}{\sqrt{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}}, \quad \varphi = c_1 + c_2, \quad \varphi > 4
\]

where \( \varphi \) is generally set to 4.1, both \( c_1 \) and \( c_2 \) are set to 2.05 and \( k \) is 0.729 as presented in [24].
2.4 Original PSO Including Inertia Weight and Constriction Factor (CBPSO)

In this algorithm, both the inertia weight and constriction factor are incorporated into the original PSO, as presented in [25]-[28]. The modified velocity of each particle can be calculated as follows:

\[ v_{id}^{t+1} = k \times [w \cdot v_{id}^t + c_1 \times rand_1 \times (p_{best_id} - x_{id}^t) + c_2 \times rand_2 \times (g_{best_d} - x_{id}^t)] \]  

where: 
- \( v_{id}^t \): velocity of particle \( i \) at iteration \( t \) in \( d \)-dimensional space; \( V_{d,min} \leq v_{id}^t \leq V_{d,max} \); 
- \( i = 1, 2, ..., n \), \( d = 1, 2, ..., m \), 
- \( x_{id}^t \): current position of particle \( i \) at iteration \( t \), 
- \( w \): inertia weight factor, 
- \( t \): number of iterations, 
- \( n \): number of particles in a group, 
- \( m \): number of members in a particle, 
- \( k \): constriction factor, 
- \( c_1, c_2 \): acceleration constant, 
- \( rand_1, rand_2 \): uniformly distributed random number between 0 and 1.

Subsequently, the modified position of each particle can be calculated as shown in the following equations:

\[ x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \]  

where: 
- \( P_m \): mutation probability.

3. Overview of Gaussian Mutation

Normally, Gaussian mutation (GM) is applied to Genetic algorithm (GA) so as to enhance the GA searching ability. Similarly, this scholarly article aims at coping with the loss of diversity in global search by incorporating Gaussian mutation into the traditional PSO as presented in [26], [29], [30]. Applying Gaussian mutation improves the PSO searching ability by mutating some selected particles. The procedures of the implementation can therefore be expressed in details as follows:

**Step 1:** Determine the mutation probability \( P_m \) by:

\[ P_m = \frac{R_m}{m} \]  

where \( R_m \) and \( m \) are mutation rate and the number of particles, respectively. As reported in [26], \( R_m \) is set to 1 at the first iteration and linearly decreases to 0 at the final iteration.

**Step 2:** Generate a uniformly distributed random number \((\text{rand}_i)\) between 0 and 1 for each particle.

**Step 3:** Compare each generated random number \((\text{rand}_i)\) with \( P_m \). If \( P_m > \text{rand}_i \), then mutate the particle by following equation [26].

\[ x_{i,\text{mutate}} = x_i \times (1 + \text{gaussian}(\sigma)) \]  

where \( x_i \) and \( x_{i,\text{mutate}} \) denote the current and mutated position of particle \( i \) at iteration \( t \), whilst \( \text{gaussian}(\sigma) \) is a random number drawn from a Gaussian distribution. It can be calculated from \( \sigma = 0.1 \times \text{The length of search space} \).

4. Economic Dispatch Problem (EDP)

The objective of the EDP is to minimize the total fuel cost subjected to various constraints [1]. In general, the mathematical model of the EDP is as follows [2]:

\[ \text{Minimize} : TC = \sum_{i=1}^{n} F_i(P_i) \]
where $TC$ is total production cost, $F_i(P_i)$ is fuel cost of $i^{th}$ generator, and $N$ is the number of generators.

Subject to the equality and inequality constraints as follows:

a) Power balance constraint

$$\sum_{i=1}^{N} P_i = P_D \quad (10)$$

b) Operating limit constraints

$$P_{i,\text{min}} \leq P_i \leq P_{i,\text{max}} \quad (11)$$

where $P_i$ is power output of $i^{th}$ generator, $P_D$ is power demand, and $P_{i,\text{min}}$, $P_{i,\text{max}}$ is minimum / maximum power output of $i^{th}$ generator.

In practical EDP, valve-point loadings have been taken into account, so it will increase multiple local minimum points in the cost function and make the problem more difficult [31]. For the standard ED problem, a single quadratic function is commonly considered. The generator’s fuel cost can be calculated from $F_i(P_i) = a_i P_i^2 + b_i P_i + c_i$ for the smooth cost functions and $F_i(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i| \times \sin(f_i \times P_{i,\text{min}} - P_i)$ for the non-smooth cost functions [4] where $a_i$, $b_i$ and $c_i$ are coefficients of the fuel cost function, $e_i$ and $f_i$ are coefficients from the valve-point loading. Figure 1 illustrates an example of smooth cost and non-smooth cost functions with valve-point loading.

### 5. Diversity Factor

According to verifying the searching capability of the hybrid PSO-GM method, the Diversity Factor ($d$) will be adopted for analysis. As reported in [32], has been recommended in preliminary investigation. Thus, the Diversity Factor ($d$) can be calculated by the following equation [32]:

$$\text{Diversity}(d) = \frac{1}{|S| \cdot |L|} \sum_{i=1}^{N} \sum_{j=1}^{N} (P_{ij} - \bar{P}_j)^2, \quad (12)$$

where $|S|$ is the swarm size; $|L|$ is the length of the longest diagonal in the search space; $N$ is the dimensionality of the problem denote; $P_{ij}$ is the value of $i^{th}$ particle at $j^{th}$ dimensional space; and $\bar{P}_j$ is the value of the average point $\bar{P}$ at $j^{th}$ dimensional space.

### 6. Development of the Hybrid PSO-GM Algorithm for EDP

The fundamental concept of the PSO-GM in this section are the same as reported in [8], [33], [34], where the Gaussian mutation (GM) is integrated into the PSO algorithm to increase the searching diversity when applying to the EDP. That will lead to an enhancement of the global searching ability. The procedures of the PSO-GM method are demonstrated in Figure 2.

### 7. Investigation of Using Gaussian Mutation for PSO

In this section, to study the effect of enhancing the diversity of the traditional PSO algorithms, an example of 3-unit system with non-smooth cost functions has been
examined by testing on the three traditional PSO algorithms (i.e. BPSO, CPSO, and CBPSO) as well as the hybrid PSO-GM technique, respectively. In this example case, the power demand is assumed to be 850 MW. From the previous research, the global best known solution is $8234.07$ as reported in [6]. The data of the system are adopted from [4]. All investigations are carried out using Matlab platform and executed on a personal computer. The values of the simulation parameters for four PSO algorithms are shown in Table 1.

Figure 3 shows the example of convergence characteristics of a PSO algorithm with randomness of the initial conditions as presented in [8]. It reveals from the figure that the results are possible to reach the best global solution, while each particle is randomly generated at the initial state. Conversely, the results may become trapped in a local optimum as well.

Concerning the investigations, they are divided into two groups according to the different aspects of the results. The first group focuses on the example of experimental results and the convergence to the optimum solution, while the second group considers the diversity characteristic amongst the four PSO algorithms.

### 7.1 Investigation into Convergence Characteristic

The example of experimental results are recorded and compared with the result obtained from some selected methods in order to demonstrate their effectiveness as shown in Table 2.
It is observed that the four PSO algorithms can reach the global best solution ($8234.07) as other methods. Moreover, it can be seen that the hybrid PSO-GM algorithm outperforms the three traditional PSO algorithms and the other methods in terms of the mean cost. Therefore, it clearly confirms that the hybrid PSO-GM show a great potential in the practical application as EDP.

Table 2 Comparison of the example of experimental results obtained by various methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean cost ($)</th>
<th>Min. cost ($)</th>
<th>Max. cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA [4]</td>
<td>-</td>
<td>8237.60</td>
<td>-</td>
</tr>
<tr>
<td>IEP [5]</td>
<td>-</td>
<td>8234.09</td>
<td>-</td>
</tr>
<tr>
<td>CEP [31]</td>
<td>8235.97</td>
<td>8234.07</td>
<td>8241.83</td>
</tr>
<tr>
<td>BPSO</td>
<td>8237.5290</td>
<td>8234.0717</td>
<td>8241.5875</td>
</tr>
<tr>
<td>CPSO</td>
<td>8238.2054</td>
<td>8234.0717</td>
<td>8241.5875</td>
</tr>
<tr>
<td>CBPSO</td>
<td>8240.4345</td>
<td>8234.0717</td>
<td>8382.7283</td>
</tr>
<tr>
<td>PSO-GM</td>
<td>8235.8008</td>
<td>8234.0717</td>
<td>8241.5875</td>
</tr>
</tbody>
</table>

In addition, the plots of average convergence curves are also plotted in Figure 4. Due to the randomness of the simulation results, each point on the following graphs is obtained from an average of the results over 100 different runs.

Considering the degree of approaching to the optimal solution in Figure 4, the PSO-GM comes first, the BPSO second, the CPSO third, and the CBPSO last. In terms of convergence rates, the BPSO and the CPSO perform well in very early stage of the search processes, but performance deteriorates quickly around 50 iterations, while the PSO-GM has a lower convergence rate at the beginning, but the convergence rate increases moderately up to 220 iterations and decrease quickly afterwards.

7.2 Investigation into Diversity Characteristic

Correspondingly, the average convergence curve in Figure 4 can be discussed and compared with the average diversity characteristics as shown in Figure 5. Considering the degree of approaching to the zero value of diversity, it can be seen conversely from Figure 5 that the CBPSO comes first, the CPSO second, the BPSO third, and the PSO-GM last, respectively. Namely, a low diversity of CBPSO will result in getting trapped in some local optimums, whereas a higher diversity of PSO-
GM will contribute to an increase in the global search diversity compared with the three traditional PSO algorithms.

8. Conclusion, Discussion, and Suggestion

The outcome of this scholarly article shows that the hybrid method between the traditional PSO method and the Gaussian mutation operator succeeds in dealing with the lack of diversity problem in global search. Furthermore, it shows the superiority over the traditional PSO algorithms and other methods in terms of high quality solutions too.

Although applying mutation operator to the standard PSO technique succeeds in overcoming the diversity problem, it should be considered that if there is too much mutation, it may lead to unnecessary amount of the diversity amongst the particles. The direction of the future trend of developing PSO performance should take balancing the diversity between global and local exploration into account. For that reason, it should be find out the new concept for dealing with this difficulty. A successful method, A Diversity-Guided Particle Swarm Optimizer [32], which has been successfully applied to solve multimodal optimization problem, could be one acceptable alternative.

References


2004.


[27] M. Lovbjerg, T. K. Rasmussen and T. Krink, “Hybrid Particle Swarm Optimiser with


