International Journal of Innovative Computing, Information and Control
Volume **9**, Number **5**, May **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1919–1934

HPSOM: A HYBRID PARTICLE SWARM OPTIMIZATION ALGORITHM WITH GENETIC MUTATION

Ahmed A. A. Esmin[1,2] and Stan Matwin[2,3]

[1]Department of Computer Science
Federal University of Lavras
Lavras, MG 37200-000, Brazil
ahmed@dcc.ufla.br

[2]School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, ON, Canada
stan@eecs.uottawa.ca

[3]Institute of Computer Science
Polish Academy of Sciences
Warsaw, Poland

Received February 2012; revised June 2012

Abstract. *In this paper, a hybrid particle swarm optimization algorithm (HPSOM) that uses the mutation process to improve the standard particle swarm optimization (PSO) algorithm is presented. The main idea of the HPSOM is to integrate the PSO with genetic algorithm mutation method. As a result, the proposed algorithm has the automatic balance ability between global and local searching abilities. The validity of the HPSOM algorithm is tested for a variety of benchmark problems. Experimental results show empirically that the proposed method outperforms significantly the standard PSO methods in terms of convergence speed, solution quality, ability to find the global optimum, and solution stability.*
Keywords: PSO, Evolutionary algorithm, Hybrid PSO

1. Introduction.

Over the past decades, several population-based random optimization techniques, such as evolutionary algorithm and swarm intelligence optimization, have been widely employed to solve global optimization problems. Four well-known paradigms for evolutionary algorithms are genetic algorithms (GAs) [1], evolutionary programming (EP) [2], evolution strategies (ES) [3], and genetic programming (GP) [4]. These methods are motivated by natural evolution. The particle swarm optimization (PSO) method is a member of a wider class of swarm intelligence methods used for solving global optimization problems [5-7].

The PSO is a population-based stochastic optimization algorithm, which is inspired by the social behaviors of animals like fish schooling and bird flocking [6,7]. As a stochastic search scheme, PSO has characters of simple computation and rapid convergence capability. PSO was used in different industrial areas such as power systems [8,9-12], parameters learning of neural network [13,14], parameters optimization of fuzzy system [15], control [16], prediction [17], and modeling [17-19]. However, PSO exhibits some disadvantages; it sometimes is easy to be trapped in local optima (it is often called premature convergence), and the convergence rate decreased considerably in the later period of evolution.

It is observed that the PSO takes into consideration the global level of information to determine their search direction. Hence, the global and local best positions are computed at each time instance (iteration), and the output is the new direction of search. Once this direction is detected, it is followed by the cluster of birds. Note that PSO differs from the ordinary genetic algorithm because no crossover and mutation is considered. A few modifications are incorporated into the method, so it becomes more robust. Such modifications arise not because a bird may follow a different direction by itself but because the local information (position) this bird carries may be exploited as a permutation.

Different variants of the PSO algorithm were proposed. Some of these variants have been proposed to incorporate the capabilities of other evolutionary algorithms, such as hybrid versions of PSO or the adaptation of PSO parameters, creating the adaptive PSO versions. Many authors have considered incorporating selection, mutation, and crossover, as well as differential evolution, into the PSO algorithm. As a result, hybrid versions of PSO have been created and tested, including the hybrid of genetic algorithm and PSO, a genetic programming-based adaptable evolutionary hybrid PSO and evolutionary PSO [5,23-26]. Using another approach, Sun et al. [24] introduced quantum theory into PSO and propose a quantum-behaved PSO (QPSO) algorithm, which can be guaranteed theoretically to find optimal solution in search space, and Xi et al. [25] introduce an improved quantum-behaved particle swarm optimization algorithm with better convergence, weighted QPSO (WQPSO).

Some hybrid PSO algorithms were proposed by adding GAs' idea in [19,20]. The algorithm selects a certain number of particles according to the hybrid probabilities at each stage of iteration. The particles are randomly matched into couples. Each couple reproduces two children by crossover. Then, the children are used to replace their parents of the previous particles to keep the number of particles unchanged.

Mutation operators are an integral part of evolutionary computation techniques, preventing loss of diversity in a population of solutions, which allows a greater region of the search space to be covered. Therefore, the addition of mutation operators to PSO should enhance its global search ability and, thus, improve its performance. Recently, different hybrid PSOs with mutation operator have been proposed to overcome the drawback of PSO mentioned previously. PSO with Gaussian mutation (GPSO) [27,28], whose global searching ability is more efficient than the standard PSO, is proposed by Natsuki and Hitoshi. Ling et al. [29] proposed a hybrid PSO with wavelet mutation named HWPSO, in which a fine-tuning mutating method is used. In HWPSO, by performing mutation operation on the selected particle, the solution space should be explored more efficiently, and premature convergence is more likely to be avoided.

Although different mutation operators have been applied to enhance the achievements of PSO, the new mutation process needs to be further investigated so as to achieve better balance between global and local searching abilities and preserving the most important advantages of PSO when compared with GA, which is easy to implement, and there are few parameters to adjust.

In previous work, we propose and apply successfully hybrid particle swarm optimizer with static mutation to loss power optimization problem [8], and in reference [9], a brief description was presented.

This paper presents the hybrid particle swarm optimizer with mutation (HPSOM), by integrating the mutation process often used in GA into PSO and by varying the mutating space along the search (dynamic mutation). This process allows the search to escape from local optima and search in different zones of the search space. As a result, the proposed algorithm has the automatic balance ability between global and local searching abilities to guarantee the better convergence. Simulations show that the proposed hybrid algorithms possess better ability to find the global optimum than that of the standard PSO algorithm. The effectiveness of the HPSOM method is demonstrated by the experiments of benchmark problems with comparison to PSO and other three hybrid PSO algorithms (GPSO, HWPSO and WQPSO) [25,27-29].

The remainder of the paper is organized as follows. Section 2 describes the main aspects of the PSO method. Section 3 presents the standard genetic algorithms. The hybrid PSO is described in Section 4. In Section 5, some experiments are performed. Finally, Section 6 presents concluding remarks and directions of the future work.

2. **The Particle Swarm Optimization.** The PSO algorithm is a population-based optimization method that tries to find the optimal solution using a population of particles [6,7,10]. Each particle is an individual, and the swarm is composed of particles. Some of the attractive features of the PSO include ease of implementation and the fact that no gradient information is required. In PSO, the solution space of the problem is formulated as a search space. Each position in the search space is a potential solution of the problem. Particles cooperate to find the best position (best solution) in the search space (solution space). Each particle moves according to its velocity. At each iteration, the particle movement is computed as follows:

$$x_i(t+1) \leftarrow x_i(t) + v_i(t), \tag{1}$$

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \tag{2}$$

In Equations (1) and (2), $x_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, $pbest_i(t)$ is the best position found by particle itself so far, $gbest(t)$ is the best position found by the whole swarm so far, ω is an inertia weight scaling the previous time step velocity, c_1 and c_2 are two acceleration coefficients that scale the influence of the best personal position of the particle ($pbest_i(t)$) and the best global position ($gbest(t)$), and r_1 and r_2 are random variables within the range of 0 and 1.

Equations (3) and (4) define how the personal and global best values are updated at time t, respectively. It is assumed below that the swarm consists of s particles and the objective function f is used to calculate the fitness of the particles with a minimization task.

Thus, $i \in 1..s$

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & \text{if} \quad f(pbest_i(t)) \le f(x_i(t+1)) \\ x_i(t+1) & \text{if} \quad f(pbest_i(t)) > f(x_i(t+1)) \end{cases} \tag{3}$$

$$\begin{array}{l} gbest(t+1) = \min\{f(y), f(gbest(t))\} \\ \quad \text{where,} \quad y \in \{pbest_0(t), pbest_1(t), \ldots, pbest_s(t)\} \end{array} \tag{4}$$

The process of PSO is shown as Figure 1.

Equation (2) consists of three terms. The first term is the current speed of the particle, which shows its present state and has the ability to balance the whole and search a local part. The second term is the cognition term, which expresses the "thought" of the particle itself and causes the swarm to have a strong ability to search the whole and avoid a local minimum. The third term is called the social term; it reflects the information sharing among the swarm and among the particles, leading the particles toward known good solutions. Under the influence of these three terms, the particles can reach an effective and best position.

Two basic approaches to PSO exist based on the interpretation of the neighborhood of particles. Equation (2) reflects the global best (*gbest*) version of PSO where, for each particle, the neighborhood is simply the entire swarm. The social component then causes particles to be drawn toward the best particle in the swarm. In the local best (*lbest*)

```
Initialize a population of particles with random positions and velocities in the search space.
While (termination conditions are not met)
{
        For each particle $i$ do
            {
          Update the position of particle $i$ according to Equation (1).
          Update the velocity of particle $i$ according to Equation (2).
          Map the position of particle $i$ in the solution space and evaluate its fitness value
          according to the fitness function.
          Update $pbest_i(t)$ and $gbest_i(t)$ if necessary according to Equations (3) and (4).
            }
}
```

Figure 1. The PSO algorithm

PSO model, the swarm is divided into overlapping neighborhoods, and the best particle of each neighborhood is determined [5,6].

The stopping criterion (termination conditions) mentioned in the aforementioned algorithm depends on the type of problem being solved. Usually, the algorithm is run for a fixed number of iterations (objective function evaluations) or until a specified error bound is reached.

The description of how the algorithm works is as follows: initially, based on particle fitness information, one particle is identified as the best particle. Then, all the particles are accelerated in the direction of this particle but, at the same time, in the direction of their own best previously encountered solutions. Occasionally, the particles will overshoot their target, exploring the search space beyond the current best particles. All particles also have the chance to discover better particles en route, in which case, the other particles will change direction and head toward the new best particle. Because most functions have some continuity, chances are that a good solution will be surrounded by equally good, or better, solutions. By approaching the current best solution from different directions in search space, the chances that these neighboring solutions will be discovered by some of the particles are good [6,7].

3. The Standard Genetic Algorithms.

The GAs algorithms have been applied successfully to problems in many fields. The GAs are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms. Their basic principle is the maintenance of a population of solutions to a problem (genotypes) as encoded information individuals that evolve in time [21,22].

Generally, GA comprises three different phases of search: phase 1: creating an initial population; phase 2: evaluating a fitness function; phase 3: producing a new population.

A genetic search starts with a randomly generated initial population within which each individual is evaluated by means of a fitness function. Individuals in this and subsequent generations are duplicated or eliminated according to their fitness values. The further generations are created by applying GA operators. This eventually leads to a generation of high-performing individuals [21,22].

There are usually three operators in a typical genetic algorithm [22]: the first is the production operator (elitism), which makes one or more copies of any individual that posses a high fitness value; otherwise, the individual is eliminated from the solution pool. The second operator is the recombination (also known as the “crossover”) operator. This operator selects two individuals within the generation and a crossover site and carries out a swapping operation of the string bits to the right-hand side of the crossover site of both individuals. Crossover operations synthesize bits of knowledge gained from both parents exhibiting better-than-average performance. Thus, the probability of a better performing offspring is greatly enhanced. The third operator is the "mutation" operator. This operator acts as a background operator and is used to explore some of the invested points in the search space by randomly flipping a "bit" in a population of strings. Because frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its activation.

4. The Hybrid Algorithm.

Since the presentation of PSO [5-7], its performance has been investigated in several papers. The work presented in [19] describes the complex task of parameter selection in the PSO model. Comparisons between PSOs and the standard GA formulation have been carried out in [19], where the author points out that PSO performs well in the early iterations but presents problems in reaching a near-optimal solution.

The behavior of PSO in the gbest model presents some important aspects related to 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 (ω) and previous velocity are different from zero. If their previous 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 of the algorithm. In fact, this does not even guarantee that the algorithm has converged on a local minimum. It means that all the particles have converged at the best position discovered so far by the swarm. This phenomenon is known as stagnation [8,9,24].

This paper presents the HPSOM, by incorporating the mutation process often used in GA into PSO and by varying the mutating space along the search (dynamically). The stagnation is alleviated by this technique and introduces diversity into the population. This process allows the swarm to escape from the local optima and to search in different zones of the search space. As a result, the proposed algorithm has the automatic balance ability between global and local searching abilities and achieves better convergence. Figure 2 lists the pseudo-code for the basic HPSOM algorithm.

This process starts with the random choice of a particle in the swarm and moves to the different positions inside the search area. The mutation process can be applied for discrete and continuous PSO version:

- Discrete PSO: Although PSO is developed for continuous optimization problem, initially, there have been some reported studies that focused on discrete problem [30]. For the discrete version, we can use the binary mutation. For binary valued

```
begin
Create and initialise:
  While (stop condition is false)
  begin
        evaluation
        update velocity and position
        mutation
  end
end
```

FIGURE 2. The pseudo code for HPSOM algorithm

individuals, mutation means the flipping of variable values because every variable has only two states.

- Continuous PSO: The mutation process is employed using the following Equation (5):

$$mut(p_k) \leftarrow -p_k + \beta \tag{5}$$

where p_k is the random choice particles from the swarm, and β is randomly obtained within the range $[0\ ,\ 0.1 * (x_{\max} - x_{\min})]$, representing 0.1 times the length of the search space.

It is possible to use different equation to calculate the *mut* value. However, in this case, the mutating space is kept unchanged all the time throughout the search, and the space for the permutation of particles in PSO is also fixed. It can be improved by varying the mutating space along the search. The β can be linearly decreasing it from u to l times the length of the search space, using the following Equation (6) and shown in Figure 3:

$$\beta = u - \frac{u - l}{iter_{\max}} \cdot iter \tag{6}$$

where u is the upper range, l is the lower range, $iter_{\max}$ is the maximum number of iteration, and $iter$ is the current iteration number.

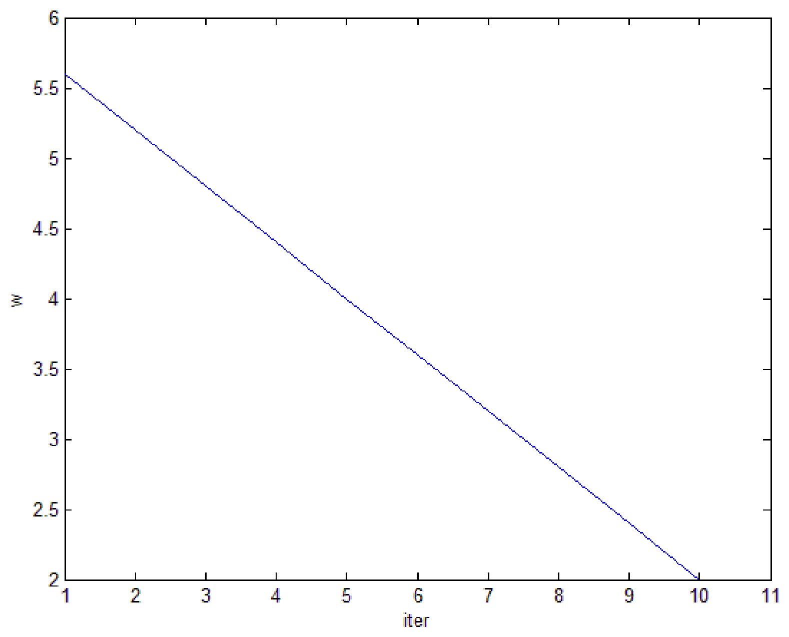
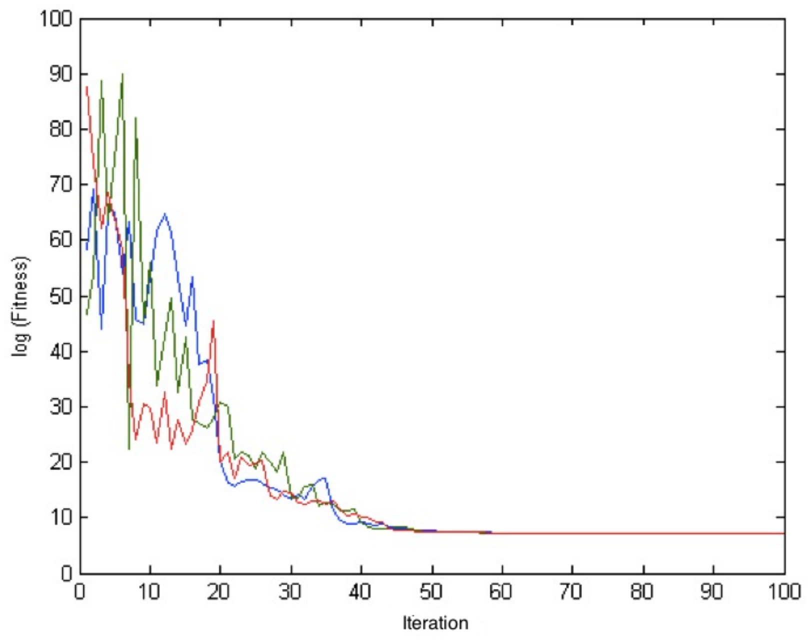


Figure 3. The β linearly decreasing

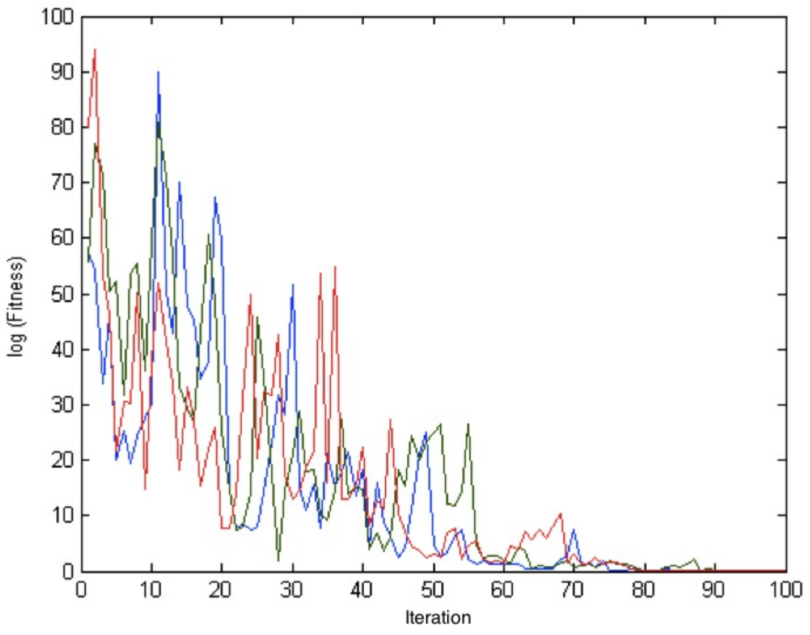
By applying the mutation process, the particle will move to the other side in the space and prevent the particles from being trapped in the local minima. Figures 4 and 5 show the behavior of the particles; we can observe that in HPSOM (Figures 4(b) and 5), the particles are more active and moving permanently to explore the search space compared with the particles in PSO (Figures 4(a) and 5).

5. **Experiment Results.**

To test the performance of HPSOM, four benchmark functions are used, all minimization problems. The first two functions were unimodal, whereas the last two were multimodal with many local minima. These four functions have been commonly used in other studies on particle swarm optimizers (e.g., [5,19,23,28]). The



(a)



(b)

FIGURE 4. (a) The behavior of the particles (pop = 3) in PSO for (F_1) stagnation iteration ∼60, (b) the behavior of the particles (pop = 3) in HPSOM for (F_1) more active and without stagnation

comparison process is stated as follows: first, we compared the original PSO and HPSOM algorithms with different configuration (varying the population size and dimensions of the functions), and then, we compare the performance of HPSOM with that of the other three hybrid algorithms, GPSO [28], WQPSO [25], and HWPSO [29] using fixed population

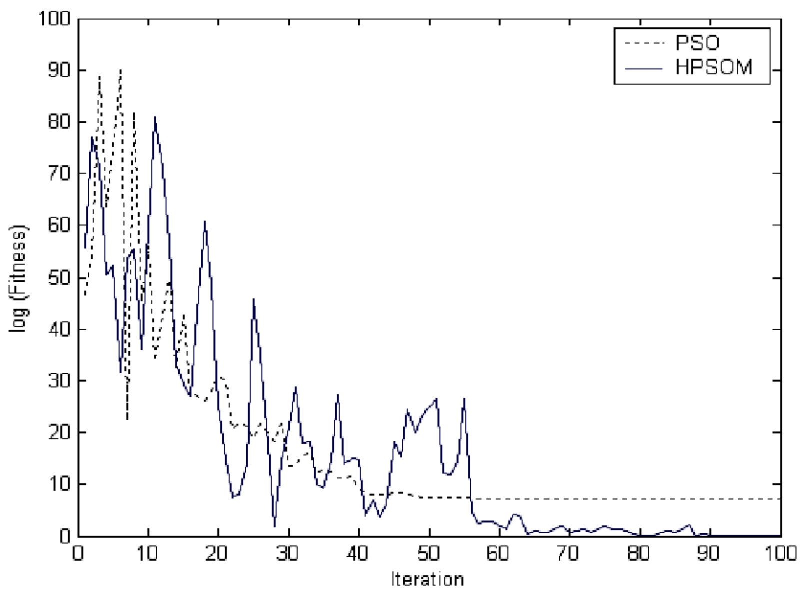


FIGURE 5. The behavior of the particles in PSO verses HPSOM (F_1)

size (20) with different function dimensions. These benchmark functions are described as follows:

- Spherical: The generalized sphere function is a very simple, unimodal function, with its global minimum located at $x = 0$, with $f(x) = 0$. This function has no interaction between its variables.

$$f_1(x) = \sum_{i=1}^{n} x_i^2$$

where x is a n dimensional real-valued vector, and x_i is the ith element of that vector.

- Rosenbrock: The second function is the generalized Rosenbrock function, a unimodal function, with significant interaction between some of the variables.

$$f_2(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$$

- Griewank: A multi-modal function with significant interaction between its variables, caused by the product term. The global minimum, $x = 0$, yields a function value of $f(x) = 0$.

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

- Rastrigin: The fourth and final test function is the generalized Rastrigin function, a multi-modal version of the spherical function characterized by deep local minima arranged as sinusoidal bumps. The global minimum is $f(x) = 0$, where $x = 0$. The variables of this function are independent.

$$f_4(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$$

The search space and initialization ranges for the experiments are listed in Table 1.

Table 1. Search space and initialisation ranges for test function

Function	Search space	Initialization ranges
f_1	$-100 \le x_i \le 100$	$50 \le x_i \le 100$
f_2	$-30 \le x_i \le 30$	$15 \le x_i \le 30$
f_3	$-600 \le x_i \le 600$	$300 \le x_i \le 600$
f_4	$-5.12 \le x_i \le 5.12$	$2.56 \le x_i \le 5.12$

5.1. **Experiments with PSO and HPSOM.**

All experiments consisted of 100 runs. The PSO and HPSOM parameters were set to the values $c_1 = c_2 = 2.0$, and a linearly decreasing inertia weight starting at 0.9 and ending at 0.4 was used. The maximum velocity ($V_{\max}$) of each particle was set to be half the length of the search space in one dimension.

We had 100 trial runs for every instance and recorded mean best fitness and standard variation. To investigate the scalability of the algorithm, different population sizes P are used for each function with different dimensions. The population sizes are 20, 40 and 80, and the maximum generation is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30 for four functions, respectively. Note that the HPSOM has the additional parameters related to the probability of mutation (p_m) that were set to 0.2, $u = 0.7$, and $l = 0.2$ for β. These parameters are chosen by simulations through experiment for good performance of all the functions.

The numerical results in Table 2 show that the HPSOM could reach the optimal solution with high precision. Because the sphere function has only a single optimal solution on origin, it usually is employed to test the local search ability of the algorithm. Thus, from the results, we can see that HPSOM has stronger local search ability than PSO. The Rosenbrock function is a mono-modal function, but its optimal solution lies in a narrow area that the particles are always apt to escape. Therefore, it is always used to test the local and global search ability of the algorithm. The experiment results on Rosenbrock function show that the HPSOM works better than the PSO algorithm. The advantage of HPSOM on PSO may be attributed to its local search ability as well as global search ability. Rastrigrin and Griewank functions are both multi-modal and usually tested for comparing the global search ability of the algorithm. On both Rastrigrin and Griewank functions, the HPSOM has better performance than the PSO algorithm.

Figures 6 to 9 provide the comparison of convergence processes of HPSOM and PSO in the above four benchmark functions averaged on 100 trial runs, when the population size is 20 and the maximum generation is 2000 according to the dimension 30 for four benchmarks. The coefficient β also decreases from 0.7 to 0.2 linearly. It can be found out that HPSOM has faster convergence speed than PSO.

5.2. **Experiments with HPSOM, GPSO, HWPSO and WQPSO.**

Table 6 and Figures 10 and 11 show the results obtained by comparing the HPSOM with other three hybrid algorithms, GPSO [27,28], WQPSO [25], and HWPSO [29].

We had 100 trial runs for every instance and recorded mean best fitness and standard variation. The population sizes is fixed to 20, and the maximum generation is set as 1000, 1500, and 2000 corresponding to the dimensions 10, 20, and 30 for four functions, respectively.

The probability of mutation for HPSOWM, HPSOM, and GPSO (p_m) and the shape parameter of the wavelet mutation (ζ_{wm}) are chosen by trial and error through experiments for good performance for all functions, and they are set at 0.2 and 0.1, respectively.

Table 2. The mean fitness value for sphere function (F_1)

P	D.	Iter.	Std. PSO		HPSOM	
			Mean best	St. Var	Mean best	St. Var
20	10	1000	3.16 e-20	6.24e-20	2.2400e-056	1.7300e-058
	20	1500	5.28e-11	1.55e-10	2.1449e-049	1.6891e-043
	30	2000	2.45e-06	7.17e-06	6.5764e-034	5.6809e-036
40	10	1000	3.11e-23	8.09e-23	6.7701e-89	6.23700e-090
	20	1500	4.16e-13	9.56e-14	8.9187e-055	7.17685e-057
	30	2000	2.25e-10	4.9e-10	4.8940e-043	2.15420e-044
80	10	1000	5.11e-28	2.76e-27	4.7651e-108	4.78300e-109
	20	1500	2.68e-17	5.23e-17	2.8762e-075	2.78267e-076
	30	2000	2.47e-11	7.9e-12	2.3345e-061	1.98310e-062

Table 3. The mean fitness for Rosenbork function (F_2)

P	Dim.	Iter.	Std. PSO		HPSOM	
			Mean best	St. Var	Mean best	St. Var
20	10	1000	72.8943	188.0902	6.9688	0.2730
	20	1500	251.9527	423.7695	17.3033	0.2629
	30	2000	321.3150	427.2838	27.5645	0.3514
40	10	1000	48.8258	125.9971	6.7682	0.2541
	20	1500	112.6777	224.3126	17.1315	0.2210
	30	2000	313.8127	455.1794	27.2902	0.2939
80	10	1000	31.3427	77.9028	6.5198	0.2921
	20	1500	78.1136	171.0607	16.9633	0.2271
	30	2000	146.8898	221.3618	27.1090	0.2624

Table 4. The mean fitness for Griewank function (F_3)

P	D.	Iter.	Std. PSO		HPSOM	
			Mean best	St. Var	Mean best	St. Var
20	10	1000	0.1016	0.0516	4.320507e-011	3.121672e-011
	20	1500	0.0282	0.0244	4.003760e-011	3.138521e-011
	30	2000	0.0123	0.0172	5.175657e-011	3.014372e-011
40	10	1000	0.0940	0.0496	3.596989e-011	2.835691e-011
	20	1500	0.0268	0.0273	4.320946e-011	2.866150e-011
	30	2000	0.0102	0.0129	4.387615e-011	3.238821e-011
80	10	1000	0.0863	0.0329	4.558345e-011	3.120239e-011
	20	1500	0.0278	0.0266	4.274249e-011	3.214253e-011
	30	2000	0.0129	0.0124	4.507656e-011	3.060030e-011

The numerical results in Table 6 show that the HPSOM and HWPSO could reach the optimal solution with high precision for f_1 and f_2. Because the sphere function has only a single optimal solution on origin, it usually is employed to test the local search ability of the algorithm. Thus, from the result, we can see that the four hybrids (HPSOM, HWPSO, GPSO and WQPS) algorithms have stronger local search ability than PSO. The Rosenbrock function is a mono-modal function, but its optimal solution lies in a narrow area that the particles are always apt to escape. Therefore, it is always used to test the

TABLE 5. The mean fitness for Rastrigrin function (F_4)

P	**D.**	**Iter.**	**Std. PSO**		**HPSOM**	
			Mean best	**St. Var**	**Mean best**	**St. Var**
20	10	1000	5.3666	2.3474	4.155822e-011	3.220292e-011
	20	1500	25.546	9.1794	4.140389e-011	3.240292e-011
	30	2000	52.942	12.533	4.800711e-011	3.188399e-011
40	10	1000	4.0124	1.9699	3.966093e-011	3.245508e-011
	20	1500	18.204	6.0499	4.403972e-011	3.301666e-011
	30	2000	39.685	10.586	4.447543e-011	3.075942e-011
80	10	1000	2.9020	1.3847	4.018133e-011	3.202956e-011
	20	1500	14.349	4.1029	4.044466e-011	3.237732e-011
	30	2000	34.732	8.8271	4.287983e-011	3.018866e-011

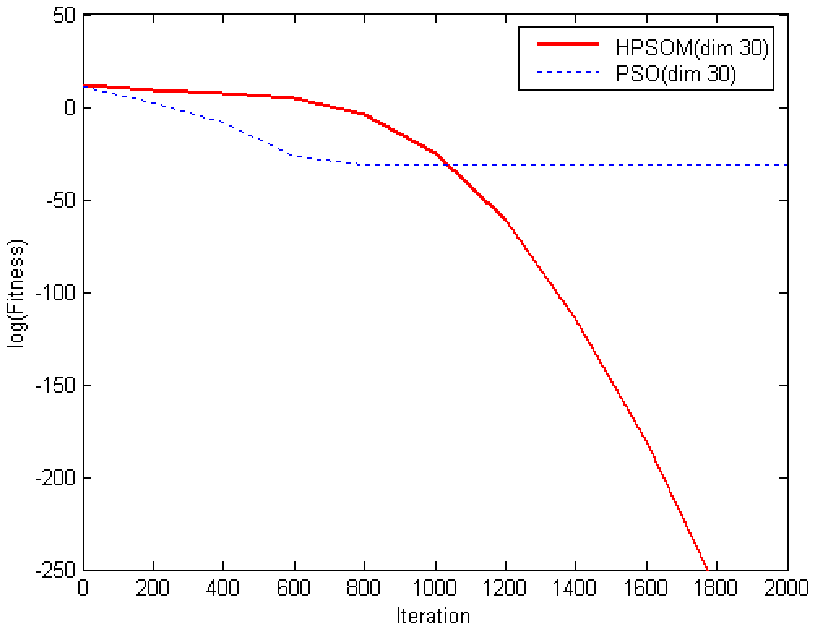


FIGURE 6. PSO versus HPSOM model for Spherical function (F_1)

local and global search ability of the algorithm. The experiment results on Rosenbrock function show that the HPSOM works better than the other hybrid algorithms. Rastrigrin function and Griewank function are both multi-modal and usually tested for comparing the global search ability of the algorithm. On Rastrigrin and Griewank functions, the HPSOM shows the best results on both functions. The advantage of HPSOM may be attributed to its local search ability and global search ability.

Figures 10 and 11 provide the comparison of convergence processes of the five algorithms (PSO, HPSOM, HWPSO, GPSO and WQPS) in the two benchmark functions (f_1, f_3) averaged on 100 trial runs, when the population size is 20, and the maximum generation is 2000 according to the function dimension, which is equal to 30. It can be found that in Figure 10, the HPSOM and HWPSO have the fastest convergence speed than the other algorithms (PSO, GPSO and WQPSO). However, in Figure 11 using the multimodal function, the HPSOM and the WQPSO have the fastest convergence speed than the other algorithms (PSO, GPSO and HWPSO).

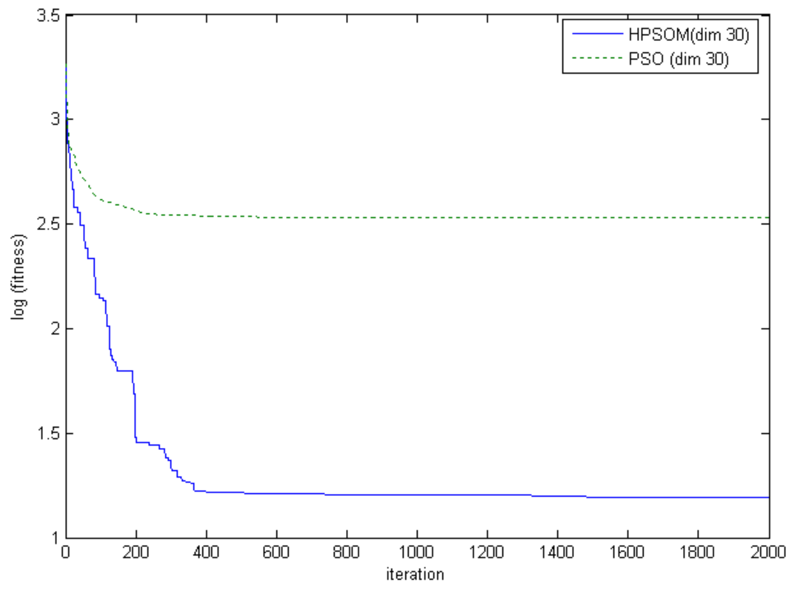


FIGURE 7. PSO versus HPSOM model for Rosenbrock function (F_2)

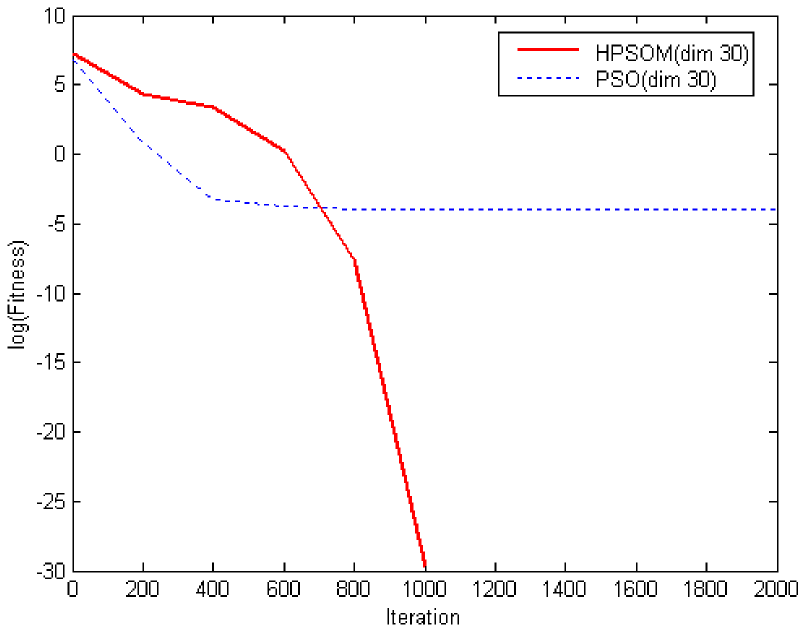


FIGURE 8. PSO versus HPSOM model for Griewank function (F_3)

6. **Conclusions.** This paper described the hybrid particle swarm algorithm (HPSOM), which exhibits the properties of both the PSO algorithm and the AG algorithm. The main idea is to integrate PSO and GA mutation. Our objective is to apply the new method to find a good balance between global and local searching abilities of PSO. Simulation was used to show that the HPSOM algorithm performs better than the standard PSO algorithm in minimizing functions.

The hybrid algorithms make use of the advantages of both GA and PSO methods; therefore, it is beneficial in solving optimal problems. The optima found by the hybrid

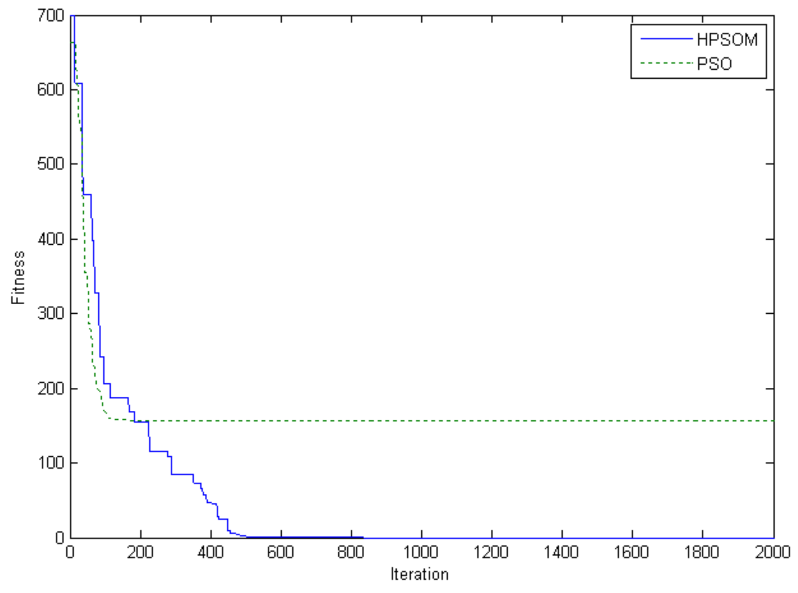


FIGURE 9. PSO versus HPSOM model for Rastrigin function (F_4)

TABLE 6. Comparison between different PSO methods for benchmark test functions

Function	**D.**	**Iter.**	**Average (Std. dev.)**				
			PSO	**HWPSO**	**GPSO**	**WQPSO**	**HPSOM**
f_1	10	1000	3.16e-20 (6.24e-20)	6.2868e-56 (1.506e-55)	2.1213e-43 (1.364e-42)	2.2922e-056 (1.5365e-058)	2.2400e-056 (1.7300e-058)
	20	1500	5.28e-11 (1.55e-10)	6.283e-45 (2.033e-44)	9.4791e-37 (2.802e-36)	2.9451e-040 (2.8717e-042)	2.1449e-049 (1.6891e-043)
	30	2000	2.45e-06 (7.17e-06)	3.794e-36 (1.406e-35)	2.496e-30 (9.781e-30)	3.9664e-033 (3.8435e-035)	6.5764e-034 (5.6809e-036)
f_2	10	1000	72.8943 (188.0902)	36.4736 (0.1844)	51.9761 (0.4737)	35.8436 (0.2843)	6.9688 (0.2730)
	20	1500	251.9527 (423.7695)	65.6678 (0.5870)	136.8782 (0.6417)	62.7696 (0.4860)	17.3033 (0.2210)
	30	2000	321.3150 (427.2838)	70.7275 (0.4813)	157.4707 (0.8287)	70.9525 (0.4283)	27.5645 (0.2939)
f_3	10	1000	0.1016 (0.0516)	0.13333 (0.33993)	0.13163 (0.37993)	5.6353e-004 (5.5093e-004)	4.320507e-011 (3.121672e-011)
	20	1500	0.0282 (0.0244)	2.9333 (2.7439)	2.2333 (2.3192)	2.1318e-004 (1.0402e-004)	4.003760e-011 (3.138521e-011)
	30	2000	0.0123 (0.0172)	9.2333 (6.1455)	8.1333 (6.0096)	2.1286e-004 (1.2425e-004)	5.175657e-011 (3.014372e-011)
f_4	10	1000	5.3666 (2.3474)	4.61 (2.5364)	4.1857 (2.2241)	4.0567 0.0094	4.155822e-011 (3.220292e-011)
	20	1500	25.546 (9.1794)	19.667 (6.7661)	16.815 (6.745)	12.1102 (0.0287)	4.140389e-011 (3.240292e-011)
	30	2000	52.942 (12.533)	44.723 (13.968)	34.956 (8.2597)	23.5593 (0.0713)	4.800711e-011 (3.188399e-011)

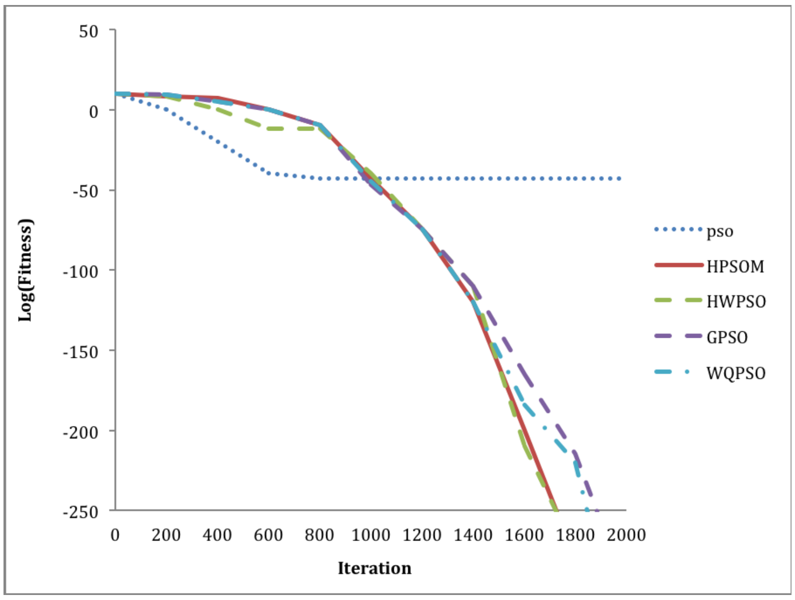


FIGURE 10. PSO versus HPSOM, HWPSO, GPSO, WQPSO function (F_1)

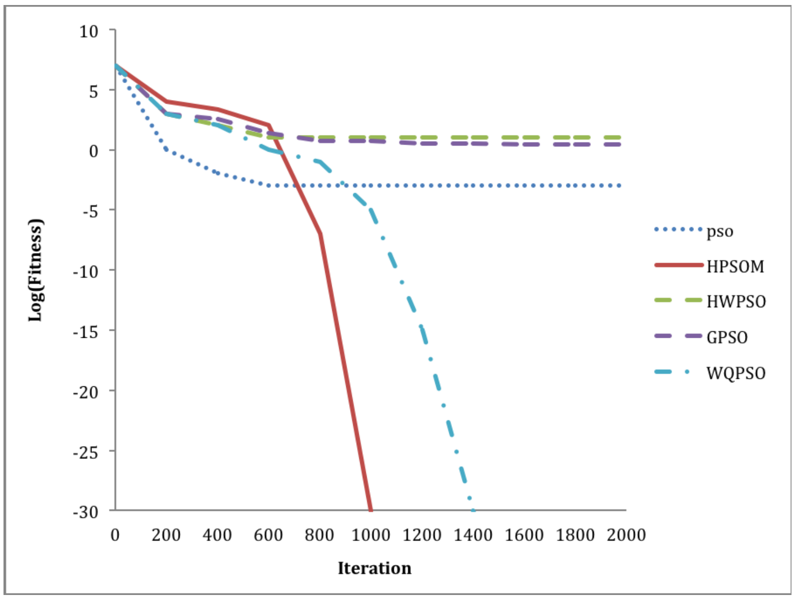


FIGURE 11. PSO versus HPSOM, HWPSO, GPSO, WQPSO function (F_3)

were better and present faster convergence. As a result, the proposed algorithm has the automatic balance ability between global and local searching abilities.

In our future work, we will be focused on finding out a more efficient parameter control for mutation process and, thus, further enhancing the performance of HPSOM.

Acknowledgment. We would like to thank CNPq, FAPEMIG (Brazilian agencies) and NSERC (Canada) for partial financial support. The authors also thank the anonymous reviewers for useful remarks and suggestions.

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.

[2] L. J. Fogel, Evolutionary programming in perspective: The top-down view, in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II and C. J. Robinson (eds.), IEEE Press, Piscataway, NJ, 1994.

[3] I. Rechenberg, Evolution strategy, in *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II and C. J. Robinson (eds.), IEEE Press, Piscataway, NJ, 1994.

[4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.

[5] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, *Proc. of the 1999 Congress of Evolutionary Computation*, vol.3, pp.1931-1938, 1999.

[6] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE Int. Conf. Neural Networks*, vol.4, pp.1942-1948, 1995.

[7] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.

[8] A. A. A. Esmin, G. Lambert-Torres and A. C. Zambroni, A hybrid particle swarm optimization applied to loss power minimization, *IEEE Trans. Power Systems*, vol.20, no.2, pp.859-866, 2005.

[9] A. A. A. Esmin, G. Lambert-Torres and G. B. Alvarenga, Hybrid evolutionary algorithm based on PSO and GA mutation, *Proc. of 6th International Conference on Hybrid Intelligent Systems*, pp.57-57, 2006.

[10] B. Zhao, C. X. Guo and Y. J. Cao, A multiagent-based particle swarm optimization approach for optimal reactive power dispatch, *Power Systems, IEEE Trans. Power Systems*, vol.20, no.2, pp.1070-1078, 2005.

[11] J. G. Vlachogiannis and K. Y. Lee, A comparative study on particle swarm optimization for optimal steady-state performance of power systems, *IEEE Trans. Power Systems*, vol.21, no.4, pp.1718-1728, 2006.

[12] C. M. Huang, C. J. Huang and M. L. Wang, A particle swarm optimization to identifying the ARMAX model for short-term load forecasting, *IEEE Trans. Power Systems*, vol.20, no.2, pp.1126-1133, 2005.

[13] C. F. Juang, A hybrid genetic algorithm and particle swarm optimization for recurrent network design, *IEEE Trans. on Systems, Man and Cybernetics: Part B: Cybernetics*, vol.34, no.2, pp.997-1006, 2004.

[14] C. Zhang, H. Shao and Y. Li, Particle swarm optimisation for evolving artificial neural network, *Proc. of IEEE Int. Conf. Systems, Man, and Cybernetics*, vol.4, pp.2487-2490, 2000.

[15] A. A. A. Esmin and G. Lambert-Torres, Fitting fuzzy membership functions using hybrid particle swarm optimization, *Proc. of IEEE International Conference on Fuzzy Systems*, pp.2112-2119, 2006.

[16] A. A. A. Esmin and G. Lambert-Torres, Application of particle swarm optimization to optimal power systems, *International Journal of Innovative Computing, Information and Control*, vol.8, no.3(A), pp.1705-1716, 2012.

[17] W. Liu, K. Wang, B. Sun and K. Shao, A hybrid particle swarm optimization algorithm for predicting the chaotic time series mechatronics and automation, *Proc. of IEEE Int. Conf. Mechatronics and Automation*, pp.2454-2458, 2006.

[18] R. Marinke, E. Araujo, Ld. S. Coelho and I. Matiko, Particle swarm optimization (PSO) applied to fuzzy modeling in a thermal-vacuum system, *Proc. of the 5th Int. Conf. Hybrid Intelligent Systems*, pp.67-72, 2005.

[19] P. J. Angeline, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences, *Proc. of the 7th International Conference on Evolutionary Programming VII*, pp.601-610, 1998.

[20] P. N. Suganthan, Particle swarm optimiser with neighbourhood operator, *Proc. of the 1999 Congress of Evolutionary Computation*, vol.3, pp.1958-1962, 1999.

[21] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand, New York, 1991.

[22] D. E. Goldberg, Genetic algorithms in search, *Optimization & Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.

[23] M. Løvbjerg, T. K. Rasmussen and T. Krink, Hybrid particle swarm optimiser with breeding and subpopulations, *Proc. of the Genetic and Evolutionary Computation Conference*, San Francisco, USA, 2001.

[24] J. Sun, W. B. Xu and B. Feng, A global search strategy of quantum-behaved particle swarm optimization, *Proc. of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, pp.111-116, 2004.

[25] M. Xi, J. Sun and W. Xu, An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, *Mathematics and Computation*, vol.205, no.2, pp.751-759, 2008.

[26] V. Miranda and N. Fonseca, EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems, *Proc. of Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition*, vol.2, pp.745-750, 2002.

[27] N. Hiqushi and H. Iba, Particle swarm optimization with gaussian mutation, *Proc. of IEEE Conf. Swam Intelligence Symposium*, pp.72-79, 2003.

[28] P. S. Andrews, An investigation into mutation operators for particle swarm optimization, *Proc. of IEEE Cong. on Evol. Comput.*, pp.1044-1051, 2006.

[29] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung and F. H. Leung, Hybrid particle swarm optimization with wavelet mutation and its industrial applications, *IEEE Trans. Syst. Man Cybernetics*, pp.743-763, 2008.

[30] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, *Proc. of IEEE Conference on Systems, Man, and Cybernetics*, Orlando, FA, USA, pp.4104-4109, 1997.