A Differential Beta Quantum-behaved Particle Swarm Optimization for Circular Antenna Array Design

Leandro dos Santos Coelho[1,5], Emerson Hochsteiner de Vasconcelos Segundo[2], Fabio Alessandro Guerra[3] and Viviana Cocco Mariani[4,6]

[1]*Pontifical Catholic University of Parana, Industrial and Systems Engineering Graduate Program (PPGEPS), Curitiba, Brazil*

[2]*Pontifical Catholic University of Parana, Mechanical Engineering Graduate Program (PPGEM), Curitiba, PR, Brazil*

[3]*Electricity Department, DPEL/DVSE/LACTEC, Institute of Technology for Development, Curitiba, PR, Brazil*

[4]*Pontifical Catholic University of Parana, Mechanical Engineering Graduate Program (PPGEM), Curitiba, Brazil*

[5]*Federal University of Parana (UFPR), Department of Electrical Engineering (DEE/PPGEE), Curitiba, PR, Brazil*

[6]*Federal University of Parana (UFPR), Department of Electrical Engineering (DEE), Curitiba, PR, Brazil*

Keywords: Metaheuristics, Particle Swarm Optimization, Quantum Mechanics, Circular Antenna Array Design.

Abstract: The classical particle swarm optimization (PSO) algorithm is inspired on biological behaviors such as the social behavior of bird flocking and fish schooling. In this context, many significant improvements related the updating formulas and new operators have been proposed to improve the performance of the PSO algorithm in the literature. On the other hand, recently, as an alternative to the classical PSO, a quantum-behaved particle swarm optimization (QPSO) algorithm was proposed. The contribution of this paper is linked with a modified QPSO based on beta probability distribution and mutation differential operator. The effectiveness of the proposed modified QPSO algorithm is demonstrated by solving three kinds of optimization problems including two benchmark functions and a circular antenna design problem.

1 INTRODUCTION

Particle swarm optimization (PSO) is a population-based algorithm of the swarm intelligence field proposed in Kennedy and Eberhart (1995) and Eberhart and Kennedy (1995). Its basic idea was based on simulation of simplified animal social behaviors. Over the years, PSO has gained significant popularity due to its simple structure and high performance. Furthermore, PSO has been shown to be efficient in a plethora of applications (Aote et al., 2013; Rini et al., 2011). However, many studies and several variants of the PSO algorithm (Sedighizadeh and Masehian, 2009; Eslami et al., 2012) have been done to improve the performance of PSO in continuous optimization.

Recently, novel optimization methods have been motivated from the concepts of quantum mechanics and computation (Han and Kim, 2002). One of the recent developments in PSO proposed by Sun et al. (2004a, 2004b) called quantum-behaved particle swarm optimization (QPSO). It is based on the perspective of quantum mechanics view rather than the Newtonian rules assumed in previous versions of PSO. QPSO is characterized by good search ability and fast convergence. Although QPSO is an efficient algorithm for solving continuous optimization problems, it is still necessary to pay enough attention to the inherent problem of possible premature.

To enhance the searching ability of PSO and accelerate its convergence, several studies (Coelho and Mariani, 2008; Fang et al., 2010; Sun et al., 2012; Mariani et al., 2012; Kamberaj, 2014) propose modifications in the QPSO.

Based on the mentioned considerations, we proposed in this paper a modified QPSO (MQPSO) based on beta probability distribution and mutation operator inspired by differential evolution paradigm. The differential evolution (DE) algorithm (Storn and Price, 1997; Das and Suganthan, 2011) is an evolutionary algorithm that uses a rather greedy and less stochastic approach to problem solving than do some evolutionary algorithms. The advantages of DE are simple structure, efficiency and robustness.

To judge the performance of the proposed algorithm, a set of two benchmark functions and a circular antenna design problem are solved. The results of simulations and convergence performance are compared with the classical PSO.

dos Santos Coelho L., Hochsteiner de Vasconcelos Segundo E., Alessandro Guerra F. and Cocco Mariani V..
A Differential Beta Quantum-behaved Particle Swarm Optimization for Circular Antenna Array Design.
DOI: 10.5220/0005070201920197
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2014), pages 192-197
ISBN: 978-989-758-052-9

The remainder of this paper is organized as follows: The fundamentals of the PSO, QPSO and MQPSO are provided in Sections 2 and 3, respectively. The description of the circular antenna array design problem is given in Section 4. Experiments on numerical optimization used to illustrate the efficiency of the proposed MQPSO are given in Section 5. Finally, a conclusion and future research are conducted in Section 6.

2 CLASSICAL PSO ALGORITHM

The classical PSO algorithm maintains a swarm of particles, where each particle represents a potential solution to the objective problem. The particles are initially placed at random positions in the search-space, moving (flying) in randomly defined directions in the n-dimensional search space. Their velocities are changed based on the results of the populational (*gbest*, global best) or personal (*pbest*, personal best) locations search, and they move toward the function optimum.

The procedure for implementing the global version of PSO is given by the following steps:

Step 1: Initialization of swarm: Initialize a population of particles with random positions and velocities in the n-dimensional problem space using uniform probability distribution function.

Step 2: Evaluation of particle's fitness: Evaluate each particle's fitness value.

Step 3: Comparison to pbest: Compare each particle's fitness with the particle's *pbest*. If the current value is better than *pbest*, then set the *pbest* value equal to the current value and the *pbest* location equal to the current location in n-dimensional space.

Step 4: Comparison to gbest: Compare the fitness with the population's overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.

Step 5: Updating of each particle's velocity and position: Change the velocity, v_i, and position of the particle, x_i, according to equations (1) and (2):

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot ud \cdot [p_i(t) - x_i(t)] + c_2 \cdot Ud \cdot [p_g(t) - x_i(t)] \quad (1)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (2)$$

where i=1,2,…,N indicates the number of particles of population; t=1,2,…,t_{max} indicates the generations (iterations); $v_i = [v_{i1}, v_{i2}, ..., v_{in}]^{\mathrm{T}}$ stands for the velocity of the i-th particle, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^{\mathrm{T}}$ stands for the position of the i-th particle of population, and $p_i = [p_{i1}, p_{i2}, ..., p_{in}]^{\mathrm{T}}$ represents the best previous position of the i-th particle. Positive constants c_1 and c_2 are referred to as the cognitive and social parameter, respectively. Index g represents the index of the best particle among all the particles in the swarm. The stochastic variables *ud* and *Ud* are random numbers generated uniformly distributed in [0,1]. Equation (2) represents the position update, according to its previous position and its velocity, considering $\Delta t = 1$.

Step 6: Repeating the evolutionary cycle: Return to *Step 2* until a stop criterion is met. In this paper the maximum number of generations is adopted.

In this paper, this is done by bounding a particle's velocity to the 20% of the full range of the search space, so the particle can at most move from one search space boundary to the other in one step.

PSO has some advantages over other similar optimization techniques such as genetic algorithm. It is easier to implement and needs fewer parameters to adjust. On the other hand, the original PSO is easily fall into local optima in many optimization problems.

Recent studies (Rini et al., 2011; Aote et al., 2013) have also attempted various ways to analyze and improve PSO. Proper selection of these w, c_1 and c_2 parameters can improve the convergence rate. However, the design of an effective method to select PSO's parameters using the relationship between w, c_1 and c_2 parameters can be a complex task.

3 QPSO ALGORITHM

The core idea of classical PSO is the exchange of information among the velocity, global best, local best, and current particles. In the QPSO, the velocity equation in the PSO algorithm is neglected. Experimental results performed on some well-known benchmark functions show that the QPSO method has better performance than the PSO method (see Sun et al. (2004a, 2004b, 2011)).

The probability of the particle's appearing in position x_i from probability density function $|\psi(x,t)|^2$, the form of which depends on the potential field the particle lies. In this context, each particle in a quantum state formulated by wavefunction $\psi(x, t)$.

Using the Monte Carlo method, the position of the particles can be obtained at iteration t+1 as (Sun et al., 2004a, 2004b):

If $k \geq 0.5$ then

$$x_{i,j}(t+1) = p_i(t) + \alpha \cdot \left| Mbest_j(t) - x_{i,j}(t) \right| \cdot \ln(1/u) \quad (3)$$

Else

$$x_{i,j}(t+1) = p_i(t) - \alpha \cdot \left| Mbest_j(t) - x_{i,j}(t) \right| \cdot \ln(1/u) \quad (4)$$

where $x_{i,j}(t+1)$ is the position for the j-th dimension of i-th particle in t-th generation (iteration); $Mbest_j(t)$ is the global point called *Mainstream Thought* or *Mean Best* (*Mbest*) for the j-th dimension; α is a design parameter called contraction-expansion coefficient in range [0,1]; u and k are numbers generated according to a uniform probability distribution in range [0,1]; and $p_i(t)$ is a local point. Trajectory analysis in Clerc and Kennedy (2002) demonstrated that the convergence of the PSO algorithm may be achieved if each particle converges to its local attractor.

The *Mainstream Thought* or *Mean Best* (*Mbest*) is defined as the mean of the *pbest* positions of all particles and it given by

$$Mbest_j(t) = \frac{1}{N} \sum_{j=1}^{N} p_{g,j}(t), \quad (5)$$

where g represents the index of the best particle among all the particles' swarm in j-th dimension. In this case, it is adopted

$$p_i(t) = \frac{c_1 \cdot p_{k,i} + c_2 \cdot p_{g,i}}{c_1 + c_2}, \quad (6)$$

where $p_{k,i}$ (*pbest*) represents the best previous i-th position of the k-th particle and $p_{g,i}$ (*gbest*) represents the i-th position of the best particle of the population. In the same form that the classical PSO, constants c_1 and c_2 are the cognitive and social components, respectively. The procedure for implementing the QPSO is given by the following steps:

Step 1: Initialization of swarm positions: Initialize a population of particles with random positions in the n dimensional problem space using a uniform probability distribution function.

Step 2: Evaluation of particle's fitness: Evaluate the fitness value of each particle.

Step 3: Comparison of each particle's fitness with its pbest (personal best): Compare each particle's fitness with the particle's *pbest*. If the current value is better than *pbest*, then set a novel *pbest* value equals to the current value and the *pbest* location equals to the current location in n-dimensional space.

Step 4: Comparison of each particle's fitness with its gbest (global best): Compare the fitness with the population's overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.

Step 5: Updating of global point: Calculate the *Mbest* using equation (5).

Step 6: Updating of particles' position: Change the position of the particles using equations (3) or (4), and (6).

Step 7: Repeating the evolutionary cycle: Loop to *Step 2* until a stopping criterion is met.

3.1 MQPSO Algorithm

QPSO may be trapped in local minima, because it easily loses the diversity of swarm during the search. To overcome the disadvantage of QPSO, a modified QPSO (MPQSO) combining strategy of QPSO and DE is introduced in this study.

The main point of the proposed MPQSO is to improve the global performance of the QPSO by using DE-inspired mutation operator. In the MQPSO, exploration behavior was enhanced at the early stage of searching so that more search space can be explored by particles. While at the later stage the MQPSO emphasized the exploitation to find the accurate optimum using a DE-inspired mutation operator. Furthermore, the MQPSO uses a linearly decreasing parameter to choice the global or local search during the generational cycle.

The *Steps* (1)-(5) of the QPSO are the same to the MQPSO. However, the *Step* (6) in MQPSO is given by following pseudocode:

$$p_i(t) = \frac{c_1 \cdot p_{k,i} + c_2 \cdot p_{g,i}}{c_1 + c_2}$$

If $r_3 > 0.8 - 0.6(t / t_{max})$

Modified approach:

$$\alpha \cdot = 0.27 \cdot betarnd(10 \cdot r_2, 10 \cdot r_3)$$

Uses Equation (3) or (4) to update $x_{i,j}(t+1)$

Else (uses the differential mutation)

$F = 0.6 + 0.3(t / t_{max})$

$$x_{i,j}(t+1) = x_{i,j}(t) + F \cdot \left[x_{s_1,j}(t) - x_{s_2,j}(t) \right]$$

End

where r_1, r_2 and r_3 are uniformly distributed random numbers bound within the range [0,1]; $s_1, s_2 \in \{1,2,...,N\}$, and *betarnd* is a number generated by a beta probability distribution, where $10 \cdot r_2$ and $10 \cdot r_3$ generate the shape parameters of the beta distribution. In this case, the Matlab's script called *betarnd.m* was employed.

The beta distribution is very flexible for modeling data that are measured in a continuous scale on the open interval (0,1) since its density has quite different shapes depending on the values of the two parameters that index the distribution.

4 CIRCULAR ANTENNA ARRAY DESIGN PROBLEM

Consider a circular array of N antenna elements spaced on a circle of radius r in the x-y plane. This is shown in Fig. 1 and the antenna elements are said to constitute a circular antenna array. The array factor for the circular array is written as follows (Das and Suganthan, 2010):

$$AF(\phi) = \sum_{i=1}^{Ne} I_i \cdot \exp[jar\cos\left(\phi - \phi_{ang}^{i}\right) - jar\cos\left(\phi_0 - \phi_{ang}^{i}\right) + \beta_i] \quad (6)$$

where Ne is the number of antennas, $\phi = 2\pi - 1$ is the angular position of the ith element on the x-y plane, $ar = Nd$ where a is the wave-number, d is the angular spacing between elements and r is the radius of the circle defined by the antenna array, ϕ_0 is the direction of maximum radiation, ϕ is the angle of incidence of the plane wave, I_i represents the amplitude excitation of the i-th element of the array and β_i is the phase excitation of the i-th element.

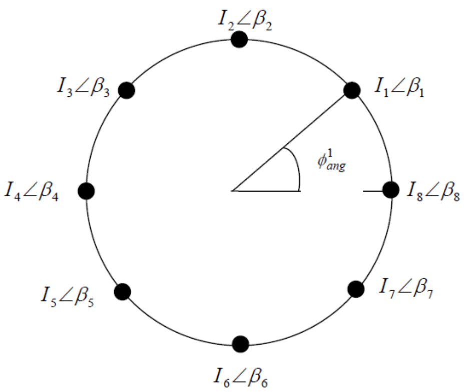

Figure 1: Geometry of circular antenna array (source: Das and Suganthan, 2010).

The purpose of the adopted optimization task is modify the current and phase excitations of the antenna elements and try to suppress side-lobes, minimize beamwidth and achieve null control at desired directions. The objective function f is taken as:

$$f = \left|AR\left(\varphi_{sll}, \vec{I}, \vec{\beta}, \varphi_0\right)\right| / \left|AR\left(\varphi_{\max}, \vec{I}, \vec{\beta}, \varphi_0\right)\right| + 1/DIR\left(\varphi_0, \vec{I}, \vec{\beta}\right) + \left|\varphi_0 - \varphi_{des}\right| + \sum_{k=1}^{num}\left|AR\left(\varphi_k, \vec{I}, \vec{\beta}, \varphi_0\right)\right| \quad (7)$$

where the first component attempts to suppress the sidelobes. φ_{sll} is the angle at which maximum sidelobe level is attained. The second component attempts to maximize directivity (DIR) of the array pattern. Nowadays directivity has become a very useful figure of merit for comparing array patterns. The third component strives to drive the maxima of the array pattern close to the desired maxima φ_{des}. The fourth component penalizes the objective function if sufficient null control is not achieved. num is the number of null control directions and φ_k specifies the kth null control direction.

We adopt the number of antenna elements in circular array as Ne=12 for a uniform separation distance of $d = 0.5\cdot\lambda$, where λ is the wavelength. The desired maxima φ_{des} is set to 180^{o} and $null$ = [50,120] in radians (no null control). In this case study, 6 excitation amplitudes (I_i) and 6 phase perturbations (β_i) are optimized. It is considered the following search space: $0.2 \le I_i \le 1$ and $-180^{o} \le \beta_i \le 180^{o}$, $i = 1,...,6$. The source code in Matlab of the circular array design benchmark problem is provided by IEEE-CEC (2011).

5 OPTIMIZATION RESULTS

In this section, two benchmark functions and the circular antenna array design are carried out to test the validity of the proposed MQPSO, and the results are compared with those of PSO and QPSO.

In the tests, the value of function is defined as the fitness function of algorithm. Due to the stochastic nature of the proposed approach, these four systems were repeatedly solved 25 times by the PSO and QPSO approaches.

The settings adopted in the tested PSO approaches for the benchmarks functions is the swarm size (population size) equal to 25 particles, 25 runs, and the stopping criterion is 10,000 generations. In the antenna case, 70 particles, 25 runs and the stopping criterion is 800 generations was adopted.

In terms of PSO setting, $c_1 = c_2 = 2.05$ and the inertia factor linear decreasing of 0.9 to 0.4 during the iterations is adopted. QPSO and MQPSO use a linearly decreasing contraction-expansion coefficient

(α) which starts at 1 and ends at 0.5.

5.1 Benchmark Functions

The Griewank function, first introduced in (Griewank, 1981), has been employed as a test function for global optimization algorithms in many papers. The function is defined as follows:

$$f_1(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad (8)$$

within search space given by $[-600,600]^n$. The global minimum value is 0 and the global minimum is located in the origin, but the function also has a very large number of local minima, which regularly distributed, exponentially increasing with *n*. It is similar to the Rastrigin function, but the number of local optima is larger in this case. Rosenbrock function (Rosenbrock, 1960) is non-convex, non-separable and quadratic function defined by

$$f_2(x) = \sum_{i=1}^{n-1}\left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right) \qquad (9)$$

with search space given by $[-30,30]^n$ and the global minimum value is 0.

Simulation results presented in Tables 1 and 2 (best results in boldface) showed that the MQPSO outperform the adopted PSO and QPSO on the basis of mean and standard deviation of the best objective function value of the total runs for the two benchmark functions with n = 10.

Table 1: Results of f_1 (Griewank function).

Index	PSO	QPSO	MQPSO
Maximum (Worst)	4.48×10^{-2}	2.20×10^{-1}	8.35×10^{-2}
Mean	2.74×10^{-2}	9.36×10^{-2}	$\mathbf{5.07\times10^{-2}}$
Minimum (Best)	1.11×10^{-2}	3.23×10^{-2}	$\mathbf{2.46\times10^{-2}}$
Standard Deviation	7.82×10^{-2}	4.68×10^{-2}	1.75×10^{-2}

Table 2: Results of f_2 (Rosenbrock function).

Index	PSO	QPSO	MQPSO
Maximum (Worst)	19.7794	12.6433	16.2680
Mean	10.3934	6.3008	**6.0639**
Minimum (Best)	5.22×10^{0}	4.28×10^{-1}	$\mathbf{5.90\times10^{-2}}$
Standard Deviation	3.5496	3.1631	3.7946

5.2 Circular Antenna Array Design

A comparison with PSO, QPSO and MQPSO of results presented in IEEE-CEC (2011) shows that the MQPSO approach provides quite encouraging results. As it is clear from Table 3 (best results in boldface), the MQPSO is able to find the global minimum and mean *f* values that outperform other 10 algorithms mentioned in IEEE-CEC (2011). The best result (minimum) using MQPSO presented *f*=-21.7586 is presented in Table 4.

Table 3: Results in terms of the best *f* values (25 runs).

Optimization method	*f* minimum	*B	*f* mean	#M
PSO	-20.7274	-	-16.9347	-
QPSO	-21.2292	-	-19.1884	-
MQPSO	-21.7586	-	-21.4360	-
GA-MPC	-21.8425	2	**-21.7022**	1
WI_DE	-21.8000	6	-21.7000	2
SAMODE	-21.8216	4	-21.6589	3
OXcoDE	**-21.8650**	1	-21.5910	4
ED-DE	-21.8320	3	-21.4210	5
EA-DE-MA	-21.7956	8	-21.2554	6
Mod_DE_LS	-21.7691	9	-21.0897	7
AdapDE171	-21.8084	5	-20.9583	8
mSBX-GA	-21.2545	11	-20.8860	9
DE-RHC	-20.5000	13	-18.3000	10
RGA	-21.0188	12	-17.2908	11
DE-Acr	-21.6010	10	-16.7560	12
ENSML_DE	-21.8000	6	-15.6000	13
CDASA	-19.0310	14	-13.5400	14

*B: ranking based on the **best** results in terms of *f* in CEC-2011
#M: ranking based on the **mean** results in terms of *f* in CEC-2011

Table 4: Best solution found using MQPSO.

Element *i*	Amplitude excitation (I_i)	Phase perturbation (β_i)
1	0.9993	-31.1885
2	0.3916	31.3933
3	0.4194	-89.5185
4	0.2026	-56.7929
5	0.4232	83.9948
6	0.6186	-16.7231
SSL (dB)	-21.8191	
DIR (dB)	10.0084	

6 CONCLUSIONS

QPSO is a complex nonlinear system, and accords to states superposition principle. In this paper, a MQPSO based on beta probability distribution and mutation differential operator is proposed and validated.

Simulation results illustrates that the incorporation of the beta probability distribution and mutation differential operation scheme enhances the search moves of a MQPSO by generating the more promising exemplars as the guidance particles. Furthermore, it provides the necessary trade-off between exploration and exploitation to global optimization. In this context, the simulation results show the effectiveness of our approach.

In future research, statistical significance tests to compare different optimization approaches with MQPSO will be carried out to monobjective and multiobjective cases.

REFERENCES

Aote, S. S., Raghuwanshi, M. M. and Malik, L. (2013). Brief review on particle swarm optimization: limitations & future directions. *International Journal of Computer Science Engineering (IJCSE)* 2(5): 196-200.

Clerc, M. and Kennedy, J. F. (2002). The particle swarm: explosion, stability and convergence in a multi-dimensional complex space. *IEEE Transactions on Evolutionary Computation* 6(1): 58-73.

Coelho, L. S. and Mariani, V. C. (2008). Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects. *Energy Conversion and Management* 49(11): 3080-3085.

Das, S. and Suganthan, P. N. (2010). Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, *Technical Report*, Case study #9: Circular Antenna Array Design Problem, December.

Das, S. and Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* 15(1): 4-31.

Eberhart, R. C. and Kennedy, J. F. (1995). A new optimizer using particle swarm optimization. In *Proceedings of the International Symposium on Micro Machine and Human Science*, Japan, 39-45.

Eslami, M., Sharref, H., Khajehzadeh, M. and Mohamed, A. (2012). A survey of the state of the art in particle swarm optimization. *Research Journal of Applied Sciences, Engineering and Technology* 4(9): 1181-1197.

Fang, W., Sun, J., Ding, Y., Wu, X., and Xu, W. (2010). A review of quantum-behaved particle swarm optimization. *IETE Technical Review* 27(4): 336-348.

Griewank A.O. (1981). Generalized descent for global optimization. *Journal of Optimization Theory and Applications* 34: 11-39.

Han, K. H. and Kim, J.H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation* 6(6): 580-593.

IEEE-CEC, New Orleans, USA (2011). Competition on testing evolutionary algorithms on real-world numerical optimization problems, Detailed results and software in Matlab, *http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm.*

Kamberaj, H. (2014). Q-Gaussian swarm quantum particle intelligence on predicting global minimum of potential energy function. *Applied Mathematics and Computation* 229: 94-106.

Kennedy, J. F. and Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of the IEEE Conference on Neural Networks*, Perth, Australia, 1942-1948.

Mariani, V. C., Duck, A. R. K., Guerra, F. A., Coelho, L. S., and Rao, R. V. (2012). A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers. *Applied Thermal Engineering* 42: 119-128.

Rini, D. P., Shamsuddin, S. M., and Yuhaniz, S. S. (2011). Particle swarm optimization: technique, system and challenges. *International Journal of Computer Applications* 14(1): 19-27.

Rosenbrock, H. H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal* 3: 175-184.

Sedighizadeh, D. and Masehian, E. (2009). An particle swarm optimization method, taxonomy and applications. *International Journal of Computer Theory and Engineering* 5: 486-502.

Storn, R. and Price, K. (1997). Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4): 341-359.

Sun, J., Feng, B., and Xu, W. B. (2004a). Particle swarm optimization with particles having quantum behavior. In *Proceedings of Congress on Evolutionary Computation*, Portland, Oregon, USA, 325-331.

Sun, J., Lai, C. -H., and Wu, X. -J. (2011). *Particle swarm optimisation: classical and quantum perspectives*. Boca Raton, USA, CRC Press.

Sun, J., Wu, X., Palade, V., Fang, W., Lai, C. -H., and Xu, W. (2012). Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences* 193: 81-103.

Sun, J., Xu, W. B., and Feng, B. (2004b). A global search strategy of quantum-behaved particle swarm optimization. In *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, 111-116.