Beta Damping Quantum Behaved Particle Swarm Optimization

Tarek M. Elbarbary, Hesham A. Hefny, Atef abel Moneim

Institute of Statistical Studies and Research, Cairo University,
Giza, Egypt
tareqbarbary@yahoo.com, hehefny@ieee.org

Abstract

In solving engineering optimization problems; very special cases of the problems can be solved in polynomial time while, most of them are considered as hard combinatorial optimization problems "NP-hard", and most of the solution algorithms for these problems are based on numerical linear and nonlinear programming methods that require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point. Quantum-behaved particle swarm optimization (QPSO) algorithm is a global convergence guaranteed algorithms, which outperforms original PSO in search ability but has fewer parameters to control. In this paper, we propose an improved quantum-behaved particle swarm optimization with new beta value according to fitness values of the particles. It is shown that the improved QPSO has faster local convergence speed, resulting in better balance between the global and local searching of the algorithm, and thus generating good performance. The proposed improved QPSO, called beta damping algorithm (MQPSO) approach for engineering optimization problems with both continuous and discrete designed variables, is tested on several benchmark functions and compared with QPSO. The experiment results show the superiority of MQPSO.

Keywords: *Quantum particle swarm optimization, beta damping, fitness function*

1. Introduction

Over the past several decades, population-based random optimization techniques, such as evolutionary algorithm and swarm intelligence optimization, have been widely employed to solve global optimization (GO) problems. Four well-known paradigms for evolutionary algorithms are genetic algorithms (GA), evolutionary programming (EP), evolution strategies (ES) and genetic programming (GP). 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 GO problems. [1][5] .The method was originally proposed by Kennedy as a simulation of social behavior of bird flock and was first introduced as an optimization method in 1995. Instead of using evolutionary operators to manipulate the individuals as in other evolutionary algorithms, PSO relies on the exchange of information between individuals. Each particle in PSO flies in search space with a velocity, which is dynamically adjusted according to its own former information. Since 1995, many attempts

have been made to improve the performance of the PSO [8]. As far as the PSO itself concerned, however, it is not a global optimization algorithm, as has been demonstrated by Van den Bergh, Sun introduce quantum theory into PSO and propose a quantum-behaved PSO (QPSO) algorithm, which can be guaranteed theoretically to find optimal solution in search space. The experiment results on some widely used benchmark functions show that the QPSO works better than standard PSO and should be a promising algorithm.[2][6].

Quantum behaved particle swarm optimization (QPSO) algorithm is a global convergence guaranteed algorithms, which outperforms original PSO in search ability but has fewer parameters to control [6]. In this paper, we propose an improved quantum-behaved particle swarm optimization with weighted mean best position according to fitness values of the particles. To balance the global and local searching abilities, we introduce a new parameter in calculating the mean best position in QPSO to render the importance of particles in population when they are evolving, and thus proposed novel quantum-behaved particle swarm optimization algorithm, beta damping QPSO (MQPSO). It is shown that the improved QPSO has faster local convergence speed, resulting in better balance between the global and local searching of the algorithm, and thus generating good performance. The proposed improved QPSO, called Modified QPSO (MQPSO) algorithm, is tested on several benchmark functions and compared with QPSO and standard PSO. The experiment results show the superiority of MQPSO.

The rest part of the paper is organized as follows. In Section 2, a brief introduction of quantum behaved PSO is given. The Proposed Novel Approach of Quantum Particle Swarm Optimization is introduced in Section 3. In Section 4, we propose comparative analysis between QPSO and Modified QPSO and show how to balance the searching abilities to guarantee the better convergence speed of particles. Some experiments result on benchmark functions and discussions are presented. Finally, the paper is concluded in Section 6.

2. Quantum PSO Algorithms

2.1 Overview of Quantum PSO Algorithms

QPSO is a kind of probabilistic algorithm and the iterative equation of QPSO is very different from that of PSO. Besides, unlike PSO, QPSO needs no velocity vectors for particles, and has fewer parameters to adjust making it easier to implement. The QPSO algorithm has been shown to offer good performance in solving a wide range of continuous optimization problems and many efficient strategies have been proposed to improve the algorithm. QPSO was inspired by analysis of the convergence of the traditional PSO and quantum system. In the quantum physics, the state of a particle with momentum and energy can be depicted by its wave function Ψ (x, t). Then in QPSO, we hypothesize that each particle is in a quantum state and is formulated by its wave function Ψ (x, t) instead of the position and velocity which are in PSO. According to the statistical significance of the wave function, the probability of a particle's appearing in a certain position can be obtained from

the probability density function $|\Psi(x, t)|^2$ [2][8]. And then the probability distribution function of the particle's position can be calculated through the probability density function. The particle's position is updated according to the following equation:

$$x = p \pm \frac{L}{2}\ln\left(\frac{1}{u}\right) \qquad (1)$$

Where u is a random number uniformly distributed in (0, 1) and L is the most important variable which determines the search scope of each particle, and it is given by the distance between the particle's current position and point p) [1][3].

$$L = 2\beta.|p - x| \qquad (2)$$

Where parameter β is called the contraction-expansion (CE) coefficient, which can be tuned to control the convergence speed of the algorithms [4].

$$\beta = \begin{cases} \beta_{min} + \dfrac{(\beta_{max} - \beta_{min})(f - f_{min})}{f_{avg} - f_{min}} & f \leq f_{avg} \\ \beta_{max} & f \geq f_{avg} \end{cases} \qquad (3)$$

Where β_m and β_n denote the maximum and minimum of β, respectively, f is the current objective value of the particle, f_a and f_m are the average and minimum objective values of all particles, respectively. Then we get the position update equation as:

$$x = p \pm \beta.|p - x|.\ln\left(\frac{1}{u}\right) \qquad (4)$$

We call the PSO algorithm with last position update equation as quantum delta-potential-well-based PSO (QDPSO) algorithm. A global point called mainstream thought or mean best position of the population is introduced into PSO. The global point, denoted m *best*, is defined as the mean of the p *best* positions of all particles [1][3].

$$mbest = \frac{1}{M} \sum P_i \qquad (5)$$

M is the population size and pi is the best position recorded by particle i. and the position is given by :

$$x = p \pm \beta.|mbest - x|.\ln\left(\frac{1}{u}\right) \qquad (6)$$

Generally, we call the PSO algorithm with the last equation quantum-behaved particle swarm optimization (QPSO). The most commonly used control strategy of β is to initially setting it to 1.0 and reducing it linearly to 0.5.

2.2- Implementation of QPSO algorithm procedures

Initialize an array of particles with random positions inside the problem space. and population size, the positions, and the dimensions of the particles The user must choose the key parameters that control the QPSO, namely population size of particles, boundary constraints of optimization variables, cognitive component (c1), social component (c2), contraction-expansion coefficient (β) and the stop criterion (t max) [3]. Then evaluate the fitness value of each particle and determine the mean best position among the particles by means of

$$mbest = \frac{1}{M} \sum_{i=1}^{M} p_i$$

Compare each particle's fitness with the particle's *p best*. Evaluate the desired objective function (e.g. minimization) for each particle and compare with the previous best values. If the current value is better than the previous best value, then set the best value to the current value, i.e. if f (Xi) < f (Pi), then Xi = Pi. Determine the current global position minimum among the particle's best positions, i.e.

$$P_{gbest} = \min_{1 \prec x \prec m} P_i \quad (7)$$

Compare the fitness with the population's overall previous best. If the current value is better than g best, then reset g best to the current particle's array index and value. Compare the current global position with the previous global best position. [3]. Update the position of the particle according to the next equation:

$$x = p \pm \beta . |mbest - x| . \ln\left(\frac{1}{u}\right)$$

Repeat until a stop criterion is satisfied or pre-specified number of iterations is completed. The stopping criterion can be one or more of the following the first is fixed number of iterations or fixed value of the fitness function or fixed number of iterations without change in the performance finally an accepted best error value (cut off)

Initialize population x_i of all particles randomly
Initialize p_i with the corresponding initial position x_i
Initialize $p_g = argmin\ (f(p_i))$
Do while the criteria is not met
      Calculate $mbest$ according to equation 6
      For each particle x_i
            Calculate β according to equation 2
            For   d   = 1 to dimension D
$$u = rand\ (0,1)$$
$$if\ rand\ (0,1)\ > 0.5$$
$$x = p - \beta\ .|mbest\ -x\ |.\ln\left(\tfrac{1}{u}\right)$$
    Else
$$x = p + \beta\ .|mbest\ -x\ |.\ln\left(\tfrac{1}{u}\right)$$
    End
      End
      Fitness evaluation for x_i
      If $f(x_i) < f(p_i)$
$$p_i = x_i$$
$$if\ f(p_i) < f(p_g)\ ,\ p_g = p_i$$
    End
  End for
End

Fig (1) Pseudo code of QPSO

Hybridization is a usually used method to improve the performance of QPSO, which aims to combine the desirable properties of different approaches to mitigate QPSO weakness. The mutation operator is one of the mechanisms of genetic algorithms (GAs)[and it provides diversity in the search and helps in exploring the undiscovered search place [5]. This mechanism can also help the QPSO algorithm to escape from the local minima. *g best* and *m best* in QPSO were mutated with Cauchy distribution the amend strategy based on annealing was adopted by the scale parameter of the mutation operator to increase the self-adaptive capability of the improved algorithm. A kind of mutation based on chaotic sequences was applied in QPSO, which was a powerful strategy to diversify the QPSO population and improve the QPSO's performance in preventing premature convergence to local minima. A set of mutation operators used in QPSO were analyzed and the performance was compared by evaluating a set of benchmark functions.

3. Proposed Novel Approach of QPSO

This section presents a new QPSO called MQPSO which uses the quantum theory of mechanics to govern the movement of swarm particles along with an interpolation (quadratic interpolation) based recombination operator. The concept of quadratic interpolation as a recombination operator will introduced, for improving the performance of classical PSO,

where it gave significantly good results. This motivated us to apply this concept for QPSO also to improve its performance. The proposed M-QPSO algorithm is a simple and modified version of QPSO in which we have introduced the concept of recombination. The MQPSO algorithm starts like the usual QPSO. At the end of iteration the quadratic interpolation recombination operator is invoked to generate a new swarm particle. The new particle is accepted in the swarm only if it is better than the worst particle the particle having maximum fitness, present in the swarm. This process is repeated iteratively until a better solution is obtained. The quadratic crossover operator is a nonlinear operator which produces a new solution vector lying at the point of minima of the quadratic curve passing through the three selected swarm particles The numerical results show that the QPSO works better than PSO for all functions that have several beaks. For benchmark functions, the performance of QPSO is superior to PSO[1][8]. However, our preliminary experimental results show that when the contraction-expansion coefficient varies from 1.2 to 0.5, QPSO outperforms PSO on all functions. The results also show that introducing the constriction factor into PSO does not improve the algorithm in general. We note that QPSO is better than PSO for the problems which have several minima and escape from the local minimum but we note that the performance of QPSO is decrease for the problem which have minimum not zero and zero (origin). So we modified the beta step value according to the following algorithm.

1. Introducing a damping oscillating equation for β

$$\beta = \left(\frac{1}{(n_{Iter} + \beta_{wavelenth} \div 2)} \right)^{0.25} \times \left(\cos\left(\frac{n_{Iter}}{\beta_{wavelenth} \times \pi} \right) + 1 \right) + \frac{1}{3} \qquad (8)$$

2. Increase diversity after certain number of iteration without finding better new solution :

$$\beta = \log\left(\left| \frac{Global\ best\ Error}{Error\ Cutoff} \right| \right) \qquad (9)$$

3. M_{best} Is considered a particle in the swarm, hence M_{best} is evaluated with fitness function.

4. Avoiding loss off particle movement if $L_{(i,j)}$ equal zero by introducing random noise ε to the position equation to be :

$$x_{(i,j)} = m_{best} \pm \left[0.5 \times L_{(i,j)} \times \log\left(\frac{1}{u} \right) + \varepsilon \right] \times \beta \qquad (10)$$

Where $\varepsilon = (R \times \beta)$ and R is random number in range of [0.1 % of domain]. The next figure presents the values of beta according to the previous steps with the number of iteration.
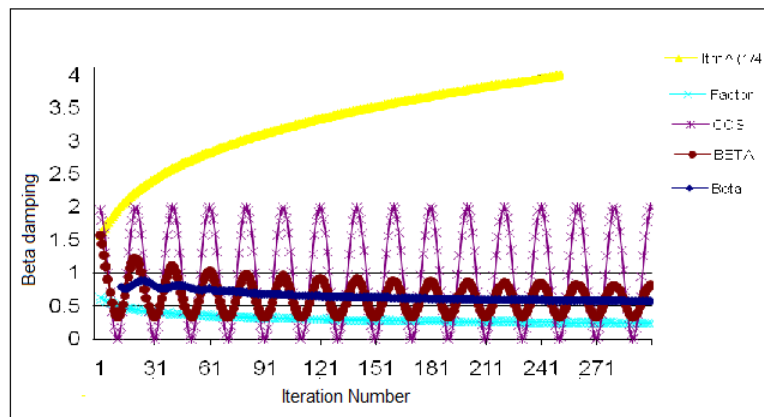
Fig 2 Beta damping modification

In Modified QPSO algorithm a nonlinearly decreasing contraction expansion coefficient (β) is used which starts by large value and decrease as in the previous figure. The acceleration constants are taken as from 1.7 to 1.8. In order to check the compatibility of the proposed MQPSO algorithm we have tested it on two out of eight benchmark problems (uncontained) the first function (De Jong function) is unimodal function with best solution at the origin and the second problem (Ackley function) is multimodal problem with different solution. All the test problems are highly scalable in nature and are considered to be starting point for checking the credibility of any optimization algorithm. There is not much literature available in which MQPSO is used extensively for solving a variety of test problems. Therefore, for the present study, we have considered two test problems out of which the first problem is the ones that have been tested with some variants of MQPSO. The remaining problem we have solved with our version and QPSO. To testify the applicability of M-QPSO to NLP problems, we performed two groups of experiments on unconstrained and constrained benchmarks.

4. Comparative Analysis between QPSO and MQPSO

This paper introduces a new version of QPSO to increase the performance to solve linear and nonlinear problems. The proposed version depends on modified the beta contraction expansion coefficient which can be tuned to control the convergence speed of the algorithms. The control method of β is vital to the convergence rate and performance of the algorithm. The philosophy of changing the value of beta to provide damping behavior to the movement of the particle to allow escaping from the local minima to other best positions [11].

The paper proposed a more efficient method which called beta damping process. In order to determine whether the modified QPSO algorithm can be more effective than PSO and other forms of PSO, QPSO, and three variants of PSO, which can be classified into the probabilistic algorithms, are used to optimize a set of benchmark functions. The benchmark functions are given in Table 1 [7][9].

Table 1 the benchmark functions

Function	Initial Range	Optima	Optimal Position
Sphere $f(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	$f(x^*)=0$	$x^* =(0,....,0)$
Rosenbrock $f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-100, 100]^n$	$f(x^*)=0$	$x^* =(1,....,1)$
Schwefel function $f(x) = 418.9829n - \sum_{i=1}^{n}(x_i sin(\sqrt{\|x_i\|}))$	$[-500, 500]^n$	$f(x^*)=0$	$x^* =(420,.., 420)$
Rastrigin $f(x) = \sum_{i=1}^{n}(x_i^2 - 10\,cos(2\pi x_i)) + 10n$	$[-10, 10]^n$	$f(x^*)=0$	$x^* =(0,....,0)$
Ackley $f(x) = -20 + \exp\left(-0.2\sqrt{\left(\frac{1}{D}\sum_{i=1}^{D}x_i^2\right)}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)$	$[-32, 32]^n$	$f(x^*)=0$	$x^* =(0,....,0)$
Griewank $f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^n$	$f(x^*)=0$	$x^* =(0,....,0)$

To test the performance of the proposed QPSO (MQPSO), two benchmark functions listed in Table 1 are used here for comparison with QPSO algorithm. These functions are all minimization problems with different minimum values. In our experiments the fitness value is set as function value and the neighborhood of a particle is the whole population. We had 100 trial runs for every instance and recorded mean best fitness and standard deviation.

In order to investigate the scalability of the algorithm. From the results we have the following comments for each of the used benchmark functions:

Sphere function

It is a unimodal function with global minimum located at x = (0, 0,....,0) with f(x) = 0 as shown in table 1. It is clear that the MQPSO algorithm is superior to the SPSO and QPSO algorithms as shown in table 2 , figures 3 and 4. The minimum best point obtained as a near global minimum found by the proposed MQPSO algorithm is given in table 6, which is the coordinates of the best point in the thirty dimensional search space. The fitness, i.e. function value, of such a point is given in the table 6 is 8.74741E-11 with stander deviation 1.02E-03.

Table 2 Sphere Function Results

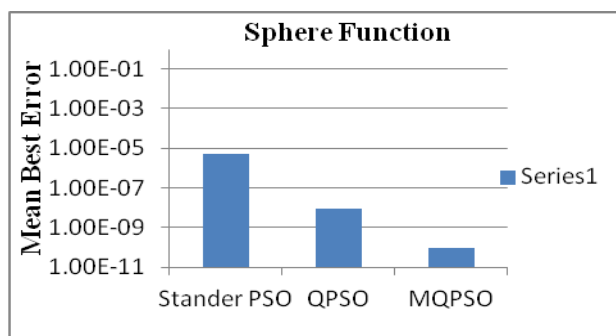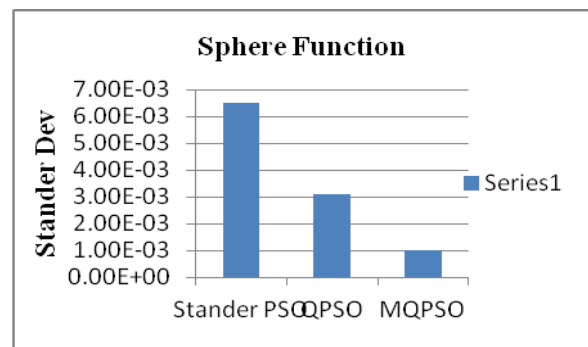| Sphere Fun	Stander PSO	QPSO	MQPSO
Best Error	5.08E-06	8.99E-09	8.75E-11
Stander Div	6.49E-03	3.11E-03	1.02E-03

Fig 3 Mean Best Error Sphere Function

Fig 4 Stander Deviation for Sphere Function

Rosenbrock function

It is another unimodal function with global minimum located at x = (l,l,....,l) with f(x) = 0 as shown in table 1. The comparison results of the three algorithms are illustrated in table 3, figures 5 and 6. It is found that the MQPSO algorithm fails to get the best error as given in the appendix on the other hand; the MQPSO is superior to the other algorithms in terms of the other performance measures. The best point obtained as a near global minimum found by the proposed MQPSO algorithm is given in the third column in appendix A, which is its coordinates in the thirty dimensional search space. The function value, stander deviation of such a point is given in the last row of table 6 as: 1.25060   and 9.29E-01 respectively.

Table 3 Rosenbrock Function Results

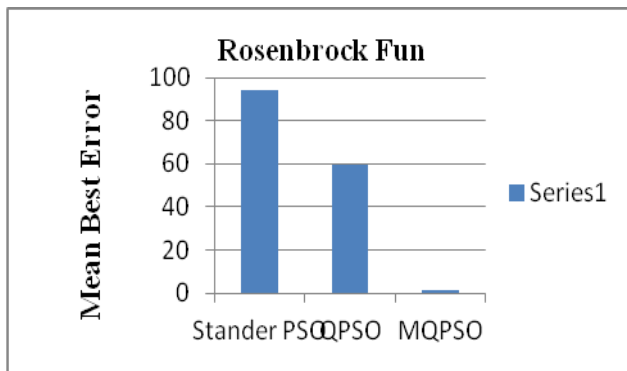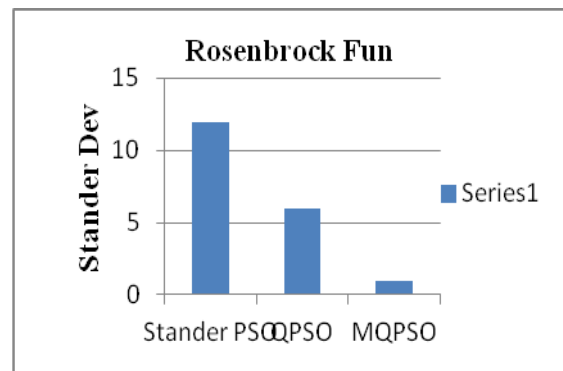| Rosenbrock Fun	Stander PSO	QPSO	MQPSO
Best Error	94.1276	5.95E+01	1.25E+00
Stander Div	11.96125	5.96E+00	9.29E-01

Fig 5 Mean Best Error Rosenbrock Function

Fig 6 Stander Deviation for Rosenbrock Function

Rastrigin function

It is a multimodal function with global minimum located at x = (0,0,....,0) with f(x) = 0 as shown in table 1. It is clear that the MQPSO algorithm is superior to the SPSO and QPSO algorithms as shown in table 4, figures 7 and 8 . The minimum best point obtained as a near global minimum found by the proposed MQPSO algorithm is given in the fourth column in table 6 , which is the coordinates of the best point in the thirty dimensional search space. The best error, stander deviation values, of such a point is given in the last three rows of appendix A as 3.97991 and 0.1451605 respectively.

Table 4 Rastrigin Function Results

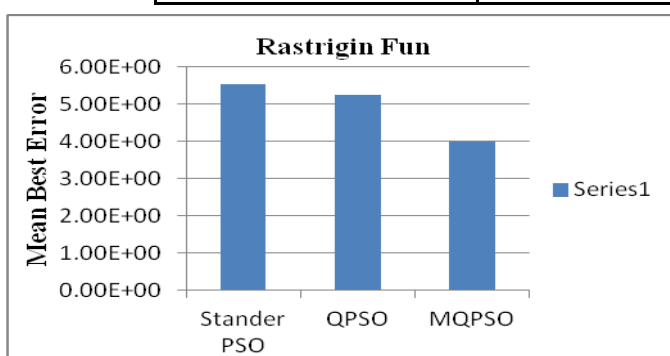| Rastrigin Fun	Stander PSO	QPSO	MQPSO
Best Error	5.54E+00	5.25E+00	3.98
Stander Div	8.60E+00	6.85E+00	6.08

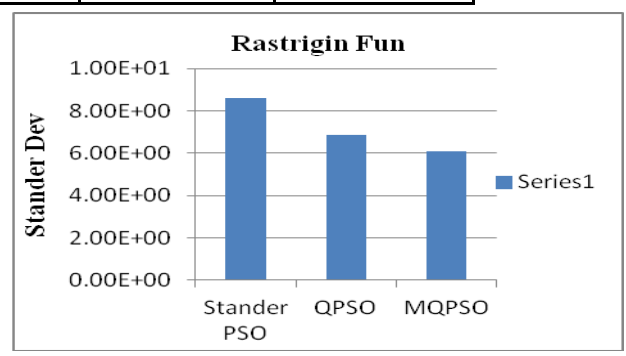**Fig 7 Mean Best Error Rastrigin Function**

Fig 8 Stander Deviation for Rastrigin Function

Schwefel function

It is a multimodal function with global minimum located at x = (420.9687, 420.9687,…., 420.9687) with f(x) = 0 as shown in table 1. It is clear that the MQPSO algorithm is superior to the PSO and QPSO algorithms as shown in table 5 , figures 9 and 10. The minimum best point obtained as a near global minimum found by the proposed MQPSO algorithm is given in the fifth column in table 6 , which is the coordinates of the best point in the two dimensional search space. The best error, stander deviation values , of such a point is given in the last row of table 6 as: **2.5455E-05** and 4.75E-02 respectively.

Table 5 Schwefel Function Results

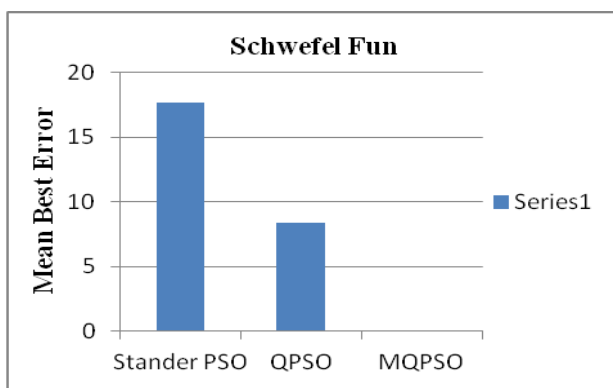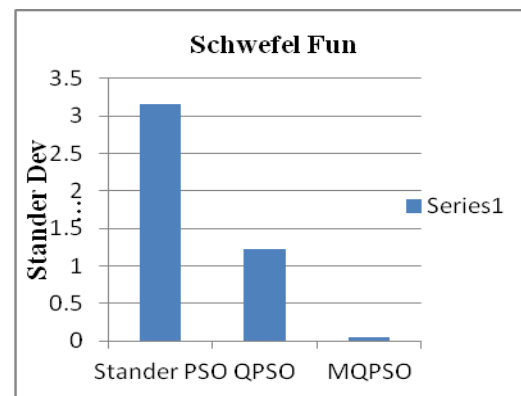| Schwefel Fun	Stander PSO	QPSO	MQPSO
Best Error	17.65	8.42009	2.55E-05
Stander Div	3.164	1.21867	4.75E-02

Fig 9 Mean Best Error Schwefel Function

Fig 10 Stander Deviation **for Schwefel Function**

Table 6 the Optimal Solution Positions found by MQPSO in ten dimensions

Dimension	Sphere	Rosenbrock	Schwefel	Rastrigin	Ackley	Griewank
1	-2E-06	0.999275	420.968746	-2E-05	0	-2.7E-05
2	-1E-06	0.999537	420.968746	-0.0004	1E-06	-2.7E-05
3	-4E-06	1.0033	420.968746	5.6E-05	0	-2.5E-05
4	0	1.0141	420.968746	0.00013	0	0.000099
5	5E-06	1.0397	420.968746	5.8E-05	0	0.000036
6	4E-06	1.0886	420.968746	-4E-05	-1E-06	-3E-06
7	4E-06	1.1872	420.968746	0.99491	-1E-06	0.00005
8	-2E-06	1.41463	420.968746	-0.9947	0	-0.00011
9	2E-06	2.0002	420.968746	0.99498	-1E-06	0.000095
10	-2E-06	4.0126	420.968746	-0.9954	0	-4.3E-05
Best Error	8.74741E-11	1.25060	2.5455E-05	3.97991	2.37145E-06	9.6698E-12

It is clear from the above comparison results that the proposed MQPSO algorithm exhibits a very promising and competitive behavior which reflects the efficacy of its adopted philosophy.

5. Conclusion

In this paper, a new approach, called "Beta Damping Quantum Behaved Particle Swarm Optimization", M-QPSO, tested with a set of two benchmark functions nonlinear functions the statistical results are compared with those obtained through QPSO methods. To improve the performance of QPSO, an adaptive mechanism is introduce for the parameter β of QPSO numerical simulations based on both continuous and discrete chaotic systems demonstrate the effectiveness and feasibility of the modified QPSO method. The comparisons indicate that M-QPSO approach is efficient in reaching near to global optimum and escape from the local minimum. We note that the approach method is faster and more accurate than stander QPSO in most of the problems. Statistical measures confirm the efficacy and the superiority of the proposed method. We can explain the advantages of the program 2 on the program 1 in the following points

- The Modified QPSO approach emulates in some part the natural phenomena.
- The Modified QPSO approach is able to solve problems with either continuous or discrete variables and constrained problems and unconstrained ones.
- The Modified QPSO approach is efficient in reaching near to global optimum and escape from the local minimum.
- The approach method is faster and more accurate than stander PSO and QPSO in most of the problems.
- Statistical measures confirm the efficacy and the superiority of the proposed method.

For the future work, we should give more consideration about how to determine the parameters; try to find other mechanisms to improve the QPSO to solve more engineering optimization problems.

References

[1] Y.Shi,and R.Eberhart." A modified particle swarm optimizer " in the 1998 IEEE, pp. 69-73,1998

[2] J. Sun, B. Feng, and W. Xu. "Particle swarm optimization with particles having quantum behavior," in IEEE Congress on Evolutionary Computation, pp. 325-31, 2004.

[3] J. Liu, J. Sun, W. Xu, and X. Kong. "Quantum-behaved particle swarm optimization based on immune memory and vaccination," in IEEE International Conference on Granular Computing, pp. 453-6, 2006.Wang, Y.H. Jin, F. Tang, and D.X. Huang, 2005, pp. 1261-1271.

[4] J. Sun, W. Xu, and B. Feng, "Adaptive parameter control for quantum-behaved particle swarm optimization on individual level," in 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005. pp. 3049-54.

[5] Y. Chi, X. Liu, K. Xia, and C. Su. "An Intelligent diagnosis to type 2 diabetes based on QPSO algorithm and WLS-SVM," in Intelligent Information Technology Application Workshops, pp. 117-21, 2008.

[6] M. Clerc, and J. Kennedy. "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," IEEE Transactions on Evolutionary Computation, vol. 6(1), pp. 58-73, 2002.

[7] Kennedy, and R.C Eberhart, 1995, "Particle swarm optimization," in , Proceedings of IEEE International Conference on Neural Networks pp.1942-8.

[8] M. Clerc. "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1957, 1999.

[9] W. Fang, and Z. Chai. "Quantum-behaved particle swarm optimization with binary encoding," Adaptive and Natural Computing Algorithms, pp. 376-85, 2007.

[10] W. Fang, J. Sun, and W. Xu,. "Particle swarm optimization with particles having quantum behavior: analysis of the individual particle and parameter selection," Evolutionary Computation, 2010.