Journal of Risk Analysis and Crisis Response, Vol. 5, No. 1 (April 2015), 47-53

# Quantum-behaved Particle Swarm Optimization with Nelder-Mead Simplex Search Method

**Weiquan Yao**

*Department of Basic Course, Guangzhou City Polytechnic, 248 Guangyuan Zhong Road*
*Guangzhou, 510405, China*
*E-mail: yaowq6916@126.com*

Received 6 December 2014

Accepted 8 February 2015

**Abstract**

This paper proposes a novel hybrid algorithm based on quantum-behaved particle swarm optimization (QPSO) algorithm and Nelder-Mead (NM) simplex search method for continuous optimization problems, abbreviated as QPSO-NM. This hybrid algorithm is very easy to be implemented since it does not require continuity and differentiability of objective functions, and it also combines powerful global search ability of QPSO with precise local search of NM simplex method. In a suite of the first 10 test functions taken from CEC2005, QPSO-NM algorithm is compared with other four popular competitors and six special algorithms that are dedicated to solve CEC2005 test function suite. It is showed by the computational results that QPSO-NM outperforms other algorithms in terms of both convergence rate and solution accuracy. The proposed algorithm is extremely effective and efficient at locating optimal solutions for continues optimization.

*Keywords*: *Swarm optimization,Nelder -Mead simplex method,hybrid algorithm, continuous optimization*

## 1. Introduction

Particle Swarm Optimization (PSO) is a swarm intelligence based algorithm which stems from the study of artificial lives and foraging behaviors of biological populations such as bird and fish schools [1,2]. In PSO, each particle represents a potential solution to the optimization problem, and flies around the search space with a certain speed which is continually updated according to flight experience of particle itself and that of others. It is also apparent, however, that PSO may easily be stuck in a local optimum [2]. To improve it, Sun et al. proposed a new PSO model in 2004, which was based on DELTA potential well and believed that particles had quantum behaviors. From this new model, a quantum-behaved particle swarm optimization (QPSO) algorithm was proposed [3]. In quantum spaces, particles are able to search in the whole feasible solution spaces. Therefore, the global search ability of QPSO may be much better than that of the standard PSO. Unlike PSO, QPSO has no velocity vector and fewer parameters to adjust, and it is much easier for implementations. But, QPSO still has the probability of being trapped in the local optimum, and it is not good at exploitation [4]. Nelder-Mead simplex method [5] is a simple direct search technique and it does not require any gradient information, so it was widely used in solving unconstrained function optimization [6]. This paper tries to integrate NM simplex method into QPSO, and proposes hybrid QPSO-NM algorithm so as to improve the local search ability of QPSO.

The rest of this paper is organized as follows: Section 2 gives some preliminaries on QPSO and NM simplex method. Section 3 describes the flowchart of the hybrid algorithm and Section 4 presents experimental results and analysis. Finally, major results of this paper are summarized in Section 5.

Published by Atlantis Press

## 2. Preliminaries

### 2.1. *QPSO Algorithm*

QPSO is a variant of PSO which uses wave function $\Psi(x,t)$ to describe the status of particles. By solving Schrodinger equation, the probability density function of particles appearing in a certain point in the space could be obtained. Then use Monte Carlo stochastic simulation, we may get position equation of particle *i* at *t*+1 th iteration, which is formulated as follows [2,3].

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \frac{1}{2} L_{i,j}(t) \ln[\frac{1}{u_{i,j}(t+1)}], \quad (1)$$

where

$$p_{i,j}(t) = \phi_j(t) \cdot P_{i,j}(t) + [1-\phi_j(t)] \cdot G_j(t) \qquad \phi_j(t) : U(0,1) \quad (2)$$

$p_i(t)$ is called local attractor point of particle *i*; $P_{i,j}(t)$ and $G_j(t)$ are *jth* dimension of *pbest* position of particle *i* and *gbest* position of the whole swarm respectively. $u_{i,j}(t+1)$ is a uniformly distributed random number over (0,1). $L_{i,j}(t)$ is calculated by the following formula.

$$L_{i,j}(t) = 2\beta \,|\, C_j(t) - X_{i,j}(t) \,|, \quad (3)$$

where $C(t)$ is the *mbest* position with coordinates being the following.

$$C(t) = (C_1(t), C_2(t), \cdots C_D(t)) = \frac{1}{M} \sum_{i=1}^{M} P_i(t)$$

$$= (\frac{1}{M} \sum_{i=1}^{M} P_{i,1}(t), \frac{1}{M} \sum_{i=1}^{M} P_{i,2}(t), \quad (4)$$

$$\cdots, \frac{1}{M} \sum_{i=1}^{M} P_{i,D}(t)).$$

Thus, the position updating equation of particle *i* in QPSO is

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \beta \cdot |\, C_j(t) - X_{i,j}(t) \,| \cdot \ln[\frac{1}{u_{i,j}(t+1)}]. \quad (5)$$

In general, we call PSO using equation (5) to update particles' positions a quantum-behaved particle swarm optimization (QPSO) algorithm. $\beta$ in (5) is known as contraction – expansion (CE) coefficient, the only parameter to be tuned by the algorithm, which is used to control convergence rate. In [8], it is suggested that the value of $\beta$ could linearly decrease from $\beta_1$ to $\beta_0$ during the search process, and that this would be a simple but effective way. So, computational formula of $\beta$ is

$$\beta = \beta_0 + (T-t) \cdot (\beta_1 - \beta_0) / T, \quad (6)$$

where $\beta_1$ and $\beta_0$ are initial and final values of $\beta$ respectively. Often we set $\beta_1 = 1, \beta_0 = 0.5$.

### 2.2. *Nelder-Mead simplex method*

Nelder and Mead proposed a simplex search method, which is a local search algorithm designed for unconstrained optimization without using any gradient information [7]. For a function minimizing problem with D variables, NM method continuously updates "simplex" through iterations after comparing objective values at D+1 vertices followed by replacing the worst vertex with a new one that is generated by using four basic procedures: reflection, expansion, contraction and shrinkage. Through these procedures, the simplex can successfully improve itself and get closer to the optimum.

## 3. QPSO-NM algorithm

The local NM search method is integrated into the main QPSO, and critical steps of QPSO-NM algorithm are illustrated below:

(1) Initialization of NM "simplex". Since NM method uses D+1 vertices for D dimensional optimization problems, D randomly generated positions in the search range, together with the gbest position are used to form the initial NM "simplex".

(2) A parameter named local search period (L) is used to control the running of NM search method, that is to say, if the current number of iterations is a multiple of L, then the main algorithm will turn to the local search.

(3) Max number of function evaluations for NM simplex method (L_MaxFEs) is used to determine the running time of the local search.

Suppose $f$ is a function to be minimized, the pseudocode of QPSO-NM may be depicted as follows.

Initialize position vectors of each particle and evaluate them

Determine *pbest* and *gbest* positions

Initializing NM "simplex"

```
gen = 1
While terminal condition is not met
    Compute mbest position C using (3)
    Compute the value of β using (5)
    For i=1:M
        For j = 1:D
            φ = rand(0,1)
            p_{i,j} = φ * P_{i,j} + (1-φ)*G_j
            u = rand(0,1)
            If rand(0,1) < 0.5
            X_{i,j} = p_{i,j} + β *| C_j-X_{i,j}|*ln(1/u)
            Else
                X_{i,j} = p_{i,j} - β *| C_j-
            X_{i,j}|*ln(1/u)
            End
        End
        Evaluated the new position
        Update pbest of the i th positon
    End
    Update gbest postion
    If mod (gen,L) ==0
        Conduct a local search using NM simplex
        search method and update FEs
        correspondingly
        Find the best position after NM simplex search,
        denoted as NMbest。 And use it to update
        gbest accordingly.
    End
    gen = gen + 1
End
```

## 4. Computational results and analysis

Now we use QPSO-NM algorithm to solve the first 10 test functions from CEC2005 (see Table 1). The experiment is conducted on the 10-D problems with the terminal criterion Max_FEs setting to 1E+5. The value of parameter L is set to 50, and that of L_MaxFEs is set to 500. To conduct fair comparisons between our hybrid QPSO-NM and other algorithms, the number of max function evaluations (Max_FEs) is used as a time measure, i.e. all algorithms will be terminated once FEs reaches to Max_FEs. And the best function error value is employed as evaluation criterion for each algorithm. The convergence curves of SPSO [9], CLPSO [10], QPSO [3], DMS-PSO [11] and QPSO-NM on each test function are plotted in Figure 1 . From Figure 1, it could be seen that convergence curves of QPSO-NM (the red solid lines) can approach the true optimal value faster than the rest algorithms on functions F1, F2, F3, F4, F6, F8 and F10. While for functions F5 and F9, DMS-PSO is better than QPSO-NM at first phase of the evolution, but the difference becomes very tiny at last. For function F7, the convergence speed of QPSO-NM is slower than SPSO at first, but eventually the former outperforms the latter in terms of the quality of solutions. It is shown by the final tendency of convergence curves that the red solid lines always locates in the lowest place, which means a higher solution accuracy of QPSO-NM than other competitors. All in all, QPSO-NM is superior to other four algorithms in terms of both convergence speed and solution accuracy.

Table 1. The first 10 test functions from CEC2005

| Func No. | Function name | Bounds | Minimum |
|---|---|---|---|
| F1 | Shifted Sphere Function | [-100,100] | -450 |
| F2 | Shifted Schwefel's Problem 1.2 | [-100,100] | -450 |
| F3 | Shifted Rotated High Conditioned Elliptic Function | [-100,100] | -450 |
| F4 | Shifted Schwefel's Problem 1.2 with Noise in Fitness | [-100,100] | -450 |
| F5 | Schwefel's Problem 2.6 with Global Optimum on Bounds | [-100,100] | -310 |
| F6 | Shifted Rosenbrock's Function | [-100,100] | 390 |
| F7 | Shifted Rotated Griewank's Function without Bounds | Initial bounds [0, 600] | -180 |
| F8 | Shifted Rotated Ackley's Function with Global Optimum on Bounds | [-32,32] | -140 |
| F9 | Shifted Rastrigin's Function | [-5,5] | -330 |
| F10 | Shifted Rotated Rastrigin's Function | [-5,5] | -330 |

The source code of these test functions is available on P.N.Suganthan's website:http://www.ntu.edu.sg/home/EPNSugan/

For each test function, the QPSO-NM is run 25 times. The minimum (MIN), maximum (MAX), mean value (MEAN) and the standard deviation (STDEV) of the best function error values are presented in Table 2. From Table 2, it is found that the mean error values are zero when solving functions F1, F2, F4 and F9, indicating that QPSO-NM is very effective in optimizing the above functions. For the purpose of comparison, the mean of the best function error values of QPSO-NM together with SPSO,CLPSO,QPSO,

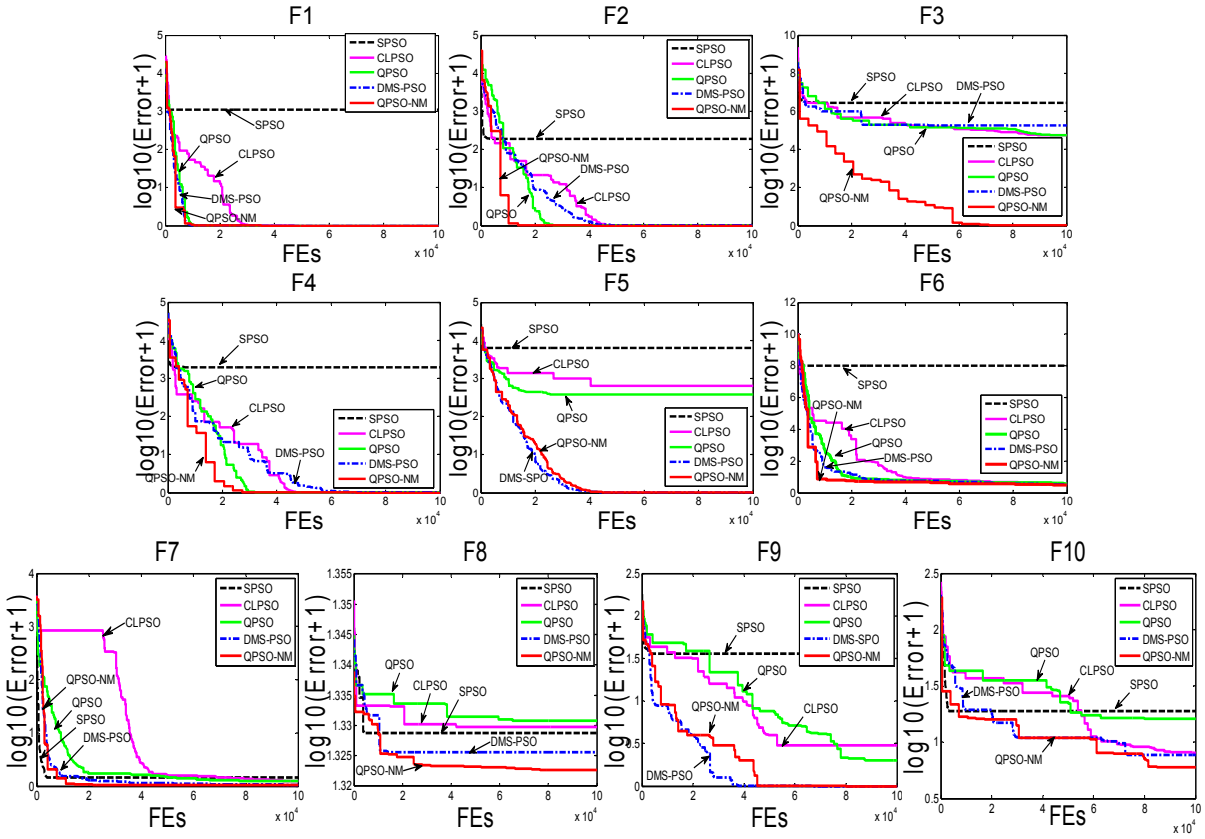


Fig. 1. Convergence curves of SPSO, CLPSO, QPSO, DMS-PSO and QPSO-NM on each test function. The abscissa is FEs, and the ordinate is log10(Error+ 1), where Error is the best function error value.

DMS-PSO, FEA[12], Hybrid RCGA[13], PSO Variant[14],DE[15]， ES [16] and RCMA[17], are listed in Table 3 and Table 4, where the mean error values of FEA, Hybrid RCGA,PSO Variant, DE, ES and RCMA are directly taken from corresponding literatures.

Table 2. The MIN, MAX, MEAN and STDEV of the best function error values of QPSO-NM

| | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| MIN | 0.00000E+00 | 0.00000E+00 | 1.59730E-11 | 0.00000E+00 | 1.36993E-10 |
| MAX | 0.00000E+00 | 0.00000E+00 | 2.43565E-06 | 0.00000E+00 | 6.04978E-07 |
| MEAN | 0.00000E+00 | 0.00000E+00 | 1.00389E-07 | 0.00000E+00 | 4.38655E-08 |
| STDEV | 0.00000E+00 | 0.00000E+00 | 4.86528E-07 | 0.00000E+00 | 1.21641E-07 |
| | F6 | F7 | F8 | F9 | F10 |
| MIN | 0.00000E+00 | 9.85728E-03 | 2.00002E+01 | 0.00000E+00 | 9.94959E-01 |
| MAX | 2.76780E+00 | 1.08312E-01 | 2.01894E+01 | 0.00000E+00 | 8.95463E+00 |
| MEAN | 1.32368E+00 | 5.94651E-02 | 2.00561E+01 | 0.00000E+00 | 5.15519E+00 |
| STDEV | 7.45577E-01 | 2.84646E-02 | 4.86346E-02 | 0.00000E+00 | 1.82574E+00 |

Table 3. The comparison of the mean best function error values for functions F1-F5

| | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|
| SPSO | 1.05387E+03 | 6.35766E+02 | 1.58418E+06 | 1.22268E+03 | 1.91479E+03 |
| CLPSO | 0.00000E+00 | 3.68345E-13 | 6.94785E+04 | 4.09273E-13 | 5.21414E+02 |
| QPSO | 0.00000E+00 | 1.02318E-12 | 1.47868E+05 | 1.87933E-09 | 3.24642E+02 |
| DMS-PSO | 0.00000E+00 | 2.69515E-07 | 1.37347E+05 | 2.24495E-04 | 9.37320E-09 |
| FEA | 7.89310E-09 | 9.15620E-01 | 6.45730E+05 | 1.34520E+02 | 3.51070E+02 |
| Hybrid RCGA | 8.34460E-09 | 8.20750E-09 | 5.70530E+02 | 8.31770E-09 | 8.93750E-09 |
| PSO Variant | 0.00000E+00 | 0.00000E+00 | 2.29034E+05 | 0.00000E+00 | 0.00000E+00 |
| DE | 0.00000E+00 | 5.70000E-02 | 2.09000E+05 | 6.19000E-01 | 1.21796E+02 |
| ES | 8.16020E-09 | 2.90000E-06 | 3.52170E+05 | 4.13570E+03 | 1.36820E+03 |
| RCMA | 9.86950E-09 | 9.93600E-09 | 4.77086E+04 | 1.99669E-08 | 2.12417E-02 |
| QPSO-NM | 0.00000E+00 | 0.00000E+00 | 1.00389E-07 | 0.00000E+00 | 4.38655E-08 |

The five functions in Table 3 are unimodal functions. F1, F2 and F3 have different condition numbers which make F3 to be harder than F2 and F2 to be harder than F1. For F1, we could observe, from the results , that the mean best function error values of QPSO-NM, CLPSO,QPSO,DMS-PSO,PSO Variant and DE are all equal to zero, which indicates that this test function could be effectively optimized by these algorithm. For F2 and F4, QPSO-NM and PSO Variant achieve a better result. F3 is a very hard function to be optimized, and QPSO-NM obtains a significant better result than other algorithms of which the order of magnitude reaches to 10-7 clearly lower than its counterparts. F5 is Schwefel's problem 2.6 with global optimum on bounds, for this function the performance of QPSO-NM is worse than that of PSO Variant, Hybrid RCGA and DMS-PSO. Based on the analysis above, we can conclude that QPSO-NM and DMS-PSO have better overall performances for functions F1-F5. Furthermore, QPSO-NM evidently surpasses PSO Variant when solving F3. Therefore, QPSO-NM is deemed to be the most competitive algorithm among these eleven algorithms.

Table 4. The comparison of the mean best function error values for functions F6-F10

| | F6 | F7 | F8 | F9 | F10 |
|---|---|---|---|---|---|
| SPSO | 9.27075E+07 | 3.19381E-01 | 2.03047E+01 | 1.75415E+01 | 3.15131E+01 |
| CLPSO | 2.40375E+00 | 1.41874E-01 | 2.03577E+01 | 9.15362E-01 | 6.68042E+00 |
| QPSO | 1.94224E+01 | 2.79719E-01 | 2.03674E+01 | 1.08812E+00 | 9.67376E+00 |
| DMS-PSO | 4.21388E+00 | 7.05375E-02 | 2.01527E+01 | 0.00000E+00 | 5.51815E+00 |
| FEA | 3.21610E+03 | 1.83840E+00 | 2.01430E+01 | 7.91790E-09 | 2.34010E+01 |
| Hybrid RCGA | 8.87370E-09 | 1.17230E-02 | 2.03540E+01 | 1.15420E+00 | 4.97480E+00 |
| PSO Variant | 1.55292E+01 | 1.71838E-01 | 2.03034E+01 | 2.78588E+00 | 1.85821E+01 |
| DE | 3.53800E+00 | 1.57751E+02 | 2.04440E+01 | 4.54000E-01 | 5.11460E+01 |
| ES | 7.49040E+01 | 1.18260E+00 | 2.03680E+01 | 4.48520E+01 | 1.03020E+02 |
| RCMA | 1.48962E+00 | 1.97139E-01 | 2.01923E+01 | 4.37855E-01 | 5.64289E+00 |
| QPSO-NM | 1.32368E+00 | 5.94651E-02 | 2.00561E+01 | 0.00000E+00 | 5.15519E+00 |

Functions F5-F10 are multimodal functions. QPSO-NM achieves the best result for F8, and it (along with DMS-PSO) also performs the best when optimizing F9. For F6, F7 and F10, Hybrid RCGA is the best one followed by QPSO-NM: the differences between the above two algorithms are small for F7 and F10, while Hybrid RCGA obviously outperforms QPSO-NM for F6. We may carefully come to the conclusion that QPSO-NM and Hybrid RCGA can achieve better effects than other algorithms in solving functions F6-F10.

For each function, rank all algorithms according to their mean function error values from small to large, and calculate total and final ranks for each algorithm shown in Table 5. It can be clearly seen from Table 5 that the top four algorithms are QPSO-NM, DMS-PSO, Hybrid RCGA and PSO Variant, respectively. Furthermore, it is also found from Table 3 and Table 4 that: PSO Variant is very effective on unimodal functions F1-F5 but fails to solve multimodal functions F6-F10; Hybrid RCGA, however, can nicely minimize multimodal functions F6-F10 but is inferior to QPSO-NM on unimodal functions F1-F5. This phenomenon may be explained by the "no free lunch" theorem [18], and it says "any elevated performance over one class of problems is offset by performance over another class," indicating that there is no algorithms perform the best over any class of problems, i.e. each algorithm has its corresponding application range. However, QPSO-NM has an overall better performance on both unimodal and multimodal functions compared with PSO Variant and Hybrid RCGA, which could be explained by a finding in [19] stating "the hybrid of algorithms may be an effective measure to expand algorithms' application range and improve their performances." Thus, QPSO-NM is the most competitive algorithm in solving the first 10 test functions from CEC2005 with an expended application range achieved by a successful hybrid of QPSO and NM simplex method.

Table 5. Ranks of all algorithms

| | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | Total ranks | Final ranks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPSO | 11 | 11 | 11 | 10 | 11 | 11 | 8 | 6 | 10 | 9 | 98 | 11 |
| CLPSO | 1 | 3 | 4 | 3 | 9 | 4 | 4 | 8 | 6 | 5 | 47 | 5 |
| QPSO | 1 | 4 | 6 | 4 | 7 | 8 | 7 | 9 | 7 | 6 | 59 | 7 |
| DMS-PSO | 1 | 7 | 5 | 7 | 3 | 6 | 3 | 3 | 1 | 3 | 39 | 2 |
| FEA | 7 | 10 | 10 | 9 | 8 | 10 | 10 | 2 | 3 | 8 | 77 | 9 |
| Hybrid RCGA | 9 | 5 | 2 | 5 | 2 | 1 | 1 | 7 | 8 | 1 | 41 | 3 |
| PSO Variant | 1 | 1 | 8 | 1 | 1 | 7 | 5 | 5 | 9 | 7 | 45 | 4 |
| DE | 1 | 9 | 7 | 8 | 6 | 5 | 11 | 11 | 5 | 10 | 73 | 8 |
| ES | 8 | 8 | 9 | 11 | 10 | 9 | 9 | 10 | 11 | 11 | 96 | 10 |
| RCMA | 10 | 6 | 3 | 6 | 5 | 3 | 6 | 4 | 4 | 4 | 51 | 6 |
| QPSO-NM | 1 | 1 | 1 | 1 | 4 | 2 | 2 | 1 | 1 | 2 | 16 | 1 |

## 5. Based on the PSO of enterprise credit risk assessment model of BP neural network research

(1)B P neural network parameter Settings. For the number of input layer nodes, because the network is the enterprise credit risk assessment index of the input sample data, therefore, the number of input layer nodes by PCA transformation after the extraction of characteristic data, the number of decisions. Determine the number of input layer node model for 14. For the number of output layer nodes, as a result of enterprise credit risk assessment for yes or no only two kind of criterion, as a result, the output layer node number 1.

For the determination of the number of hidden layer nodes, this model adopts the method of variable number in hidden layer. Namely the start into the more hidden layer nodes, according to the empirical formula to determine the hidden node number is 6. As you progress through the training process, and gradually trim off doesn't work node in the network training. For each neuron, if in the process of training, the connection power is less than the weight of the absolute value of the numerical convergence factor, and then assign the weights of zero. If all of the neurons connection weights are zero, this god by nodes are removed. This paper using B P algorithm for neural network training, factor weight to 0. 001, mistakenly sent to receive the folding for child is 0. 00001. For the network weights, we using PSO algorithm to optimize .The weight is decided by the parameters of the particles of the network. [20]

(2) the design of fitness function, the size of the fitness available under type said:

$$E = \sum_{r} \sum_{j} \frac{\left(Y_{rj} - D_{rj}\right)^2}{2}$$

The Yrj is the expected output node, Drj is a node of the actual output.

(3) the PSO parameters Settings

Particle dimension: is determined by the network connection weights, namely by the input node output nodes and the number of hidden layer nodes. Suppose I, H, O, respectively of input layer, hidden layer and output layer neuron number, the dimension of each particle size D can be expressed as: D = (I3H) + 3 O (H) + H + O. Therefore, this paper models the dimensions of the particle size for D = 109.

the number of particles, on the basis of experience for general problem, particle number take 10 can obtain good effect. In order to achieve better training effect, in this paper, the particle number 15.

Vmax: maximum speed, decided to particles in a loop of the largest mobile distance, usually set the width of the range of particle, this paper set the particle belongs to [25, 5], then the size of theVmax is 10.

Learning factor: cl and c2 is usually equal to 2, but there are other values in the literature, but its range between 0 and 4. This article take the c l = c2 = 2.

Stop condition: maximum cycle number and the minimum error. In this paper, the minimum error in the model set for a classification error, the largest circulation set for 2000.

(4) initialization: randomly generated M a particle, the particle's position and velocity range control between the (0, 1), the dimensions of the particle's position and velocity vector is equal to D. Sets the pbest of each particle to the initial position, the best value for the gbest pbest.

(5) the fitness evaluation: the population of each individual fitness evaluation. If is better than that of p best pbest is when in position. If all particles is superior in gbe st, gbe st is replaced.

(6) the position and velocity update: update position and velocity of each particle.

(7) Algorithm termination: if meet the precision requirement or the whole evolution has reached the maximum number of iterations (to 2000), an algorithm and record the current best individual in the whole group. Otherwise, go to step (5).

In Matlab 2012 implemented in the model proposed in this paper. Based on credit risk evaluation index system, we adopt 15 projects related data as sample, 12 of them as the training sample, 3) as a test sample, the data after normalization, to analyze the main points and processing, as the input data into a network, the network training and testing respectively. As you can see, in this paper, the constructed model prediction results and the actual value of the original data error is very small

## 6. Conclusions

This paper studies the hybrid algorithm that integrates QPSO and Nelder -Mead (NM) simplex search method. QPSO has powerful global search abilities, but its local search abilities are not so good. Nelder-Mead simplex search method is a popular and simple direct search technique for unconstrained function optimization without using gradient information. In order to combine

powerful global search ability of QPSO with precise local search of NM simplex method, this paper studies the hybrid of the above two algorithms and proposes QPSO-NM. In a suite of the first 10 test functions taken from CEC2005, QPSO-NM algorithm is compared with other four popular competitors and six special algorithms that are dedicated to solve CEC2005 test function suite. It is showed by the computational results that QPSO-NM outperforms other algorithms in terms of both convergence rate and solution accuracy. The experimental results show that the proposed algorithm is extremely effective and efficient at locating optimal solutions for continues optimization.

## 7. References


[1] Kennedy J， Eberhart R C . Particle Swarm optimization[C] . Proceedings of IEEE International Conference on Neural Networks,1999:1942-1948

[2] J. Sun, Research on quantum-behaved particle swarm optimization algorithm, doctoral dissertation, Jiangnan University,2009.

[3] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: IEEE Congress on Evolutionary Computation, 2004, pp.325–331.

[4] J. Sun, W. Fang, V. Palade, X.J. Wu, W.B. Xu, Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point, Applied Mathematics and Computation 218 (2011) 3763–3775.

[5] J. A. Nelder and R. Mead, A simplex method for function minimization, The Computer Journal, 7 (1965) 308-313.

[6] D. H. Chen, Z. Saleem, and D. W. Grace, A new simplex procedure for function minimization,International Journal of Modeling and Simulation, 6 (1986) 81-85

[7] J.A. Nelder, R. Mead, A simplex method for function minimization, Computer Journal, 7 (1965) 308–313.

[8] J. Sun, W. Fang, X.J. Wu, V. Palade, W.B. Xu, Quantum-behaved particle swarm optimization: analysis of the individual particle’s behavior and parameter selection, Evolutionary Computation, doi:10.1162/EVCO_a_00049.

[9] Y. Shi, R. Eberhart, A Modified Particle Swarm Optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, NJ: IEEE Press, 1998, pp. 69-73.

[10] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions, IEEE Congress on Evolutionary Computation,2006,vol.10, 281-295.

[11] J. J. Liang and P. N. Suganthan, Dynamic Multi-Swarm Particle Swarm Optimizer, in: Proceedings of 2005 IEEE Swarm Intelligence Symposium, 2005, pp. 124-129.

[12] S. Alonso, J. Jimenez, H. Carmona B. Galvan, G. Winter, Performance of a Flexible Evolutionary algorithm, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[13] C. García-Martínez, M. Lozano, Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[14] M. F. Tasgetiren, Y.C. Liang, G. Gencyilmaz, I. Eker, Global Optimization of Continuous Functions Using Particle Swarm Optimization, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[15] L. T. Bui, Y. Shan, F. Qi, and H. A. Abbass, Comparing Two Versions of Differential Evolution in Real Parameter Optimization, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[16] L. Costa, A Parameter-less Evolution Strategy for Global Optimization, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[17] D. Molina, F. Herrera, M. Lozano, Adaptive Local Search Parameters for Real-Coded Memetic Algorithms, in: Special Session on Real-Parameter Optimization at CEC-05, Edinburgh, UK, 2005.

[18] D. H. Wolpert , W. G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation ,1997, vol. 1, pp. 67–82.

[19] L. Wang, Intelligent optimization algorithm with applications, Beijing: Tsinghua University press, 2001,119-121.

[20] Shi Y, Ebe rha rt R.A modified paretic leeward optimizer [ C ] / / I EEE World Congress on Compu tational Intelligence, 1998: 69273 .