

A Fractal Evolutionary Particle Swarm Optimizer

Xiaohong Qiu

Software School, Jiangxi University of Science and Technology, Nanchang, China
Email: jxauqiu@163.com

Xiaohui Qiu

Institute of Higher Education, Jiangxi Agricultural University, Nanchang, China
Email: qxh@mail.jxau.edu.cn

Fang Liao

Temasek Laboratories, National University of Singapore, Singapore
Email: tsllf@nus.edu.sg

Abstract—A Fractal Evolutionary Particle Swarm Optimization (FEPSO) is proposed based on the classical particle swarm optimization (PSO) algorithm. FEPSO applies the fractal Brownian motion model used to describe the irregular movement characteristics to simulate the optimization process varying in unknown mode, and include the implied trends to go to the global optimum. This will help the individual to escape from searching optimum too randomly and precociously. Compared with the classical PSO algorithm, each particle contains a fractal evolutionary phase in FEPSO. In this phase, each particle simulates a fractal Brownian motion with an estimated Hurst parameter to search the optimal solution in each sub dimensional space, and update correspond sub location. The simulation experiments show that this algorithm has a robust global search ability for most standard composite test functions. Its optimization ability performs much better than most recently proposed improved algorithm based on PSO.

Index Terms—fractal, evolutionary algorithm, particle swarm optimization, fractal Brownian motion, optimization, global

I. INTRODUCTION

Particle Swarm Optimization (PSO) simulated birds foraging behavior is conceptually simple, convenient, efficient search algorithm with few parameter settings[1]. Since 1995 It has been studied all the time. It is easy to fall into local optimum by using PSO algorithm to optimize complex functions and get a premature convergence result. Therefore the inertia weight for the speed of the PSO evolution equation is introduced [2] and is dynamically adjusted to balance the global convergence feature and the convergence speed. Combined with fuzzy theory, fuzzy dynamic adjustment of the inertia weight method is proposed [3]. In order to ensure the convergence of PSO, a shrinkage factor is

included in the algorithm [4-5]. Based on the selection, hybridization and other ideas of the comprehensive evolutionary computation, the particle with worst fitness is replaced with the copier of the particle with better fitness to improve the algorithm convergence [6-7]. To improve PSO performance, neighborhood topology is combined with niche ideas [8]. Each particle neighborhood grows with the evolution of the generation from itself to the scope that the entire population is included. The neighborhood models such as Ring and Wheel basic structures have been studied by Kennedy [9].

In past several years, several improved PSO algorithms with good performance have been studied. In the Fitness-Distance-Ratio algorithm based Particle of Swarm Optimization(FDR-PSO) proposed in 2003 [10], each particle moves to multiple particles with better fitness according to the fitness - distance criteria to overcome the PSO premature convergence feature. Cooperative Particle Swarm Optimizer (CPSO) studied in 2004[11] uses the collaborative work sub-particle swarm optimized separately in the different parts of the solution vector to achieve better optimization results. Comprehensive Learning Particle of Swarm Optimizer (CLPSO) proposed in 2006 updates the velocity after comprehensive study of all other particles best location in their histories [12] to achieve global optimum. In 2010 Intelligent Single Particle Optimizer (ISPO) is studied. The particle is no longer a simple individual, but with some intelligent using smart strategic decision to update its location and velocity. The test results have demonstrated that ISPO has an outstanding ability to find the global optimum in the complicated composition test functions[13].

But ISPO has few adaptability for different optimization objective function, and should keep trying different algorithm parameter to obtain better optimal solution. If the particle swarm movement in searching the optimization problem solution is regarded as natural environment movement, the particle swarm optimization process irregular movement is just like the fractal Brownian motion (FBM) used in Fractal Theory [14] to

Manuscript received April 13, 2012; revised May 22, 2012; accepted May 27, 2012.

Project number: 60674054 and 2010GZC0090

Corresponding author: Xiaohui Qiu.

solve a wide variety of applications such as stock market analysis and forecasting applications. Therefore a fractal evolutionary algorithm is proposed to combine PSO with comprehensive utilization of the FBM randomness and some trend which reflected the imbalance of the objective function different value in different location. This is the same as the effect in the evolution equation of the PSO by adding random items to prevent premature convergence [15], and get better optimization results.

II. PSO FRACTAL EVOLUTIONARY ALGORITHM

A. Classical PSO

In a PSO system with n particles based on *inertia weight* [2], particles representing candidate solutions start their flight from random locations in a search landscape in D dimension. At each step, the i th particle updates its velocity $v_i = (v_{i1}, v_{i2}, \dots, v_{id}, \dots, v_{iD}) \in [v_{\min}, v_{\max}]$ to move to another location $x_i = (x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{iD}) \in [x_{\min}, x_{\max}]$ based on (1) and (2). The flight is influenced by a *fitness function* $f(x_i)$ that evaluates the quality of each solution.

$$V_{id}(t+1) = \omega \times V_{id}(t) + C_1 \times \phi_1 \times (P_{lid} - X_{id}(t)) + C_2 \times \phi_2 \times (P_{gd} - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

Where $X_{id}(t)$ is the location of the i th particle at time t on dimension d , V_{id} is the velocity, ω is the inertia weight, C_1 and C_2 are constant values, $\phi_1, \phi_2 \in [0, 1]$ are uniform random numbers, P_{lid} is the i th particle's best position (generating the best fitness) that has been found so far, and P_{gd} is the best position visited by the neighbors. Generally, the neighborhood is chosen as the whole population for global optimization. To prevent the PSO from premature convergence [15], random update items is added in (1) shown as the following equation (3). Where C_3 is constant value and $\phi_3 \in [-0.5, 0.5]$ is a uniform random factor.

$$V_{id}(t+1) = \omega \times V_{id}(t) + C_1 \times \phi_1 \times (P_{lid} - X_{id}(t)) + C_2 \times \phi_2 \times (P_{gd} - X_{id}(t)) + C_3 \phi_3 \quad (3)$$

B. Fractal evolutionary PSO

In (3), the random item $C_3 \phi_3$ is added with no any guided information. Therefore fractal Brownian motion (FBM) model is proposed to replace this item in this paper. FBM has been studied and effectively used to simulate many natural and social phenomena [14]. It includes two parts: one is the increment of the stochastic process of Brownian motion, the other is the Gaussian noise. Brownian motion shows that particle movement has some trends in the surrounding fluid internal molecular irregular motion. This is similar to the individual optimizing its fitness process in PSO. Thus the FBM search phase is introduced in the classical PSO algorithm. There are two phase integrated with the classical PSO framework shown in Fig. 1: one is the phase of each particle FBM exploration and the other is the phase of the best individual FBM optimization. The new

integrated algorithm is called Fractal Evolutionary PSO (FEPSo).

Supposed $B_H(t)$ stands for the FBM with the Hurst parameter H ($0 < H < 1$), the i th particle movement change $dx_i(t)$ relative to $x_i(t)$ in the dt time contains two parts. One part is used to reflect the overall trend of the objective function, the other part includes the random interference as the FBM. The movement is defined by the following equations according to [14]:

$$\frac{dx_i}{x_i} = \mu dt + \sigma dB_H(t), x_i(0) = x_{i0} > 0 \quad (4)$$

The solution of equation (4) is

$$x_i(t) = x_{i0} e^{\sigma B_H(t) + \mu t - \frac{1}{2} \sigma^2 t^{2H}}, t \geq 0 \quad (5)$$

The expected value of $x(t)$ is

$$E(x_i(t)) = x_{i0} e^{\mu t}, t \geq 0 \quad (6)$$

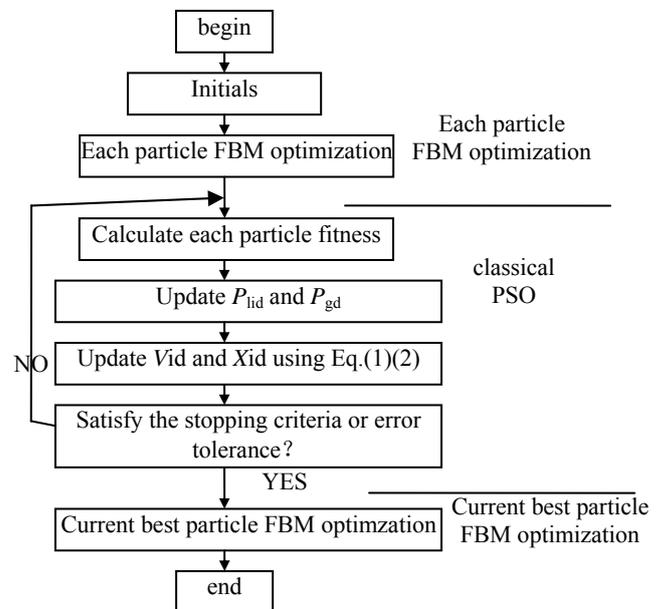


Figure 1. New framework for the FEPSo algorithm

We have a new equation (7) derived from (4),

$$v_i = \frac{dx_i}{dt} = x_i \mu + \sigma \frac{dB_H}{dt}, x_i(0) = x_{i0} > 0 \quad (7)$$

According to [14], $\frac{dB_H}{dt}$ is fractal white noise. To simplify the calculation, the following formula is used to approximately estimate the value in (7):

$$v = \frac{a\sigma}{t+2} e^{\sigma_1 - \sigma_2 t} f_{random}(-\alpha, \alpha) + b \times \mu, \quad \text{where}$$

$f_{random}(-\alpha, \alpha)$ is uniform distribution random function in the interval $[-\alpha, \alpha]$. Obviously in the process of fractal motion, the velocity vector consists of two parts: random incremental part $\frac{a\sigma}{t+2} e^{\sigma_1 - \sigma_2 t} f_{random}(-\alpha, \alpha)$ and reflecting fitness value to optimum trend part $b \times \mu$.

Random incremental part has reflected the fractal Brownian motion is the random motion characteristics on the basis of the increment of the stochastic process of Brownian motion with the addition of Gaussian noise new formation. Global fitness value changes μ can be estimated by the velocity v as a reference value. σ is reflected the value of the trend of fluctuations and is estimated by (8), where $x_{g_{best}}$ is the location with best fitness value and $x_{g_{worst}}$ is the worst fitness value location of the experienced.

Because the fractal parameter estimation is difficult, one-dimensional FBM is used to update particle velocity and location sub-vector $j(1 \leq j \leq m)$ in turn. The following formula used for the velocity and location updated will iterate N times:

$$\sigma = \sqrt{\frac{1}{D} \sum_{i=1}^D (x_{i_{g_{best}}} - x_{i_{g_{worst}}})^2}, \sigma_1 = \sigma f_{random}(0.2, 1.0), \sigma_2 = \sigma f_{random}(0.05, 0.5) \quad (8)$$

$$v_j^k = \frac{a\sigma}{k+2} e^{\sigma_1 - k\sigma_2} f_{random}(-0.5, 0.5) + b \times \mu_j^{k-1}, a=0.382, b=\sigma f_{random}(0, 2.0) \quad (9)$$

$$x_j^k = \begin{cases} x_j^{k-1} + v_j^k & \text{if } f(\mathbf{x}_0^k) > f(\mathbf{x}_0^{k-1}) \\ x_j^{k-1} & \text{if } f(\mathbf{x}_0^k) \leq f(\mathbf{x}_0^{k-1}) \end{cases} \quad (10)$$

$$\mu_j^k = \begin{cases} \mu_j^{k-1} / s & \text{if } \sum abs(\mu_j^{k-1}) > \varepsilon \text{ And } \text{if } f(\mathbf{x}_0^k) \leq f(\mathbf{x}_0^{k-1}) \\ 0.14 / D \times f_{min}(\varepsilon) - 0.236 \ln \mu_j^k & \text{if } \sum abs(\mu_j^{k-1}) \leq \varepsilon \text{ And } f(\mathbf{x}_0^k) \leq f(\mathbf{x}_0^{k-1}) \\ 0.382 \mu_j^{k-1} + 1.382 \mu_j^k & \text{if } f(\mathbf{x}_0^k) > f(\mathbf{x}_0^{k-1}) \end{cases} \quad (11)$$

Where $k = 1, 2, \dots, N$ is the iteration number of searches, stands for time; $\mathbf{x}_0^k = [x_1, x_2, \dots, x_j^{k-1}, \dots, x_m]$ stands for the k -th location vector; $\mathbf{x}_0^k = [x_1, x_2, \dots, x_j^{k-1} + v_j^k, \dots, x_m]$ is the trial new location; parameter μ_j^k for the j th component of μ ; $f(x_0)$ is the algorithm fitness evaluation function. Component v_j^k of the change rate of the location x_j^k is determined by (9). μ_j^k is determined by its fitness value $f(x_0)$, compared with the original value $f(x_0)$ after its initial value is set to zero. If $f(x_0)$ get to a better value, the location is update by the formula (10). μ_j^k is estimated by (11). By (9), $v_j^k = \frac{a\sigma}{k+2} e^{\sigma_1 - k\sigma_2}$

will decay to zero with the iterative parameter k increased rapidly. This means the global search to local search conversion. The second part $b \times \mu$ of the formula (9) will get a large value if the fitness is better each step and get rapidly a small value if the fitness is worse each step. In formula (11) $s = 4.0$ is the attenuation factor, the error threshold is $\varepsilon = 0.0001$. The golden section method is included in (11) to estimate the value μ as soon as possible. Therefore the pseudocode for the FEPSO algorithm is shown as Fig. 2. By (11), a new search decision is used:

(1) In each iteration, if the particle has found a better location than its history, the μ will increase and the

particle velocity of the next iteration will be amplified by (9).

(2) If in this iteration the particle has not found a better location, μ will decrease and the particle velocity of the next iteration will be lower by (9) and will help to search the global optimum in the new iteration.

Begin

Initialize particles, define the population size n , search space dimension D and fitness calculation maximum times N_{max} .

Calculate the fitness of initial particles. $N_{FEs} = n$

Calculate the value σ estimated by Eq.(8).

Set individual FBM optimization times N_{loop} , search one dimension times N , classical PSO maximum times N_{PSOMax} , the current best individual times N_b , satisfied with the equation $N_{max} \geq N_{loop} \times N \times D \times n + N_{PSOMax} + N_b \times N \times D$

Begin for each particle FBM optimization

Step 1 Select i th particle to search in FBM called function FrOptimization(i)

Begin of function FrOptimization(i)

Initial location vector $x^0 = [x_1^0, x_2^0, \dots, x_D^0]$;

Calculate the fitness value $f(x^0)$,

using $f(\mathbf{x}_0^k) = f(x_1^{k-1}, x_2^{k-1}, \dots, x_j^{k-1}, \dots, x_D^{k-1})$

Set function evaluation times $N_{FEs} = N_{FEs} + 1$;

Step 2 Individual FBM search times loop=1

Step 3 Sub vector number $j = 1$;

Initialize the trend value $\mu_j^0 = 0$;

Step 4 Sub vector update times $k = 1$;

Calculate the v_j^k by using eq.(9);

Calculate the fitness value, and the by μ_j^k Eq.(11)

$f(\mathbf{x}_0^k) = f(x_1^{k-1}, x_2^{k-1}, \dots, x_j^{k-1} + v_j^k, \dots, x_D^{k-1})$;

$N_{FEs} = N_{FEs} + 1$;

If $f(\mathbf{x}_0^k) > f(\mathbf{x}_0^{k-1})$, $x_j^k = x_j^{k-1} + v_j^k$;

$\mu_j^k = 0.382 \mu_j^{k-1} + 1.382 v_j^k$;

If $f(\mathbf{x}_0^k) \leq f(\mathbf{x}_0^{k-1})$

If $\sum_{j=1}^m |\mu_j^k| > \varepsilon$, then

$x_j^k = x_j^{k-1}$; $\mu_j^k = \mu_j^{k-1} / s$;

If $\sum_{j=1}^m |\mu_j^k| \leq \varepsilon$,

$\mu_j^k = 0.15 / D \times f_{min}(-\varepsilon, \varepsilon) - 0.236 \ln \mu_j^k$;

If $k < N$, $k = k + 1$ go to Step 4;

If $j < D$, $j = j + 1$, go to Step 3;

If loop < Nloop, loop=loop+1, go to Step 2;

Update the estimating value σ by Eq.(8).

End of the function FrOptimization(i)

If $i < n$, $i = i + 1$, go to Step 1;

End of initial particles optimization in FBM mode.

Set the parameter N_{PSOMax}

Begin the classical PSO, $k=1$

Step 5 Update the velocity and position by Eq.(1),Eq.(2)

Calculate the fitness for the new position

If $k < N_{PSOMax}$, $k=k+1$, go to step 5;

End of classical PSO

$N_{FEs} = N_{FEs} + N_{PSOMax}$;

The current best particle $i_{g_{best}}$ Optimization by FrOptimization($i_{g_{best}}$)

$N_{FEs} = N_{FEs} + N_b \times N \times D$;

End

Figure 2. Pseudocode for the FEPSO algorithm

- (3) If the particle fitness value is still not improved after several iterations, μ will be reduced to a smaller value than ε . μ is set to the sum of the opposite velocity and additional random value by (11), which means that the velocity changes with greater diversity, more easily escape from local optima.

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

The six standard composite test functions CF₁~ CF₆ studied in [16] is used to test the performance of FEPSO. The composite test function is constructed by the following equation:

$$CF(x) = f_{bias} + \sum_{i=1}^n \{w_i(f'_i((x - o_i + o_{iold}) / \lambda_i M_i) + bias_i)\} \quad (12)$$

Where $CF(x)$ represent the new composition function, $f_i(x)$ is the i th basic function used to construct the composition function. They are the Sphere Function, Rastrigin's Function, Weierstrass Function, Griewank's Function, and Ackley's Function defined in [16]. n is the number of basic functions. D is the dimension. M_i is the orthogonal rotation matrix for each $f_i(x)$, o_i is the new shifted optimum position for each $f_i(x)$, o_{iold} is the old optimum position for each $f_i(x)$. Using $o_i, bias_i$, a global optimum can be placed anywhere. w_i is the weight value for each $f_i(x)$, calculated as below:

$$w_i = \exp\left(\frac{\sum_{k=1}^D (x_k - o_{ik} + o_{iold})^2}{2D\sigma_i^2}\right) \quad (13)$$

Where σ_i is used to control each $f_i(x)$'s coverage range, a small σ_i gives a narrow range for $f_i(x)$. λ_i is used to stretch or compress the function, $\lambda_i > 1$ means stretch, $\lambda_i < 1$ means compress. Since basic function has different search range, in order to make full use of the basic function, λ_i usually is set as the following:

$$\lambda_i = \sigma_i \times \frac{X_{max} - X_{min}}{x_{maxi} - x_{mini}} \quad (14)$$

$[X_{min}, X_{max}]$ is $CF(x)$'s search range. $[x_{mini}, x_{maxi}]$ stands for the $f_i(x)$'s search range. And $f_i(x)$

$$f'_i(x) = C \times \frac{f_i(x)}{|f_{maxi}|} \quad (15)$$

Where C is a predefined constant. $|f_{maxi}|$ is the biggest function value for 10 function $f_i(x)$, is an estimated value by the equation :

$$|f_{maxi}| = f_i\left(\frac{X_{max}}{\lambda_i} \times M_i\right) \quad (16)$$

In [16], CF1-CF6 composition test functions is constructed by the equation 11 to 15. These functions with a large number of local minima are recognized as difficult function optimizations [13]. Most of the optimization algorithm in the optimization process is easy

to fall into local minima. To compare the test optimization results, the other PSO algorithm parameters setting and experimental results are obtained from [13]. In the FEPSO, the Hurst parameter is estimated by $\sigma_1 = \sigma_{f_{random}}(0.2, 1.0)$ in (8).

First FEPSO run 10 times to optimize each standard composite test function. The maximum fitness evaluations including each particle FBM optimization phase and the final FBM optimization phase are set at 50,000. The mean and standard deviation values of the results are recorded and compared with [13] in Table I. It is observed that FEPSO has achieved better result compared with ISPO algorithm on all the six composition functions. Compared with the other algorithms, FEPSO only on function CF4 has not gotten a best result. On the other test functions FEPSO algorithm also has obvious advantages. Then FEPSO run 20 times, 30 times and 60 times respectively. The mean and standard deviation values of the test results shown in Table II have less changes with run times. So the FEPSO optimization results have robustness.

TABLE I.

RESULTS ACHIEVED BY USING FEPSO ON SIX COMPOSITION FUNCTIONS

Test function	Different Algorithm			
	FEPSO	CPSO	CLPSO	ISPO
CF ₁	2.53e-07±3.66e-07	1.39e+02 ±2.79e+01	1.41e+02 ±2.13e+02	2.47e+01 ±7.79e+01
CF ₂	3.13e+01 ±3.66e+01	1.15e+02 ±3.48e+01	1.12e+02 ±1.40e+02	6.54e+01 ±6.03e+01
CF ₃	2.06e+02 ±4.15e+01	2.27e+02 ±6.45e+01	3.01e+02 ±1.95e+02	2.12e+02 ±6.06e+01
CF ₄	3.90e+02 ±4.38e+01	3.77e+02 ±4.79e+01	3.32e+02 ±1.26e+02	4.11e+02 ±5.87e+01
CF ₅	2.44e+01 ±2.48e+01	1.11e+02 ±2.47e+01	1.95e+02 ±2.36e+02	3.37e+01 ±2.88e+01
CF ₆	4.94e+02 ±1.60e+01	6.71e+02 ±1.58e+02	8.39e+02 ±1.44e+02	6.95e+02 ±2.07e+02

TABLE II.

RESULTS ACHIEVED BY USING FEPSO ON SIX COMPOSITION FUNCTIONS IN DIFFERENT TIMES

Test function	Different Times			
	10	20	30	60
CF ₁	2.53e-07±3.66e-07	1.79e-07 ±2.35e-07	1.87e-07 ±2.57e-07	1.82e-07 ±2.78e-07
CF ₂	3.13e+01 ±3.66e+01	3.92e+01 ±4.50e+01	3.43e+01 ±4.15e+01	4.65e+01 ±5.07e+01
CF ₃	2.06e+02 ±4.15e+01	2.00e+02 ±4.00e+01	1.96e+02 ±4.44e+01	2.02e+02 ±5.17e+01
CF ₄	3.90e+02 ±4.38e+01	4.02e+02 ±3.83e+01	4.02e+02 ±3.60e+01	4.02e+02 ±4.18e+01
CF ₅	2.44e+01 ±2.48e+01	2.86e+01 ±2.81e+01	2.58e+01 ±2.36e+01	2.43e+01 ±2.05e+01
CF ₆	4.94e+02 ±1.60e+01	4.95e+02 ±1.43e+01	4.93e+02 ±2.02e+01	4.96e+02 ±4.42e+01

The convergence characteristics compared with each other algorithm in the optimization of the composite test functions is shown on Fig.3, where the other algorithm convergence characteristics data is obtained from the Fig.3 in [13]. The horizontal axis is the function evaluation times in the range of 0 ~ 5 × 10⁴, the vertical axis is objective function value. In the Fig.3, "FEPSO

mean” stands for the average convergence curve and “FEPSO best” stands for the best convergence curve achieved by FEPSO algorithm. FEPSO has achieved better convergence speed and is significantly faster than other algorithms on the composite test function of CF₂, CF₃, CF₅ and CF₆.

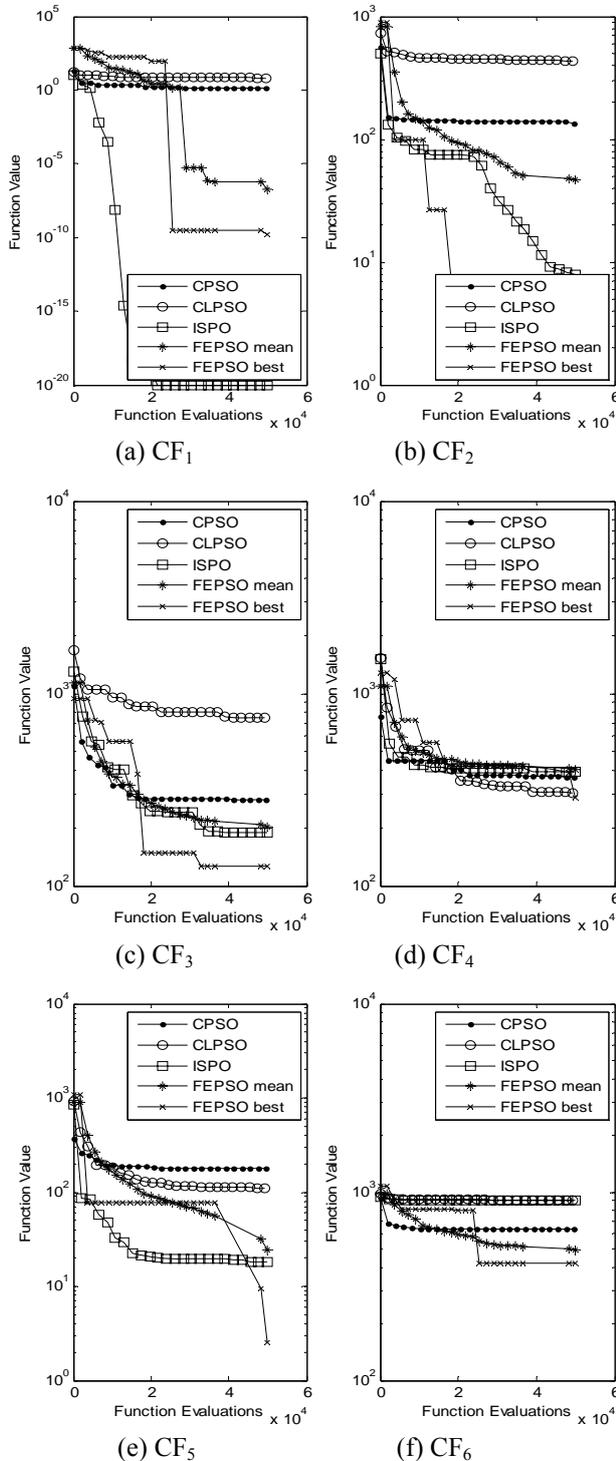


Figure 3. Convergence graph of FEPSO algorithm compared with [13]

IV. CONCLUSIONS

After analysis of the existing PSO algorithms, the particles should include a self-optimize exploration phase in order to effectively solve complex function optimization problem. Particle self-optimize exploration phase is very similar as the irregular movement of FBM. A new Fractal Evolutionary particle swarm optimization(FEPSO) framework is introduced. In FEPSO, each particle has FBM optimization phase in the beginning, then enter the classical PSO phase, and finally get the optimal result after the current best particle FBM optimization again. In this paper because the fractal parameter estimation is difficult, one-dimensional FBM is used to update particle velocity and location. The numerical simulation results show that the FEPSO has a strong global search capability to obtain optimal results for the composite test function mentioned in [13]. It is worthwhile to continue the further study in the application[17,18].

ACKNOWLEDGMENT

The research is supported by the National Natural Science Foundation of China(Grant No.60674054) and the National Natural Science Foundation of Jiangxi Province, China(Grant No. 2010GZC0090).

REFERENCES

- [1] Kennedy J, Eberhart R C. “Particle swarm optimization”. In: *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, IEEE Service Center, 1995,pp194~198.
- [2] Shi Y, Eberhart R C. “A modified particle swarm optimizer”. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, IEEE Press, 1998,pp69~73.
- [3] Shi Y, Eberhart R C. “Fuzzy adaptive particle swarm optimization”. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, Seoul, Korea, 2001,pp101~106.
- [4] Corne D, Dorigo M, Glover F. “*New ideas in optimization*”. McGraw Hill, 1999,pp379~387.
- [5] Clerc M. “The swarm and the queen: toward a deterministic and adaptive particle swarm optimization”. In: *Proceedings of the Congress on Evolutionary Computation*, 1999,pp1951~1957.
- [6] Angeline P J. “Evolutionary optimization versus particle swarm optimization: philosophy and performance differences”. In: *Proceedings of 7th Annual Conference on Evolutionary Programming*, Germany, 1998,pp601~610.
- [7] Angeline P J. “Using selection to improve particle swarm optimization”. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, 1998,pp.84~89.
- [8] Suganthan P N. “Particle swarm optimizer with neighborhood topology on particle swarm performance”. In: *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1999,pp. 1958~1962.
- [9] Kennedy J. “Small worlds and Mega-minds: effects of neighborhood topology on particle swarm performance”. In: *Proceedings Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1999,pp. 1931~1938.
- [10] Peram T, Veeramachaneni K, Mohan C K. “Fitness-distance-ratio based particle swarm optimization”. In: *Proceedings Swarm Intelligence Symposium*, Indianapolis, Indiana, USA, 2003,pp.174~181.

- [11] Bergh F, Engelbrecht A P. "A cooperative approach to particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, vol8,no.3,2004, pp.225~239.
- [12] Liang J J, Qin A K, Suganthan P N et al. "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions". *IEEE Transactions on Evolutionary Computation*, vol10, no.3,2006, pp.281~295.
- [13] Ji Zhen,Zhou Jiarui,Liao Huilian, et al. "A novel intelligent single particle optimizer". *Chinese Journal of computers*, vol33,no.3,2010, pp.556~561.
- [14] Hu Y , Oksendal B. "Fractional white noise calculus and application to finance". *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, vol1,no.6,2003 ,pp1-32.
- [15] Xiaohong Qiu, Jun Liu , "The Random Factor in Particle Swarm Optimiazation", *Proc. of IEEE International Conference on Intelligent Computing and Intelligent Systems(ICIS)*, Shanghai,China. 2009,pp787-791.
- [16] Liang J J, Suganthan P N, Deb K. "Novel composition test functions for numerical global optimization". *Proceedings of IEEE International Swarm Intelligence Symposium*, Messina, Italy, 2005,pp 68~75.
- [17] Liangyou Shu, Lingxiao Yang. "A Modified PSO to Optimize Manufacturers Production and Delivery". *Journal of Software*,vol7,no.10,2012,pp.2325-2332.
- [18] Zhibin Liu, Ling Zhang, Xiangsong Meng. "A Novel Hybrid Stochastic Searching Algorithm Based on ACO and PSO: A Case Study of LDR Optimal Design". *Journal of Software* ,vol6,no.1,2011,pp.56-63.



Xiaohong Qiu Jiangxi Province, China. Birthdate: April, 1967. is Vehicle Control Guidance and Simulation Engineering Ph.D., graduated from the Department of Automation, Beijing University of Aeronautics and Astronautics. And research interests on Intelligent Control and Computer Application Technology.

He is currently a professor of Software School, Jiangxi University of Science and Technology. Previously he held positions with Jiangxi Agricultural University, Jiangxi Normal University, Beijing University of Aeronautics and Astronautics.

Dr. Qiu is a Fellow of Jiangxi Computer Association.



Xiaohui Qiu Jiangxi Province, China. Birthdate: Jul, 1970. is Higher Education M.S, graduated from Institute of Higher Education, Jiangxi Agricultural University. And research interests on Human Resource Management, Science and Education.

He is currently an associate professor of Institute of Higher Education, Jiangxi Agricultural University.

Mr.Qiu is a Fellow of Jiangxi University Personnel Management Association.



Fang Liao Henan Province, China. Birthdate: Nov, 1970. received Master Degree in Vehicle Control Guidance and Simulation Engineering from the Department of Automation, Beijing University of Aeronautics and Astronautics, and Ph.D in Science in Computer Control and Automation from the School of Electrical and Electronic Engineering at Nanyang Technological

University, Singapore. And research interests on Intelligent Control and Computer Application Technology.

She is currently a scientist of Temasek Laboratories, National University of Singapore.