

Application of Particle Swarm Optimization in Figuring out Non-differentiable Point of Function

Mo Yuanbin

1 School of statistics and mathematics, Zhongnan University of economics and law, Wuhan 430073, China
 2 Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning, 530006 China
 Email: moyuanbing@263.net

Zhao Xin-quan and Xiang Shu-jian

School of statistics and mathematics, Zhongnan University of economics and law, Wuhan 430073, China
 Email: zxq556@163.com, xiangsj63@yahoo.com.cn

Abstract—Owing to the importance of non-differentiable point in a function for economy, engineering and theoretical analysis, this paper brings forward a novel methodology to figure out the non-differentiable point of function which is based on adjusting the models for global extremum and local extremum of particle swarm optimization (PSO). The algorithm takes the difference between left and right difference quotients as the adaptive value of the particle. By them, it defines local extremum and global extremum for PSO and makes particle close to non-differentiable point of target function, and then figures the point out. The validity of the algorithm is verified by the result of numerical calculation.

Index Terms—Non-differentiable Point; Particle Swarm Optimization (PSO); Difference Quotient

I. INTRODUCTION

The non-differentiable point of function is importantly applied in analyzing the characteristics of function, economy and engineering practices. It will be used in figuring out the extremum of function. Therefore, it is actually and theoretically significant to figure out the point of function within certain interval. However, only few scholars pay attention to how to figure out non-differentiable point of function within certain interval until now.

It is not easy to figure out non-differentiable point of function. According to definition, the point has to be estimated one by one. That is obviously not an operable method. And there is no Iterative formula can be used to calculate non-differentiable point as used in other mathematic problems.

In this paper, by adjusting Particle Swarm Optimization (PSO)^[1,3-10] properly, the algorithm can be more suitable to figure out non-differentiable point of function.

And there are examples in the following text which shows the validity of the algorithm.

II. NON-DIFFERENTIABLE POINT OF FUNCTION

Derivable point^[2] x_0 of function is the point that the follow limit is existed.

$$\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} \quad (1)$$

Non-differentiable point of function is the point that the limit is not existed.

According to the above definition, the non-differentiable point of function has to be figured out one by one to testify the existence of the limit. In this way, it is actually not feasible to figure out the point.

Particle Swarm Optimization (PSO) is a kind of optimized algorithm designed after the simulation of foraging behavior of birds. It imitates the group biological behavior of birds and the like. In it every organism is called as particle. PSO means to find out the optimum solution among a group of particles in n -dimensional space. The position of every particle means the vector for one solution. As an agent, the particle may memorize the optimum solution found by itself and acquire the optimum solution experienced by the whole group of particles to direct its movement and gradually iterate the most optimum solution.

Take the position of No.i particle in generation-j as $\mathbf{x}_i(j)$, the optimum solutions found out by itself and the whole group of particles are individual extremum (Pbest) $\mathbf{p}_{id}(j)$ and global extremum (Gbest) $\mathbf{p}_{gd}(j)$ respectively, its movement speed vector $\mathbf{v}_i(j)$, then No.i particle will move to the following position in generation-(j+1)

$$\mathbf{v}_i(j+1) = \omega\phi_1\mathbf{v}_i(j) + \eta_1\phi_2(\mathbf{p}_{id}(j) - \mathbf{x}_i(j)) + \eta_2\phi_3(\mathbf{p}_{gd}(j) - \mathbf{x}_i(j)) \quad (1)$$

$$\mathbf{x}_i(j+1) = \mathbf{x}_i(j) + \mathbf{v}_i(j+1) \quad (2)$$

Among which ω, η_1, η_2 are control parameters all taking values from $[0, 1]$, and ϕ_1, ϕ_2, ϕ_3 are random numbers in the interval $[0, 1]$. The movement of every particle may be illustrated as follows

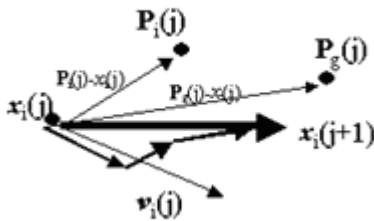


Fig.1 Complex Movement of Particle

III. FIGURE OUT NON-DIFFERENTIABLE POINT OF FUNCTION BY PSO

PSO adjusts itself in each step by contrasting the $p_{id}(j)$ and $p_{gd}(j)$ of its former step. In this way, the optimum value of function will be figured out after adjustment one step by one step. Owing to this thought, it is possible to figure out the solution for the certain characteristic by means of PSO iteration, if provided that $p_{id}(j)$ and $p_{gd}(j)$ have certain characteristic. Since the non-differentiable point of function is a point with certain characteristic, so it can be figured out by PSO.

For function $y = f(x), x \in [a, b]$

The key to figure out the non-differentiable point by PSO is how to confirm the global extremum and local extremum in PSO. Different global extremum and local extremum will bring forward different algorithm in figuring out non-differentiable point by PSO. According to the definition of derivative of function (1); if a function is not an odd function and is differentiable at certain point, the absolute value of left and right difference quotients at this point is relatively small;

$$p(x_0) = \left| \frac{f(x_0 + h) - f(x_0)}{h} - \frac{f(x_0 - h) - f(x_0)}{-h} \right| \quad (3)$$

If the function is at the non-differentiable point, the absolute value of left and right difference quotients at this point is relatively big.

If $p(x_0) = 0$ appears in the calculation of formula (3), it means $f(x)$ is an odd function within the local field of x_0 . If it is differentiable at x_0 , that

$$p(x_0) = \left| \frac{f(x_0 + h) - f(x_0)}{h} - \frac{f(x_0 + \frac{h}{2}) - f(x_0)}{\frac{h}{2}} \right| \quad (4)$$

is relatively small; if the function is at the non-differentiable point, the absolute value is relatively large.

In this paper, the absolute value of the difference between left and right difference quotients of every particle is used as the adaptive of this particle. The individual extremum and global extremum are same with general algorithm. In this way, PSO is used to figure out the non-differentiable point of function.

A. Flow of Algorithm

The basic procedure in figuring out mean value by PSO is:

Step1 $k = 1$ given $h, M > 0$

Given m initial points like x_1, x_2, \dots, x_m and initial speed $v_i (i = 1, 2, \dots, m)$,

Step2 Figure out respective adaptive values by formula (4) or formula (3) according to $f(x)$ whether or not it is odd function.

Find out the maximum among $p(x_1), p(x_2), \dots, p(x_m)$ and take it as $f(p_{gd})$.

Accordingly, the current global mean value point is p_{gd} .

Local mean value is $p_{id} = x_i$, calculate $f(p_{id})$.

Step3 Update according to iteration formula (1), (2).

Step4 Confirm the local mean value.

On the assumption that after the update of Step2, the m points and their speeds are x_1, x_2, \dots, x_m and $v_i (i = 1, 2, \dots, m)$ respectively, calculate

$$p(x_1), p(x_2), \dots, p(x_m)$$

Compare $p(x_i), p(p_{id})$, and take the maximum as the new $p(p_{id})$, the local mean value point is p_{id} .

Step5 Confirm the global mean value

Compare $f(p_{id}), (i = 1, 2, \dots, m)$ and $f(p_{gd})$, take the maximum as the new $f(p_{gd})$, the global mean value point p_{gd} and the global mean value $f(p_{gd})$ are acquired.

Step6 $k = k + 1$, judge if it conforms to the condition of conclusion. If it does, it may conclude. If not, return to Step2.

Step7 When the final maximum is bigger than M , it is considered that the function is non-differentiable at $x = x_{gd}$. When it is smaller than M , it is considered that the function is differentiable at every point within the interval $[a, b]$.

B. Performance Test of Arithmetic

Test: to test following functions in different types,

$$(1) \quad y = |x|, \quad |x| \leq 20$$

$$(2) \quad y = \begin{cases} 1 & 0 \leq x \leq 20 \\ 0 & -20 \leq x < 0 \end{cases}$$

$$(3) \quad y = \begin{cases} \sin \frac{1}{x} & -20 \leq x \leq 20, \text{ and } x \neq 0 \\ 0 & x = 0 \end{cases}$$

$$(4) \quad y = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$(5) \quad y = \begin{cases} 1 - \cos x & x \geq 0 \\ x & x < 0 \end{cases}$$

$$(6) \quad y = x^{\frac{2}{3}}, \quad |x| \leq 20$$

$$(7) \quad y = |\ln |x - 1||, \quad |x| \leq 0.5$$

The initial population of the algorithm is 100 and iteration generation number is 100.

Example 1 Parameter $h = 0.1$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure2 and the non-differentiable points was listed in the appendix A.

Example 2 Parameter $h = 0.01$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure3 and the non-differentiable points was listed in the appendix B.

Example 3 Parameter $h = 0.001$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure 4 and the non-differentiable points was listed in the appendix C.

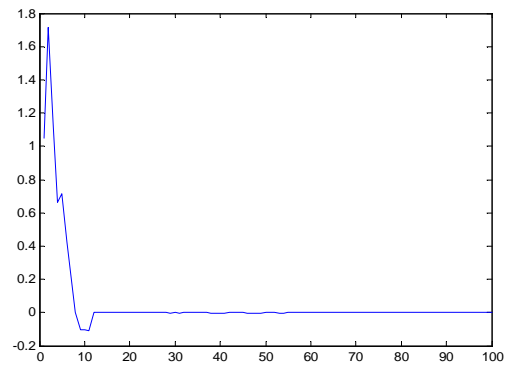


Fig.2 The curve of the relation the non-differentiable point and steps of example 1

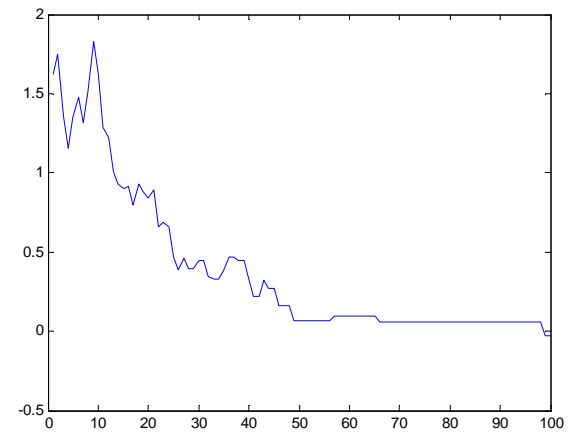


Fig.3 The curve of the relation the non-differentiable point and steps of example 2

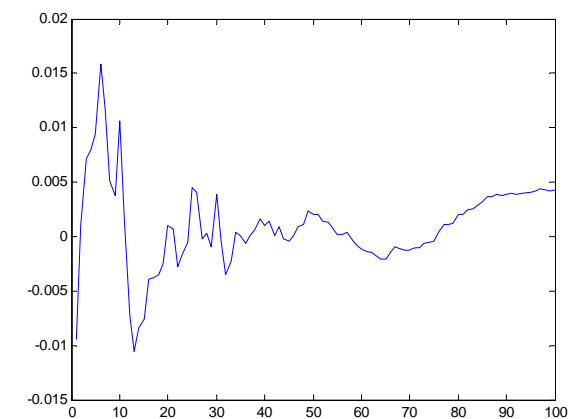


Fig.4 The curve of the relation the non-differentiable point and steps of example 3

Example 4 Parameter $h = 0.1$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure5 and the non-differentiable points was listed in the appendix D.

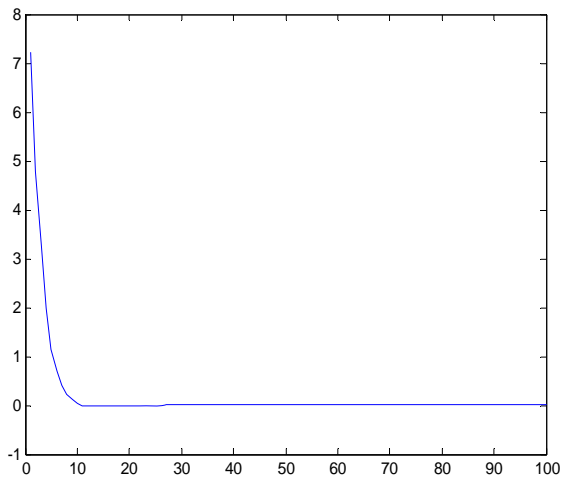


Fig.5 The curve of the relation the non-differentiable point and steps of example 4

Example 5 Parameter $h = 0.1$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure 6 and the non-differentiable points was listed in the appendix E.

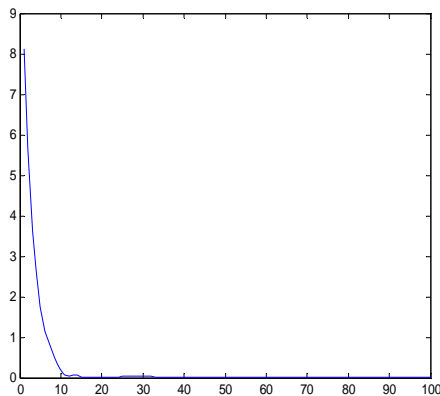


Fig.6 the curve of the relation the non-differentiable point and steps of example 5

Example 6 Parameter $h = 0.1$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure7 and the non-differentiable points was listed in the appendix F.

Example7 Parameter $h = 0.1$, after 200 times of calculation, the curve of the average of the acquired non-differentiable points was given in figure 8 and the non-differentiable points was listed in the appendix G.

From the above results, it is obvious that the algorithm can figure out non-differentiable points of function relatively and correctly within certain interval. This is because the algorithm makes proper definition for local extremum and global extremum in PSO, which may

move closely to the non-differentiable points of function and finally figure them out.

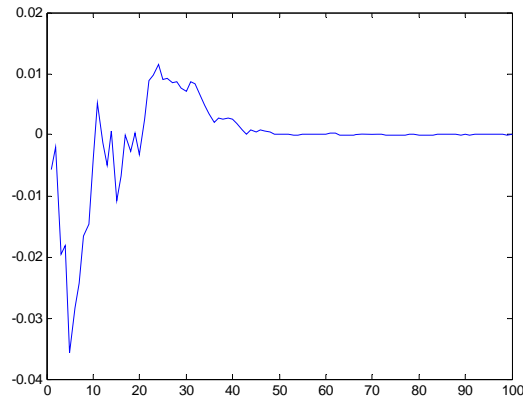


Fig.7 The curve of the relation the non-differentiable point and steps of example 6

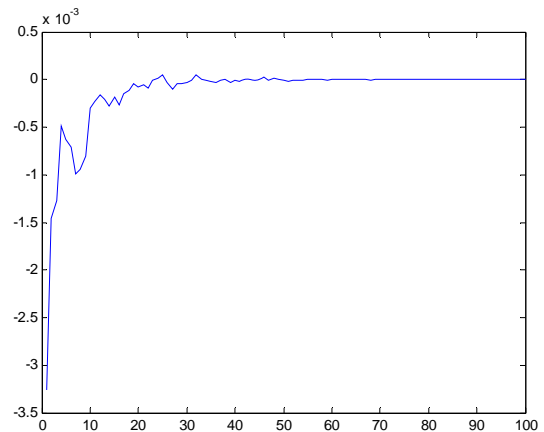


Fig.8 the curve of the relation the non-differentiable point and steps of example 7

C. Analysis of Time and Capability of Algorithms

The calculation of the algorithm is same with the calculation of standard PSO without additional calculation. Therefore, it is same with PSO in complicity of calculation. Through the above test, good result is figured out. This is because PSO is a kind of global search arithmetic. During the course of iteration, the main trend is always moving forward the optimum value. As a result, after global mean value and local mean value are defined in this paper, the main trend of the algorithm moves forward the non-differentiable points of function and finally figure out the non-differentiable points of function.

Performance analysis: The results of the above showed that the algorithm has strong searching property for figuring out Non-differentiable Point of function if the function exist the non-differentiable Point. It is because that the algorithm can fully take advantage of the idea of the PSO. By the definition of the non-differentiable point of the function, it gave extremum and global extremum. From the extremum, we can see that it fully use the

change rate in the neighborhood of one point of the function. So, in the course of the iteration, the extremum point closer to the non-differentiable by step and step, and finally find out the non-differentiable.

Convergence analysis: Since the algorithm is based on PSO, and no other performance was added to it. Then, if the non-differentiable is exists, the algorithm will be convergence to it

Time Complexity analysis: The algorithm is based on PSO, and only advance the extremum and global extremum definition of the non-differentiable, in order to make PSO can use to find out the non-differentiable, so the time complexity of this algorithm is same to the PSO's, and their run-time is similar.

IV. APPLICATION OF ALGORITHM

We all know that it is important to solving optimization problems, so lots of algorithm was advanced to solve it. Especially, evolution algorithm provides more possibility and advantage for optimization algorithm to solve optimization problems. But evolution optimization algorithm is easy to be trapped into local minima in optimizing it, and lots of improve tactics^[11-39] was put forward to improve evolution algorithm. Because, we know that the non-differentiable point is quietly possible the minima point, so it is entirely to improve one optimization algorithm by leading searching non-differentiable points mechanism to it to enhance the searching of the algorithm.

For example, we can improve Particle Swarm Optimization by added non-differentiable points searching to it (PSOS), and the principle is as follows::

$$\min f(\mathbf{x})$$

Step 1: Randomly produce k particles $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ in solution space, and given $h > 0$

Step2 Figure out respective adaptive values by formula (4) or formula (3) according to $f(\mathbf{x})$ whether or not it is odd function. And find out the maximum among $p(\mathbf{x}_1), p(\mathbf{x}_2), \dots, p(\mathbf{x}_m)$, let it is $p(\mathbf{x}_0)$

Step 3: Compute the current global optimum \mathbf{p}_g and local optimum \mathbf{p}_i , substitutes iterative equations with the values

$$\begin{aligned} \mathbf{v}_i(t+1) &= \omega \mathbf{v}_i(t) + c_1 \phi_1(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 \phi_2(\mathbf{p}_g(t) - \mathbf{x}_i(t)) \\ &+ c_3 \phi_3(\mathbf{x}_{0i} - \mathbf{x}_i(t)) \\ \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (i = 1, 2, \dots, k) \end{aligned}$$

Compute $f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_k)$

Step 4: Return to Step 2, repeat the process until a stop criterion is met.

One benchmark functions are given to examine the performance of PSOS.

Test1 30 dimensions Griewank function

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos(x_i / \sqrt{i}) + 1 \\ |x_i| &\leq 600. \end{aligned}$$

The best result is $\min f(\mathbf{x}) = f(0, 0, \dots, 0)$.

Test 2 Hartman's Function

$$f(\mathbf{x}) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2],$$

where $0 \leq x_j \leq 1, c = (1 \ 1.2 \ 3 \ 3.2)$,

$$(p_{ij}) = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1415 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$$

$$A = (a_{ij}) = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$$

$\min(f(\mathbf{x})) = f(0.201, 0.15, 0.477, 0.275, 0.311, 0.657) = -3.32$.

PSOS and PSO was run 10 times, and the final optimal results showed that PSOS is better than PSO

V. FURTHER EXTENSION OF ALGORITHM

By using this algorithm, the non-partially differentiable point of multi-function can also be figured out. Meanwhile, the maximum and minimum of function may be calculated in combination of this algorithm.

VI. CONCLUSION

In this paper, a novel method of constructing local extremum and global extremum in PSO is given according to the need of figuring out the non-differentiable point of function. By this method, PSO can be used to calculate the non-differentiable point of function at a certain interval. The results in the paper show the algorithm is practical.

APPENDIX A THE DATE OF EXAMPLE 1

0.0002	-0.0000	-0.0000	-0.0000	0.0000	0.0001
0.0011	0.0000	-0.0000	0.0045	-0.0000	0.0000
0.0000	-0.0000	-0.0000	0.0000	-0.0000	-0.0000 ...

APPENDIX B THE DATE OF EXAMPLE 2

0.0050	-0.8458	0.0043	0.0037	0.0022	0.0050
-0.0041	0.0011	0.0013	0.0035	0.0028	0.0024
0.0029	0.0048	0.0049	0.0009	0.0001	-0.0016...

APPENDIX C THE DATE OF EXAMPLE 3

-0.0033	-0.0054	-0.0033	0.0075	0.0045	0.0075
0.0036	0.0075	0.0075	0.0075	-0.0036	0.0075
0.0075	0.0075	0.0075	0.0075	0.0075	0.0075 ...

APPENDIX D THE DATE OF EXAMPLE 4

0.0001 0.0000 0.0049 0.0191 0.0016 0.0000
 0.0392 0.0056 0.0148 0.0000 0.0000 0.0164
 0.0137 -0.0216 0.0000 0.0088 0.0037 0.0000 ...

APPENDIX E THE DATE OF EXAMPLE 5

-0.0001 0.0212 0.0002 0.0056 0.0000 0.0178
 0.0000 0.0000 0.0176 0.0001 0.0062 0.0000
 0.0208 0.0000 0.0038 0.0110 0.0014 0.0356

APPENDIX F THE DATE OF EXAMPLE 6

1.0e-003 *
 -0.0083 -0.0244 -0.0227 -0.0107 -0.0638 0.1712
 0.0216 -0.0200 -0.0085 -0.0088 -0.0437 -0.0067 -
 0.0050 -0.0089 -0.0229 -0.0516 -0.0194 0.1974...

APPENDIX G THE DATE OF EXAMPLE 7

1.0E-005 *
 -0.1851 0.0515 -0.2821 -0.0893 0.0207 0.0264 -
 0.1997 -0.1132 -0.3145 0.0292 -0.0787 0.0647
 0.1891 0.0655 -0.2352 0.3584 -0.1287 -0.3723....

REFERENCES

[1] KENNEDY J, EBERHART R C. Particle swarm optimization [A]. Proceedings of IEEE International Conference on Neural Networks. Piscataway NJ: IEEE , 1995: 1942-1948.

[2] Mathematical Analysis(M), Department of Mathematics, East China Normal University. China Higher Education Press,2009.

[3] Lu Yan-ping, Wang Sheng-rui. Li Shao-zi,etal. Particle swarm optimizer for variable weighting in clustering high-dimensional data [J]. Machine Learning, 2011,82:43-70

[4] Kulkarni R V, Venayanamoorthy G K. Particle swarm optimization in wireless-sensor networks: a brief survey [J]. IEEE Transaction on Systems, Man and Cybernetics-Pat C: Applications and Reviews, 2011:262-267.

[5] Chen Guo-long, Guo Wen-zhong, Chen Yu-zhanong. A PSO –based intelligent decision algorithm for VLSI floor planning [J]. Soft Computing, 2010,14(12), 1329-1337

[6] Liu H, Cai Z x, Wang Y . Hybridizing particle Swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing, 2010, 10 (2): 629-640 .

[7] Kim P, Lee J. A n integrated method of particle swam optimization and differential evolution. Journal of Mechanical Science and Technology, 2009, 23(2): 426 — 434.

[8] Onwubolu, G. C. and Clerc, M., "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization," *International Journal of Production Research*, vol. 4 pp. 473-491, 2004.

[9] Rana S, Jasola S , Kumar R. A review on particle swarm optimization algorithms and their applications to data clustering. Artificial intelligence Review, 2011,35 (3): 211-222.

[10] Glover F. Tabu search-part I. ORSA Journal on Computing, 1989, 1(3): 190-206.

[11] Zhang C S, Ning J X, Lu S, Ou yang D T , Ding T N . A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Operations Research Letters, 2009,37(2): 117-122.

[12] Zhan Z H,Zhang J.Discrete particle swarm optimization for multiple destination routing problems [C]. Evo Workshops, 2009:117-122.

[13] Kadirkamanathan V,Selvarajah K,Fleming P J.Stability analysis of the particle dynamics in particle swarm optimizer [J]. IEEE Ttransaction on Evolutionary Computation,2006,10(3):245-255.

[14] Noel MM,Jeannette T C.Simulation of a newhybrid particle swarm optimization algorithm. Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory,2004:150-153

[15] Wachowiak MP,Smolfkova R,Zheng Y,et al.An approach to Multimodal Biomedical Image Registration Utilizing Particle Swarm Optimization. IEEE Transac -tions on Evolutionary Computation, 2004, 8 (3) :289- 301

[16] Victoire TAA,Jeyakumar AE.Hybrid PSO- SQP for economic dispatch with valve- point effect.Electric Power Systems Research,2004,71(1):51- 59.

[17] Shi X,Lu Y,Zhou C,et al. Hybrid evolutionary algorithms based on PSOand GA.Proceedings of IEEE Congress on EvolutionaryComputation(CEC),Canberra, Australia, 2003:2393- 2399.

[18] LIU Wei; CAI Qian-feng1; LIU Hai-lin. Particle swarm optimization algorithm based on parametric equation method to handle equality constraints[J]. Computer Engineering and Design. 2008 , 29 (3) ,697-699

[19] CAO Chun-hong; ZHANG Bin; LI Wen-hui. The Research Based on the Composite Particle Swarm Optimization Algorithm in the Geometric Constraint Solving. Journal of Image and Graphics; 2007, 12(4),713-717.

[20] WANG Jin-hua; YIN Ze-yong. Solving constrained discrete optimization problem with Particle Swarm Optimizer. Computer Engineering and Applications ; 2008,44(3), 242-244.

[21] Wei Jing-xuan; Wang Yu-ping. Fuzzy Particle Swarm Optimization for Constrained Optimization Problems . Journal of Electronics & Information Technology; 2008, 30(5), 1218-1221.

[22] WANG Fang; DING Haili; GAO Chengxiu. Advanced Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands. Journal of Wuhan University (Natural Science Edition); 2007, 53(1), 41-44.

[23] LI Hong-mei; SUN Jun; XU Wen-bo. Empirical study based on Quantum-behaved Particle Swarm Optimization stochastic programming algorithm. Computer Engineering and Applications; 2007, 43(24), 185-189.

[24] ZHAO Pei-xin; WANG Hong; MA Jian-hua. Particle Swarm Optimization Algorithm for Stochastic Loader Problem. Computer Engineering and Applications; 2006(19), 215-218.

[25] LU Lin; TAN Qing-mei. Hybrid particle swarm optimization algorithm for stochastic vehicle routing problem. Systems Engineering and Electronics; 2006, 28(2), 244-247.

[26] SUN Kai ; GE Xin-sheng. Optimal controf of stretching process od solar array on spacecraft using particle swarm optimizatin algorithm. Engineering Mechanics ; 2007, 24(9), 188-192.)

[27] Li Hong; Zhang Bing. Application of Particle Swarm Optimization in Balance Control for the Acrobot. Microcomputer Information; 2006,22,(3-1),80-82.

- [28] GUAN ShengTao; CHU Ji Zheng; SHAO Shuai. Application of nonlinear model predictive control based on particle swarm optimization. *Journal of Beijing University of Chemical Technology (Natural Science Edition)*; 2007,34 (6),653-656.
- [29] MA Chang-xi; QIAN Yong-sheng; WANG Chun-lei. Urban loop-road traffic-coordination-control system based on TWP-PSO. *Journal of Computer Applications*; 2007, 27(11), 2640-2642.
- [30] MO Yuan-bin; CHEN De-zhao; HU Shang-xu. Particle Swarm Optimization for Multi-Objective Process System Optimization Problems. *Journal of Chemical Engineering of Chinese Universities*; 2008, 22(1), 94-99.
- [31] Schutte, J. F. and Groenwold, A. A., "Sizing design of truss structures using particle swarms," *Structural and Multidisciplinary Optimization*, 2003, vol. 25, no. 4, pp. 261-269.
- [32] HE Yijun; YU Huanjun; CHENG Biao; CHEN Dezhao. Multi-objective particle swarm optimization approach to solution of fed-batch bioreactor dynamic multi-objective optimization. *Journal of Chemical Industry and Engineering(China)*; 2007, 58(5), 1262-1270.
- [33] ZHANG Wen-ming; DONG Zeng-chuan; ZHU Cheng-tao; QIAN Wei. Automatic calibration of hydrologic model based on multi-objective particle swarm optimization method. *Journal of Hydraulic Engineering*; 2008, 39(5), 528-534.
- [34] PENG Zhi-ping; CHEN Ke. A Multi-Objective Particle Swarm Optimization Algorithm for Solving Negotiation Deadlock. *Acta Electronica Sinica*; 2007, 35(8), 1452-1457.
- [35] JIN Xin-Lei; MA Long-Hua; WU Tie-Jun; QIAN Ji-Xin. Convergence Analysis of the Particle Swarm Optimization Based on Stochastic Processes. *Acta Automatica Sinica*; 2007 33 (12) 1263-1268.
- [36] CUI Hong-mei; ZHU Qing-bao. Convergence analysis and parameter selection in particle swarm optimization. *Computer Engineering and Applications*; 2007, 43(23), 89-92.
- [37] LIU Hong-bo; WANG Xiu-kun; TAN Guo-zhen. Convergence Analysis of Particle Swarm Optimization and Its Improved Algorithm Based on Chaos. *Control and Decision*; 2006, 21(6), 636-641
- [38] LUO Jin-yan; JIANG Zhong-liang. Stability Analysis of the Particle Swarm Optimization Algorithm. *Journal of Jimei University (Natural Science)*; 2007, 12(4) 376-379.
- [39] Correa E S, Freitas A A, Johnson G G. A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set [C]// Genetic and Evolutionary Computation Conference. New York: ACM, 2006:35-42

Yuanbin Mo received the B.S. degree from the Guangxi University, Nanning, China, in 1991 and the M.S. degree from Guizhou University, Guiyang, China, in 2001 and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2007. Currently, he is a Associate Professor at Guangxi University for Nationalities, Nanning, China. His research interest covers computational intelligence and complex network theory.