An Algorithm Based on the Improved Particle Swarm Optimization

Ri-Bo GE
City Institute Dalian University of Technology
Dalian 116600, China
ececity@dlut.edu.cn

***Abstract*—As Aiming at the shortage of the basic particle swarm algorithm is easy to fall into local solution，this paper makes the improvement to the inertia weight factor based on basic particle swarm algorithm called Non-Linear Decreasing Random Inertia Weight Particle Swarm Optimization. Experiments show that the performance of the improved algorithm outperforms standard algorithm.**

Keywords-particle swarm optimization; parameter estimation; inertia weight

I. INTRODUCTION

Regression analysis can be used to predict and control, in natural science, social science and technology. It has important applications, and is the most important tool statistics. By regression analysis, multivariate linear regression analysis is the most widely used multivariate quantitative analysis method. It is widely application domain in industry, agriculture, medicine, social investigation, biological information processing.

In the regression analysis, the maximum likelihood estimation method is a basic method when estimating the unknown system parameters, before the application of the maximum likelihood estimation method, a function of the unknown model parameters is first established, called Likelihood Function. Maximum likelihood estimation of the parameters of the model is the parameter estimate that is the choice of the maximum value of the likelihood function[1]. Due to high order matrix inverse operation, the conventional maximum likelihood estimation calculation process is complicated, and the program structure is not universal. In recent years, along with the development of the genetic algorithm, simulated annealing technique, ant colony algorithm and particle swarm algorithm, the use of intelligent algorithm for parameter estimation of multivariate regression model has gradually become a hot. Yuguang Chen[2] propose a multivariate regression model of engine based on genetic algorithm according to the parameters of car engine; Qi Shen [3] put forward the particle swarm optimization algorithm is used to select multiple linear regression variables, and applies it to the relationship study of the chemistry of aromatic amine carcinogenic activity of structure-activity; Xinjie Wu[4] the particle swarm algorithm is used to estimate the parameters in multivariate linear regression, and three linear regression analysis was verified as an example; Qian Huo[5] put forward the application of genetic algorithm is used to the study of multivariate nonlinear regression model parameters, and with examples, the validity and practicability of the algorithm is verified.

For the maximum likelihood estimation method, this paper presents a multiple linear regression model parameter estimation algorithm based on the improved particle swarm optimization algorithm, and the algorithm is simple, practical and efficient.

II. PARTICAL SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a kind of the evolutionary computation technique based on swarm intelligence, this technique was first proposed by Eberhart and Kennedy in 1995[6], from the research on the bird (fish) preying behavior. Its main advantage is simple, easy to implement, less parameters, and producing high quality solutions in a relatively short period of time. A large number of testing functions show that, it has faster convergence speed compared with the traditional optimization techniques [7, 8].

PSO initializes into a group of random particle (stochastic), by tracking the optimal particle to search for the optimal solution. When PSO in solving optimization problems, each candidate solution is a position of a bird in the search space, called these birds "particle". Each particle has its own position and speed (to determine the direction and distance of flight), and each particle's performance depends on the optimization objective function to determine the fitness value. Each particle memorizes and follows the optimal particle current to search in the solution space. At the same time each iterative process is not completely random, if find a better solution, according to this solution to find the next solution.

In a D dimension target search space, M particles form a group, In the T iteration, P_i particle position vector can be expressed as: $X_i(t) = (x_{i1}, x_{i2}, ..., x_{in})$, the velocity vector can be expressed as: $V_i(t) = (v_{i1}, v_{i2}, ..., v_{in})$. In each iterative process, the particle is updated by following two "extreme", The first is the best solution, called individual extreme points, expressed as $pbest_i = (pbest_{i1}, pbest_{i2}, ..., pbest_{in})$;Another extreme point is the best solution found in the whole population, known as the global minimum points, expressed as $gbest_i = (gbest_{i1}, gbest_{i2}, ..., gbest_{in})$. When the calculated at the T + 1 iteration, the particle P_i to update the velocity and position according to the following equation:

$$v_{id}(t+1) = wv_{id}(t) + c_1 rand_1(pbest_{id} - x_{id}(t)) + c_2 rand_2(pbest_{id} - x_{id}(t)) \qquad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), d = 1,2,..., D \qquad (2)$$

Among them, w is the inertia weight; c_1 and c_2 are two learning factor, usually taken to be 2; $rand_1()$ and

$rand_2()$ is two random numbers usually taken to be [0,1].

To prevent the particles from the search space, particle's one-dimensional velocity will be limited in [-vdmax, vdmax], if vd > vdmax then vd = vdmax,or vd < -vdmax then vd = -vdmax .

III. NONLINEAR INERTIA WEIGHTS DECREASING RANDOM CHANGED PARTICAL SWARM OPTIMIZATION

In the process of the standard particle swarm optimization algorithm to update the velocity and position, the inertia weight is the most important parameter, which is the convergence effectively control algorithm's convergence and exploration capacity. The inertia weight determines the size of the particles on the rate of inheritance; Greater weight will make the particles with larger speed, In order to improve the algorithm's global search ability; while smaller weights will enhance the ability of local search algorithm, which is beneficial to the control algorithm convergence. The inertia weight selection largely determines the algorithm execution effect. In recent years, scholars are very keen on the research of inertia weight parameter value setting[9~12], and put forward a variety of change program. This paper adopts to achieve the same generation of particle inertia weight difference strategy based on nonlinear decreasing weight. Compared with the single linear decreasing inertia weight original way, the improved particle swarm optimization algorithm can flexibly adjust the global search and local search ability in the whole process of implementation of the algorithm.

A. Nonlinear Inertia Weights Decreasing Random Change Strategy

The formula (1), Shi[4] proposed the linear decreasing inertia weight strategy LDW(Linearly Decreasing Inertia Weight , LDW for short.) .That is:

$$w = w_{\max} - (w_{\max} - w_{\min})(t / t_{\max}) \qquad (3)$$

Among them, $w_{\max}$ is the largest of the inertia weight, $w_{\min}$ is the least inertia weight; $t_{\max}$ indicating the number of iterations the maximum allowed; T indicates the number of iterations. LDW method is simple, intuitive, and compared to constant weight has better optimization performance, so it is widely used in engineering optimization problem. This strategy is inspired, this paper selected nonlinear decreasing inertia weight is:

$$\begin{aligned} w_t &= (w_{\max} - w_{\min})(t / t_{\max})^2 + \\ &\quad 2(w_{\max} - w_{\min})(t / t_{\max}) + w_{\max} \end{aligned} \qquad (4)$$

This is an open up parabola. Let $w_{\max}$ =0.9, $w_{\min}$ =0.1. In formula (4), if t=0 then w= $w_{\max}$;if t= $t_{\max}$ then the value of w reduced to $w_{\min}$ along the concave nonlinear function.

Using formula (4), calculated for each generation of inertia weight $w_t, t = 0,1,2...,t_{\max}$,triangular probability distribution is as the T generation in each particle random inertia weight, which is broken line taking $(w_{\max},0),(w_{\min},0),\left(w_i,\frac{2}{w_{\max} - w_{\min}}\right)$ as a vertex. The T generation, each particle randomly selected inertia weight formula is as follows:

$$x_i = G_I(u_i), i = 1,2,...M \qquad (5)$$

Among them, $u_1, u_2, ..., u_M$ are random number selected from random variable obeying u uniform distribution on [0,1]. Formula (5), $\mathrm{G}_0(\mathrm{u})(\mathrm{t} = 0), G_t(u)(t > 0)$ respectively:

$$G_0(u) = w_{\min} + (w_{\max} - w_{\min})\sqrt{u}, 0 \le u \le 1 \qquad (6)$$

$$G_t(u) = \begin{cases} w_{\min} + \sqrt{(w_{\max} - w_{\min})(w_t - w_{\min})u}, \\ \qquad 0 \le u \le \dfrac{w_t - w_{\min}}{w_{\max} - w_{\min}} \\ w_{\max} + \sqrt{(w_{\max} - w_{\min})(w_t - w_{\min})(u-1)}, \\ \qquad \dfrac{w_t - w_{\min}}{w_{\max} - w_{\min}} \le u \le 1 \end{cases} \qquad (7)$$

B. Inertia Weight Nonlinear Decreasing Random Change the Process of Particle Swarm Optimization

Inertia weight nonlinear decreasing random change the process of particle swarm optimization is as follows:

(1)Initialization M particles position and speed in the swarm (usually randomly generated in the allowable range).

(2)According to the fitness function, calculates particle's adaptive value. For each initial particle, the particle $\mathrm{P_i}$ is set for all extreme points of $pbest_i$. In all $pbest_i (i = 1,2,...,M)$, the particle to make the fitness function value reaching the maximum is called the global extremum point $gbest$.

(3)Determine the termination conditions (the conditions for the termination of the iteration is Maximum number of iterations or meeting the preset threshold condition given), if meet, steering (7); otherwise, executive (4).

(4) By the formula (4) to calculate w_i ; randomly generates M random numbers in [0, 1] interval. Into the formula (6) (when $t = 0$) or formula (7) (when $t > 1$), then can calculate the inertia weight of each particle in this condition; According to the formula (1), (2) to update particle $\mathrm{P_i}$'s velocity vector and the position vector .

(5) According to the fitness function, fitness is calculated for each of the new particle, if the new particle's fitness is better than the current particle individual extreme, the new particle set to $pbest_i$. If the global extremum current is better than the current global extremum, the individual extremum point is set to $gbest$.

(6) Determine the termination conditions, if meet, steering (7); otherwise, executive (4).

(7)The optimal $gbest$ finally gets is the required solution.

C. The Validity Test of Improved Particle Swarm Optimization

In order to test the effectiveness of the improved algorithm, in this paper, five Classic benchmark functions that domestic and foreign scholars often use to evaluate the performance of the algorithm optimization, and compared with other particle swarm algorithm. The parameter settings are as follows:

Population size $M = 30, c_1 = c_2 = 2$, the inertia weight of the basic PSO for $w = 0.75$; the inertia weight LDWPSO uses the formula (8) to set, and the inertia weight NLDWPSO uses the formula (9) to set. Among them, $w_{\max} = 0.9, w_{\min} = 0.1$, for the NLDRWPSO, $w_{\max} = 0.9, w_{\min} = 0.1$.

Four algorithms of the continuous operation 20 times, the maximum number of iterations are 1000. For each function, were given error threshold of 10^{-8}; when the objective function value of the best particle population and the value in the known extreme distance is less than the error threshold, or iterative execution to the maximum algebra, so the algorithm is terminated. Performance comparison of four algorithms lists in table I.

TABLE I. COMPARATIVE PERFORMANCE IN 10 DIMENSIONS OF STANDARD TEST FUNCTIONS OF THE FOUR ALGORITHMS

Function	**Algorithm**	**Optimal value**	**The average**	**The average number of iterations**	**Run time （ms）**	**The convergence rate（%）**
Sphere	PSO	- 4. 955 098E - 11	- 4. 836 299E - 09	69	0. 585 417	100
	LDWPSO	- 4. 857 637E - 10	- 5. 296 163E - 09	135	1. 034 117	100
	NLDWPSO	- 7. 397 043E - 10	- 4. 713 737E - 09	102	0. 763 283	100
	NLDRWPSO	- 9. 514 736E - 10	- 5. 534 827E - 09	38	0. 370 567	100
Rosenbrock	PSO	- 5. 782 793E - 10	- 5. 247 432E - 09	95	0. 743	100
	LDWPSO	- 1. 978 520E - 10	- 5. 406 032E - 09	166	1. 122	100
	NLDWPSO	- 6. 550 292E - 10	- 4. 730 213E - 09	133	0. 752	100
	NLDRWPSO	- 7. 193 305E - 11	- 4. 169 699E - 09	53. 9	0. 549	100
Rastrigin	PSO	- 8. 155 130E - 10	- 4. 692 926E - 09	96	0. 57	100
	LDWPSO	- 3. 296 832E - 10	- 5. 701 939E - 09	176	1. 005	100
	NLDWPSO	- 6. 156 855E - 11	- 4. 404 017E - 09	129	1. 21	100
	NLDRWPSO	- 1. 421 628E - 10	- 3. 443 026E - 09	53	0. 547	100
Griewank	PSO	- 1. 144 633E - 09	- 4. 807 921E - 03	721	5. 509	40
	LDWPSO	- 6. 726 752E - 11	- 2. 218 815E - 03	405	2. 923	70
	NLDWPSO	- 2. 249 092E - 09	- 4. 068 318E - 03	571	3. 179	50
	NLDRWPSO	- 8. 429 112E - 10	- 7. 396 041E - 03	792	5. 728	64
Schaffer′ f 6	PSO	1. 000 000E + 00	9. 946 562E - 01	707	4. 791	100
	LDWPSO	1. 000 000E + 00	9. 927 131E - 01	793	4. 421	100
	NLDWPSO	1. 000 000E + 00	9. 951 422E - 01	600	3. 368	100
	NLDRWPSO	1. 000 000E + 00	9. 951 428E - 01	580	3. 212	100

The experimental results can be obtained based on the above comparison: Changed the algorithm in the *Sphere* function in the accuracy of less than PSO, the average number of iterations, the calculation speed, convergence effects are very well behaved; especially the optimization effect in *Rosenbrock*, *Rastrigin* function in the algorithm is especially prominent; However, the improved algorithm optimization performance slightly in *Griewank* function, and the performance of the LDWPSO algorithm is more prominent, high efficiency; finally, the performance of two kinds of nonlinear algorithm in $Schaffer' f6$ function are good, the difference is not much，the improved algorithm has better performance. Comprehensive consideration, the average performance of NLDRWPOPS algorithm is the best.

REFERENCES

[1] Jichuan Wang, Zhigang Guo. Logistic Regression model--Methods and Applications [M] . Beijing: Higher Education Press, 2001.

[2] Yuguang Chen, Shan Li, Shili Liao. Multiple regression model of the engine control based on Genetic Algorithm [J] . Agricultural Machinery, 2006, 37 (6):9212.

[3] Qi,Shi , Weimin Shen, Dongxun Jiao. QSAR analysis algorithms for the toxicity of aromatic compounds improved particle swarm optimization [J]. Henan University of Science and Technology,2008 ,29 (6) :96299.

[4] Xinjie Wu, Yuan Li, Dan Zhan. Application of multiple linear

regression analysis of the problem of PSO algorithm [J] . Journal of Liaoning University, 2007, 34 (1):18220.

[5] Qian Huo, Shuquan Li, Wenyuan Wang. The research of Genetic algorithm applied to multi-parameter nonlinear regression models seeking [J]. Journal of Agricultural University of Hebei, 2002, 25 (2): 1072110.

[6] Kennedy J, Eberhart R. Particle Swarm Optimization[C] ∥Proc of the IEEE Conf on Neural Networks , 1995 : 194221948.

[7] Eberhart R , Shi Y. Comparison between Genetic Algorit hmsand Particle Swarm Optimization[C] ∥Proc of t he 7th Int’l Conf on Evolutionary Computation , 1998 :6112616.

[8] Shi Y, Eberhart R. A Modified Particle Swarm Optimization[C] ∥Proc of the IEEE Int’1 Conf of Evolutionary Computa2tion , 1998 :69273.

[9] Dingxue Zhang, Zhihong Guan, Xinzhi Liu. A dynamic inertia weight adaptive particle swarm optimization [J] . Control and Decision, 2008, 23 (11):125321257.

[10] Jianhua Liu, Xiaoping Fan, Zhihua Zhai .A dynamic inertia weight adjustment novel particle swarm optimization [J]. Computer Engineering and Applications,2007 ,43 (7) :68270.

[11] Chongpeng Huang, Weili Xiong, Baoguo Xu. The impact and improvement of Inertia weight PSO algorithm convergence [J]. Computer Engineering, 2008, 34 (12):31233.

[12] Wenzhong Guo, Guolong Chen. A new strategy of inertia weight adjustment of Particle swarm optimization algorithm [J]. Computer engineering and Science, 2007, 29 (1):71275.