ISSN : 0974 - 7435 Volume 10 Issue 18

2014

BioTechnology

An Indian Journal

FULL PAPER

BTAIJ, 10(18), 2014 [10365-10371]

A novel algorithm for analyzing influence factors of sports industrialization

Liu Qin[1]*, Qin Wei[2]

[1]Department of Physical Education, Shanghai University of Electric Power, Shanghai 200090, (CHINA)

[2]Department of Physical Education, Donghua University, Shanghai 201620, (CHINA)

E-mail : 407706483@qq.com

ABSTRACT

The researches on improved types of particle swarm optimization (PSO) algorithm to solve all kinds of optimization problems in real life have become hot topics. The paper presents a new cloud particle swarm optimization algorithm and applies it to analyze influence factors of sports industrialization. First, in order to present targeted improvement measures, the working principle and the root cause of the defects of PSO algorithm is analyzed; Second, the improvements such as global optimization searching method and crossover operation of cloud particle are taken, and then a new cloud particle swarm optimization algorithm is presented. Finally, the improved algorithm is applied to analyze influence factors of sports industrialization and the experimental results show that the improved algorithm has better convergence rate, optimize performance and better accuracy.

KEYWORDS

Particle swarm optimization; Global optimization searching; Crossover operation; Influence factor analysis; Sports industrialization.

INTRODUCTION

Particle Swarm Optimization (PSO) algorithm is a new type algorithm which based on the calculation of the evolution iterated. It is proposed by American Social Psychology Kennedy and Electrical Engineer Eberhart in 1995, the basic idea comes from their early studies of the behavior of birds group, the PSO algorithm has the bigger advantage because its concept is simple, easy to implemented, and has a good convergence rate, more and more scholar are beginning to taking Concern and research on it in recent years, Algorithm is Effective optimization tools on the problem of Nonlinear Continuous Optimization, Combinatorial optimization, Mixed-integer nonlinear optimization, nowadays, it has been widely used in function optimization, the Area of Neural Network Training, Fuzzy System Control, GA, the application In signal processing, pattern recognition, robot activity planning, system design, decision-making, job scheduling, image segmentation, time-frequency analysis of such issues also have been reported[1,2].

But in practical application, PSO algorithm has some defects which limits the real application of the algorithm. On one hand, because the weight ω plays an important role in iterative equation in PSO algorithm, the greater ω is helpful to the speed of convergence, global search, but difficult to obtain accurate solution, the smaller weight ω is helpful to the local search[3,4], it make easy to obtain accurate solution but the convergence is slow. Now the methods widely used currently is to make the weight linearly decrease by the iterations. In order to overcome above defects, the paper presents a new Cloud Particle Swarm Optimization (CPSO) algorithm.

MATERIALS AND METHODS

Working principle of particle swarm optimization algorithm

Particle Swarm Optimization algorithm, based on swarm intelligence, is an evolutionary calculation method simulating flying behavior of birds. The working principle of PSO is the potential solution of each optimization problem like a bird in search space which is called “particle”. Each particle in the algorithm has an adaptive value which is determined by optimization function, and a speed vector which is determined by their flying distance and direction; All the particles should follow the search of existing optimal particle in solution space. PSO algorithm is started as a number of random particles, and then through iteration to find optimal solution for each particle. In every iteration, Each particle should update its optimal solution by follow and optimize the two extreme values, in which one extreme value is the best optimal solution which help other particles to finds their current moment which is called individual best value. The other best solution is for the entire group to find their current moment, called best global value[1,5].

Suppose that a group is formed by m particles forming in D-dimensional target search space, in the group the ith particle can be expressed as a D-dimensional vector and expressed as $\vec{x_i} = (x_{i1}, x_{i2}, ...x_{id})$ [6]. In D-dimensional, the location of the ith particle search space is a potential solution[7]. We can get its adaptive value by substituting $\vec{x_i}$ into a target function, measuring the weakness and strength of $\vec{x_i}$ according to the size of its adaptive value. The flying speed of the ith particle can be expressed as a D-dimensional vector also, expressed as $\vec{v} = (v_{i1}, x_{i2}, ...v_{id})$, and $\vec{p_{id}} = (p_{i1}, p_{i2}, ...p_{id})$ can be used to express the optimal location searched till now of the ith particle, and $\vec{p_{gd}} = (p_{g1}, p_{g2}, ...p_{gd})$ is used to express the optimal location searched till now of the entire particle swarm. Original PSO algorithm carries out the calculation of each particle by equation 1 and equation 2 followed[8].

$$v_{id}^{t+1} = \varpi v_{id}^{t} + c_1 r_1 (p_{id} - x_{id}^{t}) + c_2 r_2 (p_{gd} - x_{id}^{t}) \quad \textbf{(1)}$$

$$x_{id}^{t+1} = x_{id}^{t} + \alpha v_{id} \quad \textbf{(2)}$$

In equation 1 and equation 2, $i = 1,2,3......m, d = 1,2,3......D$, c_1 means learning factors c_1 and c_2 means nonnegative constants respectively; r_1 and r_2 means random numbers and their values should follow in the uniform distribution of [0, 1][9]. In the equation 1, ϖ is called momentum coefficient and means nonnegative number, used for controlling the influence of previous speed on current speed; Generally speaking, the greater ϖ means the stronger global searching ability and the bigger influence of previous speed; and vice versa, the smaller ϖ means the weaker global searching ability and the less influence of previous speed. So the local minimum value can be able to be jumped out by adjusting the size of ϖ. The variable α is use and called constraints factor to control the weight of speed. According to specific problems, the condition for terminating iteration is selected as optimal location particle swarm or maximum iterations searched up until the desired minimum adaptive threshold is met[10].

Cloud particle swarm optimization algorithm design

Cloud PSO algorithm adopts updating form of basic PSO speed and position to carry out global optimization searching. However, after searching, the combination of cloud model and particles makes the particles obtain randomness and stabilization tendency, but part of the particles fail to carry out global searching within the whole searching scope due to losing the diversity of population. Hence, the thesis adopts cloud self-adaptive inertia weight for adjustment.

ICPSO (improved cloud particle swarm optimization) guarantees the effectiveness of particles in later searching phase. The improvement measures make use of the crossover operator of cloud genetic algorithm to carry out crossover operation on the particles of cloud PSO, then making the particles after crossover carry out chaos immigration searching, thus making the part of particles falling into local extreme value achieve global best, improving the algorithm optimization accuracy.

In the basic PSO, assuming there are N particles forming a population in a D-dimensional target searching domain[6], X_i indicates the position of the ith particle, $X_i = (x_{i1}, x_{i2}, ...x_{id})$, P_i indicates the historical optimal position of the ith particle, P_g indicates the optimal position searched by the entire particle up till now, $P_i = (p_{i1}, p_{i2}, ...p_{id})$, $P_g = (p_{g1}, p_{g2}, ...p_{gd})$, V_i indicates the flight speed of the ith particle, $V_i = (v_{i1}, v_{i2}, ...v_{id})$, particles adjust their position and speed according to equation 3 and equation 4.

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{gd}^{k}) \tag{3}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{4}$$

Speed adjustment rules are shown in equation 5.

$$\begin{cases} v_{id} = v_{\max}, & v_{id} \succ v_{\max} \\ v_{id} = -v_{\max}, & v_{id} \le v_{\max} \end{cases} \tag{5}$$

In the basic PSO algorithm, the weight is calculated according to iterative formula; as the larger inertia weight ω helps jumping out of local optimization, for global optimization searching, the smaller inertia weight ω helps local optimization, accelerating algorithm convergence, thus clustering particle swarm via evolutionary strategy of different weights[8].

Assuming the size of particle swarm as N, in the t times of iteration, the fitness value of particle X_i is f_i, and the average fitness of particles is equation 6.

$$f_{avg} = \frac{1}{N} \sum_{i=1}^{N} f_i \tag{6}$$

Averaging the fitness value that fitness value is superior to f_{avg} to obtain f'_{avg}, averaging the fitness value that fitness value is inferior to f_{avg} to obtain f''_{avg}, the fitness value of optimal particle is $f_{\min}$, dividing the group into 3 subgroups, adopting different weights to generate strategy which is: (1) if f_i is superior to f'_{avg}, the fitness value of particles is smaller, closer to optimal solution, thus adopting smaller inertia weight, evolutionary strategy adopting "social model", accelerating global convergence speed, ω is valued as 0.2; (2) if f_i is inferior to f''_{avg}, and the fitness value of particles is larger, evolutionary strategy of optimal solution adopts "cognitive model", making the particles with poor performance accelerate the rate of convergence, further, adopting larger inertia weight, ω is valued as 0.9; (3) if f_i is superior to f''_{avg} but inferior to f'_{avg}, the fitness value of particles at this time is moderate, inertia weight adopting cloud self-adaptive inertia weight, evolutionary strategy adopting "complete model".

The generating algorithm of cloud self-adaptive inertia weight ω adopts equations 7-11 successively.

$$Ex = f_{avg} \tag{7}$$

$$En = (f'_{avg} - f_{\min}) / c_1 \tag{8}$$

$$He = En / c_2 \tag{9}$$

$$En' = normrnd(En, He) \tag{10}$$

$$\omega = 0.9 \sim 0.5 \times e^{\frac{-(f_i - Ex)^2}{2(En')^2}} \tag{11}$$

With the decrease of the fitness of cloud particle, we can know from limit theorem, $0 \prec \frac{-(f_i - Ex)^2}{2(En')^2} \prec 1$, thus making $\omega \in [0.4, 0.9]$; we can know from the value formula of ω, ω will be decreased with the decrease of fitness value of cloud particle, thus realizing optimal particle obtaining smaller value of ω.

Crossover operation of cloud particle

Under the condition of large scale of population, while solving multimodal function, as cloud particle algorithm, on the condition of searching large scale of population, in later phase of searching, is easy to fall into local optimization, thus failing to find global optimal solution, the reason of which is that this part of particles have lost the diversity of population, crossover operator in cloud genetic algorithm is adopted to carry out particle crossover operation for such part of particles losing diversity, for the purpose of generating new offspring particles to substitute the position of old particles.

The main idea of crossover is to introduce the crossover operation in cloud genetic algorithm in the process of basic cloud particle swarm iteration; users define a crossover probability to give it to the cloud particle, and the probability is irrelevant to the fitness value of cloud particle. In the process of iteration, only looking for crossover probability, choosing a certain quantity of particles to put them into a particle pool; these cloud particles crossover with each other randomly and generate filial generation, with quantity unchanged, substituting parent particles with filial generation particles to guarantee the unchanged particles quantity of population.

After the crossover cloud particles complete initialization, update the position and speed of particles according to Formula 1 and Formula 3; the crossover strategy is: (1) sorting all the cloud particles according to high and low fitness in each iteration process, making half of particles with high fitness to directly enter the next generation, and the left half of particles enter crossover-waiting area; (2) randomly generating crossover position $C_i \in [1, D]$, carrying out crossover on the $C_i \sim D$ position of the ith particle and the $N \times 3/2 + 1 - ith$ particle; while crossover, choosing certain crossover probability, and the crossover probability in the cloud genetic algorithm is adopted here.

Use a and b to indicate the pointer of the chosen two parental individuals; calculation formula using cloud genetic operator crossover is shown in equation 12.

$$\begin{cases} Child_1(X_j = p_j \cdot parent_1(X_j) + (1.0 - p_j) \cdot parent_2(X_j) \\ Child_2(X_j = p_j \cdot parent_2(X_j) + (1.0 - p_j) \cdot parent_1(X_j) \\ Child_1(V) = \dfrac{parent_1(V) + parent_2(V)}{|parent_1(V) + parent_2(V)|} |parent_1(V)| \\ Child_2(V) = \dfrac{parent_1(V) + parent_2(V)}{|parent_1(V) + parent_2(V)|} \end{cases} \tag{12}$$

After calculating the crossover operation of cloud particles according to Formula 10, the offspring cloud particles generated obtain and inherit the advantages of their parental cloud particles; via population processing, theoretically, the regional searching ability among cloud particles is greatly improved. For example, parental cloud particles are located in the different local optimization region, after carrying out crossover on such two cloud particles and generating offspring cloud particles, thus obtaining superior searching result.

Building chaos immigration

In the theory of biological evolution, one of the effective methods for maintaining population diversity is immigration. Bring in certain scale of excellent individuals of the same species from outside the population to substitute bad ones in the original population and participate in the mating and reproduction of the population to guarantee the quality of the population and avoid genetic lesions and recession due to inbreeding.

As the features of normal cloud model are randomness and stabilization tendency. Randomness can keep the diversity of individuals, thus avoiding searching falling into local extremum; and stabilization tendency can well protect superior individuals, thus carrying out self-adaptive positioning for global extremum.

Single cloud genetic algorithm is low in calculation efficiency while reactive power optimization and slow in rate of convergence, thus bringing in chaos immigration. The specific operation is to carry out chaos immigration searching on part of individuals after the population completes one evolution (choosing, crossover, mutation) to obtain optimal individual with larger fitness value, leading population evolution. As the point near optimal point is lower, 3 individuals whose fitness value closer to optimal individual participate in chaos immigration searching.

Chaos is the nondeterministic and unpredictable motion state obtained by certain linear equation; variable appearing chaos state in nonlinear equation is called chaotic variable. Mapping the potential solution of the function to be optimized into chaotic channel; through chaotic iteration, we can make optimization searching process have the ability to jump out of local optimal solution. Such searching mechanism is called chaotic local searching.

Chaotic motion has ergodicity, randomness and regularity, which can traverse all the states without repetition according to self rule within certain range. Compared with other power systems generating chaotic change, Logistic mapping is simple and small in calculated quantity, thus mapping is as shown in equation 13.

$$x_{k+1} = \mu x_k (1 - x_k) \qquad 0 \le x_0 \le 1 \tag{13}$$

In which, μ is controls parameter, x_k is variable, $k = 0,1,2,\ldots$. When $\mu = 4$, the sequence generated by Logistic mapping appears chaotic state feature.

To sum up, chaos is the inherent randomness possessed by certain nonlinear system, which has the following characteristics: (1) extremely sensitive to initial conditions; in the long run, the future behavior of chaotic system is unpredictable; (2) chaos shows strange attractor in state space, having the self-similar structure of infinite nest in the finite phase space geometry; (3) the frequency spectrum of chaotic system is continuous, similar to that of noise, having continuous broadband; (4) Lyapunov index of chaotic system has one number larger than 0 at least.

While Cloud PSO carries out reactive power optimization, later phase of iteration generates “sluggishness” motion, making each cloud particle lost population diversity, so each control variable obtained by the reactive power optimization of the algorithm is easy to fall into local solution region, therefore, in the later phase of searching, cloud particles adopt chaos immigration operation, making cloud particles with small population diversity able to participate in reactive power optimization.

The specific steps for building immigration population

(1) initialization of chaotic variable. Iteration times $k = 0$, carry out linear transformation on the value of each variable of 3 individuals after cloud genetic iteration, mapping decision variable x_j^k according to Formula 11, making each variable of each individual become the value among (0,1), as the initial value of chaotic variable, current solution recorded as $x_c = (x_{10}, x_{20}, \ldots, x_{n0})$, cx_j^k is calculated according to equation 14.

$$cx_j^k = \frac{x_j^k - x_{\min,j}}{x_j^k - x_{\max,j}} \qquad j = ,1,2,\ldots,n \tag{14}$$

(2) Generate n (0,1) distributed chaotic variable new sequence λ_j with Formula 11, $j = ,1,2,\ldots,n$, n is the number of control variable, setting chaos searching times as N, generation kept by optimal individual in chaotic searching being N_1;

(3) transforming chaotic variable to allowable solution space $cx_j^k = a_j + \lambda_j^k (b_j - a_j)$, in which a_j and b_j are the upper and lower limits of corresponding control variables, and calculating the value of current target function $f(x^k)$;

(4) if $f(x^k) \prec f^*$, keep existing optimal solution, current solution and corresponding optimal target function value, f^* being existing optimal value;

(5) if f^* keeps unchanged after the searching of N_1 step, finish chaotic searching and return to cloud genetic iteration; otherwise, continue to search until meet the times of chaotic searching.

After adopting chaos immigration operator, make cloud particle swarm be able to keep good population diversity within searching domain, and let more particles be able to participate in searching motion.

RESULT AND DISSCUSS

The influence factors of sports industrialization

Taking successful experience in the influence factor analysis of sports industrialization at home and abroad as reference[11,12], also in consideration of the specialist consultation, the paper divides the influence factors of sports industrialization into three types, that are institutional factors, demand factors, supply factors. In which, institutional factors means resource supply condition of sports industrialization and includes funding, human resource, sports products and service and relative price advantage conditions; Demand factors means demand for sports products and service quantity and demand structure and income level, demand preference, population, consumer expectations, marketing and cultural fashion; Supply factors means the driving force of sports industrialization and includes capital investment and talents with high quality.

Experimental results and analysis

The paper takes the data of the sports industrialization in Shanghai and Beijing for experimental example. Relevant data of three type typical factors of each city are selected as the basis for data training and experimental verification in the paper.

Limited to paper space, the analysis of intermediate results is omitted here, only providing parts analysis results, see TABLE 1.

TABLE 1 : Effect strength of different factors of sports industrialization

Factors	Effect strength	Factors	Effect strength
Institutional factors	0.320	Funding,	0.132
		Human resource	0.086
		Sports products and service	0.050
		Relative price advantage	0.052
Demand factors	0.490	Income level	0.092
		Demand preference	0.045
		Population	0.075
		Consumer expectations	0.133
		Marketing	0.091
		Cultural fashion	0.054
Supply factors	0.190	Capital investment	0.123
		Talents with high quality	0.067

In order to illustrate feasibility and practicability of the improved algorithm and original PSO algorithm[4] and BP neural network algorithm[12] are realized in the same calculation platform in the paper. The calculation platform: Windows 8,ThinkPad, Intel i5-3320M, 2.6GHz, DDR3 1600,4GB DDR3. The realization of different algorithms is shown in TABLE 2.

TABLE 2 : Realization results of different algorithms

	Algorithm in the paper	Ordinary PSO model	BP neural network
Calculation Accuracy	93.78%	72.44%	87.87%
Time Consuming (S)	11	251	874

CONCLUSIONS

The researches on improved types of particle swarm optimization (PSO) to solve all kinds of optimization problems in real life have become hot topics. The paper improves PSO algorithm based on cloud particle calculation. The experimental results show the improved PSO algorithm has the better convergence rate, optimize performance and has the better accuracy when use to analyzing the effect strength of the factors of sports industrialization. In the next study, we should pay our focus on the application universality of the improved algorithm.

ACKNOWLEDGEMENT

This work is supported by the scientific research (innovation) of Shanghai Education Committee under the grant No.12YS113.

REFERENCES

[1] Zhou Huan; Study of Analyzing Influence Factors of Sports Industrialization with PSO Model. Computer Engineering and Design, **29(3)**, 659-664 **(2013)**.

[2] Deng Weibing, Wang Yan; B2C Customer Classification Algorithm Based on Improved Particle Swarm Optimization Algorithm. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), **21(4)**, 568-572 **(2009)**.

[3] Coello Polido; Application of Improved PSO Algorithm in Telecom Customer Segmentation. Computer Simulation, **28(3)**, 138-143 **(2012)**.

[4] Fink Sietsma, Platt Rumelhart; Application of Particle Swarm Optimization Algorithm Based on Commercial Bank Customer Subdivision. Computer Simulation, **28(6)**, 357-362 **(2012)**.

[5] Yang Benzhao, Tian Den, Zhang Tiangen; Research on Improvement of Particle Swarm Optimization Algorithm Based on BP Neural Network Model. Calculation Science and Technology Research, **23(12)**, 68-72 **(2013)**.

[6] Post Bradley, Managasarian Lucent; Research on the Improvement and Application of Particle Swarm Optimization algorithm. Journal of Global Optimization, **16(1)**, 23-32 **(2014)**.

[7] Tang Yong, Rong Qiusheng, Huang Shengyi; An Implementation of Clustering Algorithm Based on Particle Swarm Optimization Algorithm. Journal of Hubei Institute for Nationalities, **22(1)**, 69-74 **(2014)**.

[8] Reynolds Lee, Lechuga Fahoo; An Improved Particle Swarm Optimization Algorithm. Computer Application, **33(8)**, 31-35 **(2013)**.

[9] Wang Zhigang, Li Xueer, Zhang Jianguo; A Modified Particle Swarm Optimization. Journal of Harbin University of Comm erce (Natural Sciences Edition), **25(4)**, 454-458 **(2009)**.

[10] Bhupesh Gour, Ravindra Patel; Improved Particle Swarm Optimization Algorithm and Its Simulation. Process Automation instrumentation, **30(7)**, 297-304 **(2012)**.

[11] Li Jing, Wu Cheng, Huang Gungying; Research on Analyzing Influence Factors of Sports Industrialization. Science Technology and Engineering, **9(6)**, 231-235 **(2011)**.

[12] Amandel John, Mathews Hart, Platt Lebiere; Study on Analyzing Influence Factors of Sports Industrialization Based on BP Algorithm. Journal of Conputer Research and Devolopment, **17(11)**, 761-769 **(2013)**.