Dynamic multi-species coevolution large-scale optimize based on the fuzzy clustering and trust region

Xuhui Zhang[1, 2*]

[1]*Department of Mechanical Engineering, Xi'an University of Science and Technology, No.58 Yatan Road, Xi'an 710054, China*

[2]*Xi'an Coal Mining Machinery Co., Ltd., Xi'an 710032, China*

Received 1 July 2014, www.cmnt.lv

Abstract

Based on the above-mentioned studies, this article put the modified Fuzzy Clustering method into the particle swarm optimization, which solved the curse of dimensionality existing in the conventional algorithms. The large-scale parameter optimization method applied the modified Fuzzy C-Means method to clustering the large-scale dimension under the coevolution frame and achieved the valid dimension grouping. Later on, the dynamic neighbourhood topology multi-species particle algorithms divided the whole species into packets and constructed the subspecies sharing neighbourhood information, which improve the searching efficiency. The bring-in trust region could have the self-adapt adjustment for the particle optimization range, accelerate the optimizing speed, and decrease the iterations in the dead space. We use 20 standard large-scale testing functions for simulation. Compared with the top-ranked tournament algorithm, the mentioned algorithm achieved a better optimizing result in most functions, which surely laded the foundation of the large-scale neural network parameter optimization and the application in the control system.

Keywords: multi-species coevolution, particle swarm optimization, guzzy clustering

1 Introduction

Particle Swarm Optimization doesn't rely on the specific objective function form and avoided the counting process of the Jacobi matrix and Hessian matrix in the classical mathematics methods. But its search capability would heavily decline facing the large-scale problem. The phenomenon was not only caused by the existence of the locally optimal solution, but also by the possible degradation of the particle's speed, which lead to the search in a continuous subspace of the whole searching hyperplane. Besides, there are many complicated high-dimensional optimization problems in the actual industrial production process. Therefore, it's highly necessary to find out a fast and efficient algorithm to solve the large-scale optimization problem. This article put forward the Dynamic Neighborhood Particle Swarm Optimization Algorithm with the Fuzzy Clustering, and mainly improved the three things as follows:

Adjust searching space in the self-adapt way and accelerate the optimization process. When the next iterative point could fully decrease the objective function, the trust region (Independent variable search space) would diminish and the particle could rapidly find out the optimal point, without wasting more iteration in the dead space.

The most important problem in the coevolution is the packets. How many dimensions should be there in one packet? And which dimensions should appear in the same packet? Hence, using the improved Fuzzy C-Means algorithm suggested in this article to analyse the relationship between variables and divide the similar variables into one packet for optimizing.

Adopting the Dynamic Neighbourhood Particle Swarm Optimization Algorithm, this article introduced an optimize method in packets. The self-Adapting topological structure could learn the neighborhood position and self-organize and construct the subpopulation to share the information, which improve the continuous space search ability

2 The particle large-scale parameters optimization method based on the Fuzzy C-Means Clustering

2.1 FCM CLUSTERING ALGORITHM

Traditional clustering algorithms assign each object to one and only one cluster in accordance with strict standards. The clusters have distinct boundaries. Objects in a cluster are very dissimilar from objects in the other clusters. This is called hard clustering. Traditional clustering algorithms are all based on *"hard partition"* [1]. However, things in the objective world are closely connected to each other; nothing can exist in total isolation. In other words, there is not a thing that belongs to one and only one class; objects cannot be divided into classes that share no similarities. People have a better understanding of the world as it develops. But traditional hard clustering methods fail to solve practical problems well. *"Soft partition"*, a more realistic clustering method, was thus created in contrast to *"hard partition"*.

[*]***Corresponding author*** e-mail: zhangxh@xust.edu.cn

Fuzzy C-Means Clustering Algorithm (FCM clustering algorithm) is a fuzzy clustering algorithm based on objective functions, which is different from traditional clustering algorithms. FCM clustering algorithm, in essence, is rooted in Hard-means Algorithm (HCM) [2].

2.2 SOLVING PROCESS OF FCM CLUSTERING ALGORITHM

Based on the principle of least squares, FCM clustering algorithm aims to minimize its objective function $J(\mu, A)$ and realize convergence of the iteration process. The objective function $J(\mu, A)$ of FCM is defined as:

$$J(\mu, A) = \sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}\left\|X_j - A_i\right\|^2, \tag{1}$$

where $A_i (i = 1, 2, ..., c)$ is the cluster centre and m (m>1) is the fuzziness index. Below are the value formulas of μ and A during convergence of the iteration process:

$$\mu_{ij} = \begin{cases} \left(\sum_{k=1}^{c}\frac{\left\|X_j - A_i\right\|^{2/(m-1)}}{\left\|X_j - A_k\right\|^{2/(m-1)}}\right)^{-1}, & \left\|X_j - A_i\right\| \neq 0; \\ 1, & \left\|X_j - A_k\right\| = 0 (k = j); \\ 0, & \left\|X_j - A_k\right\| = 0 (k \neq j), \end{cases} \tag{2}$$

$$A_i = \sum_{j=1}^{n}\mu_{ij}^{m}X_j \Big/ \sum_{j=1}^{n}\mu_{ij}^{m}. \tag{3}$$

Suppose the sample space $X = (x_1, x_2, ..., x_m)$ and c denotes all positive integers greater than 1. A fuzzy matrix $\mu = (\mu_{ij})$ is used to divide Z into c classes. μ_{ij} means the i^{th} sample point belongs to the j-th membership. Obviously, μ_{ij} meets the following conditions:

$$\begin{aligned} &\mu_{ij} \in [0,1], \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., c \\ &\sum_{j=1}^{c}\mu_{ij} = 1, \forall i, \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., c. \\ &\sum_{i=1}^{n}\mu_{ij} > 1, \forall j, \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., c \end{aligned} \tag{4}$$

The iteration process of FCM clustering algorithm is as follows [3, 4]:

Step 1: Start with a random initialization of $\mu^{(0)}$; initialize $A^{(0)}$ and calculate $\mu^{(0)}$; make the number of iterations k=1; select the number of cluster centres c and the fuzziness index m;

Step 2: Calculate the cluster centre, assign a value to $A^{(k)}$, and calculate $\mu^{(0)}$ according to Expression (2);

Step 3: Recount the membership, assign a value to $\mu^{(k)}$, and calculate $A^{(k)}$ according to Expression (3);

Step 4: If $\max|\mu_{ij}^{k} - \mu_{ij}^{k-1}| \leq \varepsilon$ and the algorithm convergence is achieved, stop the iteration process; otherwise, make k=k+1 and return to Step 2; ε is a given threshold value.

2.3 PARAMETER SETTING OF FCM CLUSTERING ALGORITHM

The fuzzy weighting exponent m is a parameter introduced by Bezdek in order to popularize the criterion functions in fuzzy clustering. m must lie between 0<m<∞. When m approaches to 1, FCM clustering algorithm will lose its fuzziness; when m approaches to +∞, FCM clustering algorithm will lose the meaning of clustering. We usually assign 2 to m.

2.4 ADVANTAGES AND DISADVANTAGES OF FCM CLUSTERING ALGORITHM

FCM clustering algorithm is featured by simple design, fast convergence, wide application areas, easy implementation, and strong local search ability. The algorithm is rooted in traditional C-means clustering algorithms and in essence, it is a type of local search algorithm. So, it inevitably has the disadvantages of C-means clustering algorithm. If the initial cluster centre is next to a local optimum, the algorithm will converge to a local minimum. It is sensitive to the initial value and noisy data. For different initial values, the clustering results are dissimilar. As a result, the objective function may easily falls into local optima. Worse still, it may lead to a degenerate solution or no solutions at all [5].

3 PSO algorithms

3.1 DESCRIPTION OF PSO

Suppose the speed of a basic particle is v and its position is x. The iterative formula of the basic PSO algorithm is described as:

$$v_{t+1} = v_t + c_1 r_1 (pbest - x_t) + c_2 r_2 (gbest - x_t), \tag{5}$$

$$x_{t+1} = x_t + v_{t+1}. \tag{6}$$

Equations (5) and (6) are respectively the iterative formulas of the speed and position. c_1 and c_2 are learning factors; r_1 and $r_2 \in$ U(0,1); *pbest* and *gbest* stand for the position of the optimal solution of the individual and the entire group, separately. The value of speed v is:

$$v_{t+1} = \begin{cases} v_{\max}, v_{t+1} > v_{\max} \\ v_{t+1}, -v_{\max} < v_{t+1} < v_{\max} \\ -v_{\max}, v_{t+1} < -v_{\max} \end{cases}, \quad (7)$$

where $v_{\max}$ is the value range of speed v. The iterative formula of speed v in Expression (6) is divided into three parts: v_t, c_1r_1 ($pbest$-x_t), and c_2r_2 ($gbest$-x_t). The first part stands for the speed of the particle; the second part represents its cognition level, meaning the particle should be considered based on its own conditions; the third part is the social part of the particle in group learning, which means that learning is conducted through information exchange with the *"society"*. The particle updates itself through its current state (first part), self-learning (second part), and the social information (third part). If the third part is not available, the particle will lack information exchange, and the probability of obtaining the optimal solution is thereby reduced. The behaviour of working behind closed doors and learning blindly will lead to a lack of exchanges and has a negative influence on the final results. If the second part is not available, the particle will easily fall into local optima, though there are enough exchanges and extended searching space. Hence, PSO is an intelligent model composed of both *"cognitive part"* and *"social part"* [6, 7].

However, the basic PSO algorithm ignores the first part – the effect of the particle speed on algorithm convergence. To solve the problem, Y. Shi and R. C. Eberhart improved the basic PSO algorithm in 1998 by adding an inertia weight factor into the iterative equation of the speed. The improved equation is:

$$v_{t+1} = wv_t + c_1r_1(pbest - x_t) + c_2r_2(gbest - x_t), \quad (8)$$

where w is the inertia weight factor and the rest parameters are the same as before. At present, the standard PSO algorithm is the amended algorithm with an inertia weight factor [8].

3.2 ITERATIVE STEPS OF STANDARD PSO

Step 1: Initialize m particles, including their initial speed and positions; initialize all relevant parameters;
Step 2: Suppose the present position of the particle is the best position $pbest$; select the global best position $gbest$ according to the best position of all particles;
Step 3: Each particle updates its speed in line with Expression (8), keeps the speed within v_{max} based on Expression (7), and updates its position according to Expression (6);
Step 4: If the current position after the update is better than the previous best position of the particle, update $pbest$ and replace the previous best position with the current position;
Step 5: Update the global best position $gbest$ according to the best position of all particles;
Step 6: If it meets the requirements for termination, stop the iteration process; otherwise, return to Step 3.

4 SP-FCM algorithm design

4.1 BASIC IDEAS OF THE ALGORITHM

Rooted in HCM, FCM clustering algorithm inherits both its advantages and disadvantages. In nature, it is also a type of local search algorithm. Below are the disadvantages [9]:

1) Very sensitive to the initial values and noisy data; easy to fall into local optima;

2) The rate of convergence will be greatly reduced if the initial cluster centre is far away from the optimal point;

3) The algorithm will converge to a local optimum if the initial cluster centre is next to the local optimum.

To solve the above disadvantages, the thesis uses PSO, SFLA, and other intelligent optimization algorithms to find the optimal initial cluster centre and thus prevent FCM from being sensitive to initial values or falling into local optima. The specific theoretical basis is as follows [10]:

1) PSO has strong global search ability, fast rate of convergence, few parameters, and other advantages. PSO makes it easier to rapidly lock the range of solutions of the optimal initial cluster but also easier to fall into the local optimum;

2) SFLA has the advantages such as strong global optimization ability, strong ability to jump out of local optima, and ability to enhance the search ability of the optimal initial cluster centre. But it has poor local search ability and slow rate of convergence.

Therefore, the thesis puts forward a new SP-FCM algorithm on the basis of current literature. Experiments show the algorithm can effectively increase the searching ability and clustering results of fuzzy clustering algorithms. It combines SFLA and PSO by setting a searching granularity coefficient. Thus, it can find the optimal initial cluster centre and avoid the disadvantages of FCM by dint of the advantages of the two. After that, it adopts the basic FCM algorithm [11].

4.2 IMPROVED BASIC ALGORITHMIC PROCESS

The algorithmic process contains four modules: Init Module, SP Module, FCM Module, and Output Module. The specific functions of each module are as follows.

4.2.1 Init Module

The function of this module is to initialize Q cluster centres. Make the cluster sample set $X_t = (x_1, x_2, ..., x_i, ..., x_n)$, where x_i is the d-dimensional vector. Make an individual represent the set of cluster centres, namely, $C_t = (c_1, c_2, ..., c_i, ..., c_c)$; c_i is the d-dimensional code of the i^{th} cluster centre. Real-number encoding is applied to each d-dimensional cluster centre. Count the fuzzy matrix and the fitness value. Arrange the Q cluster centres in descending order of their fitness values. The fitness function of individual c_i is defined as:

$$f(x_i)=\frac{1}{J(\mu,A)+1}. \tag{9}$$

4.2.2 SP Module

This module is aimed at finding the optimal cluster centre according to the fitness value. Combine PSO and SFLA through a searching granularity coefficient. Integrate PSO into SFLA and update the operators of SFLA. The basic procedure of the module is shown in Figure 1.

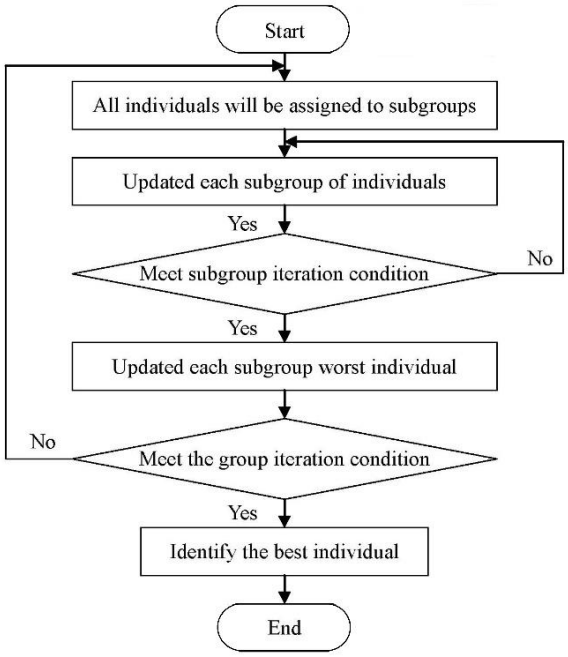

FIGURE 1 Flow Chart of SP Module

The algorithm needs to fully make use of the strong search ability of PSO in the earlier stage so as to narrow down the search range and accelerate convergence. During late stages of the iteration, it is necessary to make full use of the ability of SFLA to jump out of local optima so as to expand the search range and avoid falling into local optima. PSO and SFLA have their own advantages and disadvantages [12]. To put it simply, the combination of the two does not only inherits their advantages but also their disadvantages. The update strategy of PSO is used in each subgroup during the search. Meanwhile, mixed update operation is applied to each subgroup, which reduces the rate of algorithm convergence as well as the local search ability of PSO. Thus, the thesis proposes a searching granularity coefficient for the search based on the advantages of the two. So, the algorithm can make full use of the advantage of PSO in realizing rapid optimization in the earlier stage so as to find the basic range of the optimal solution rapidly; after that, it greatly increases the ability of SFLA to jump out of local optima and thus avoids falling to a local optimum. The searching granularity coefficient is defined as:

$$Granularity=\left[\sqrt[\eta]{\frac{Iter}{runIter+\varepsilon}}\right], \tag{10}$$

where ε is the initial searching granularity; η is the global searching granularity; *Granularity* represents the searching granularity coefficient; *Iter* is the group's maximum number of iteration; *runIter* is the group's current number of iteration. We can control the requirements for searching precision by initializing initial searching granularity ε and global searching granularity η. Prior knowledge is required for each data set. Following a linear decrease, we assign the values of 0.9 to 0.1 to inertia weight factor *w*. The definition is as follows:

$$\omega=\omega_{\max}-run\frac{(\omega_{\max}-\omega_{\min})}{runMax}, \tag{11}$$

where $\omega_{\max}$, $\omega_{\min}$, *run*, and *runMax* denote the maximum inertia weight coefficient, the minimum inertia weight coefficient, the current number of iterations, and the maximum number of iterations, respectively.

4.2.3 Fcm module and output module

The function of Fcm Module is to update the fuzzy matrix through FCM algorithm after obtaining the optimal cluster centre *c*. Output Module outputs the clustering results based on the fuzzy matrix [13].

4.3 ALGORITHM DESCRIPTION

Step 1: Initialize *Q* cluster centres; *runIter* is the counter of the group's number of iterations and *runsubIter* is the counter of the subgroup's number of iterations; initialize all relevant parameters;
Step 2: Count the fuzzy matrix of each individual with Equation (2) and count the fitness value $f(x_j)$ with Equation (9);
Step 3: Arrange the cluster centres in descending order of the fitness values and assign each individual in *Q* cluster centres to subgroup *q* by the allocation strategy of SFLA. Calculate the current optimal solution X_{best} and the current worst solution X_{worst} of each subgroup as well as the group's optimal solution in line with the fitness value X_{gbest};
Step 4: Use Equations (8) and (6) to update V_j and X_j;
Step 5: If *runsubIter*<*Granularity*, return to Step 4; otherwise, go to Step 6;
Step 6: Update the current worst solution X_{worst}; if successful, switch to Step 8; if not, switch to Step 7;
Step 7: Update the current worst solution X_{worst};
Step 8: Update the fitness value $f(x_j)$; if *runIter*<*Iter*, return to Step 3; otherwise, go to Step 9;
Step 9: Find the optimal fitness value X_{gbest}, which represents the fuzzy matrix set of the optimal cluster centres.

4.4 SELF-ADAPT ADJUSTMENT FOR SEARCHING SPACE BASED ON TRUST REGION

The trust region method is another important way to solve the non-restriction optimization, different with the linear searching. The most obvious advantage is the global convergence.

The trust region thought is define the model function max $m_k(x)$ in the first on every iteration point x^k, and desire it could well approximate the object function *f* in the x^k 's suitable neighbourhood. The neighbourhood is called Trust Region, TR, which defined as:

$$S_k = \left\{ x \in \Re^n \middle| \left\| x - x^k \right\|_k \le \Delta_k \right\}, \tag{12}$$

where Δ_k is the trust region radius, $\|\bullet\|$ is a norm rely on the iteration. Based on this trust region, seek for the optimal trail step s^k for $m_k(x)$ which satisfies $x^k + s^k$, and make $m_k(\bullet)$ decrease and $\|s^k\| \le \Delta_k$. If $x^k + s^k$ make *f* decline enough, it will be accepted as the next iteration point which increases Δ_k , or keeps Δ_k unchanged; otherwise, make x^{k+1} as x^k , decrease Δ_k and shrink the trust region to a better similarity for the model function and object function on the new trust region. Part 1 algorithm flow is:

Step 1: Set the initial point $x^1 \in \Re^n$, trust region radius Δ_1, the constant η_1, η_2, r_1 and r_2 , which satisfied $0 < \eta_1 \le \eta_1 < 1, 0 < r_1 \le r_2 < 1$, and make $k = 1$;

Step 2: Select the norm $\|\bullet\|_k$ and define the model $m_k(x)$ on the $S_k = \left\{ x \in \Re^n \middle| \left\| x - x^k \right\|_k \le \Delta_k \right\}$;

Step 3: Calculate s^k to make m^k declined enough and $x^k + s^k \in S_k$;

Step 4: $p_k \dfrac{f(x^k) - f(x^k + s^k)}{m_k(x^k) - m_k(x^k + s^k)}$, if $p_k \ge \eta_1$, set $x^{k+1} = x^k + s^k$; otherwise $x^{k+1} = x^k$;

Step 5: Make $\Delta_k \in \begin{cases} [\Delta_k, \infty] & p_k \ge \eta_2 \\ [r_2\Delta_k, \Delta_k] & p_k \in [\eta_1, \eta_2) \\ [r_1\Delta_k, r_2\Delta_k] & p_k < \eta_1 \end{cases}$

4.5 THE DYNAMIC NEIGHBORHOOD MULTI-SPECIES PARTICLE SWARM PARAMETERS OPTIMIZATION

Bring in the dynamic neighbourhood strategy into the particle swarm algorithms and divide the total swarm into several sub-species with self-adaptation, increase the information share between particles in the neighbourhood topology. First, we define the neighbourhood space distance R_1 as follows:

$$R_1 = \max\left(p_{bestjd}(k), x_{id}(k)\right) - \min\left(p_{bestjd}(k), x_{id}(k)\right), \tag{13}$$

$p_{bestjd}(k)$ is the *d-th* dimension of the *j-th* particle's best solution in the neighbourhood, x_{id} is the *d-th* dimension's position of the *i-th* particle. The Equation (13) could be divided into 3 situations:

Situation 1: If $x_i < \min\left(p_{bestj}\right)$, that is the particle position is less than the neighbourhood particle historical optimal value, shown as Figure 2a. There into the circle mark is the current particle position, the pentagon is the swarm optimize position in the neighbourhood. The rhombus is the distance from unit historical optimal value P_{besti} to every optimal value in each neighbourhood, then $R_1 = \max(L_i) = L_s = \max\left(\mathrm{p}_{bestj}\right) - x_i$;

Situation 2: If $\mathrm{x}_i > \min\left(\mathrm{p}_{bestj}\right)$ and $\mathrm{x}_i < \max\left(\mathrm{p}_{bestj}\right)$, that is the particle is among all the optimal value, shown as the Figure 2b, then $R_1 = L_1 + L_s = \max\left(\mathrm{p}_{bestj}\right) - \min\left(\mathrm{p}_{besti}\right)$;

Situation 3: If the particle position is in the right of every optimal value, shown as the Figure 2c, then $R_1 = \max\left(L_i\right) = L_1 = x_i - \min\left(\mathrm{p}_{bestj}\right)$, thus we can conclude Equation (13).

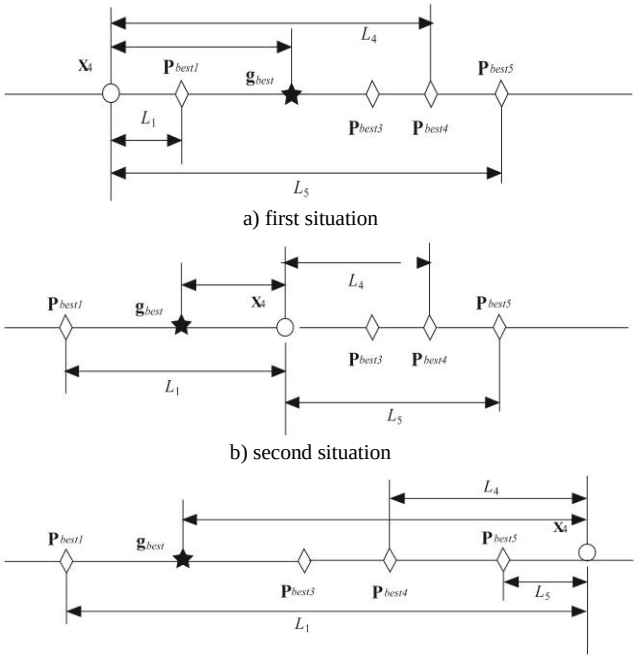

FIGURE 2 Diversity analysis diagram

Define the diversity of the potential searching space of the global particle swarm algorithms, the *j-th* neighbourhood, as:

$$diversity_j = \frac{1}{SL} \sum_{i=1}^{s} |R_1|, \tag{14}$$

where *S* stands for the particle number in the neighbourhood, *L* stands for the linking length. There are three neighbourhood creation methods.

In the first method, the nearest neighbourhood in the searching space $N_i(k)$ is:

$$N_i(k)=\left\{j, j\neq i \middle\| x_i(k)-x_j(k)\right\| \leq \zeta_i\}. \quad (15)$$

Make a judgment rely on the distance between different particle space positions $x_i(k)$. When it's less than the threshold value ζ_i, then create the topological relation between the two neighbourhoods;

The second method is nearest neighbourhood of the adaptive value function space:

$$N_i(k)=\left\{j, j\neq i \middle\| f(x_i(k))-f(x_j(k))\right\| \leq \zeta_i\}. \quad (16)$$

Make the judgment rely on the adaptive function distance and divide the particles with different degree of evolution into different neighbourhood;

The third method is random neighbourhood topology:

$$N_i(k)=\left\{j, j\neq i \middle| \varepsilon_{ij} \leq \varepsilon\right\}. \quad (17)$$

The global Particle Swarm Optimization with Dynamic Neighbourhood Topology, PSO-DNT, could be achieved with follow steps, shown as Figure 3.

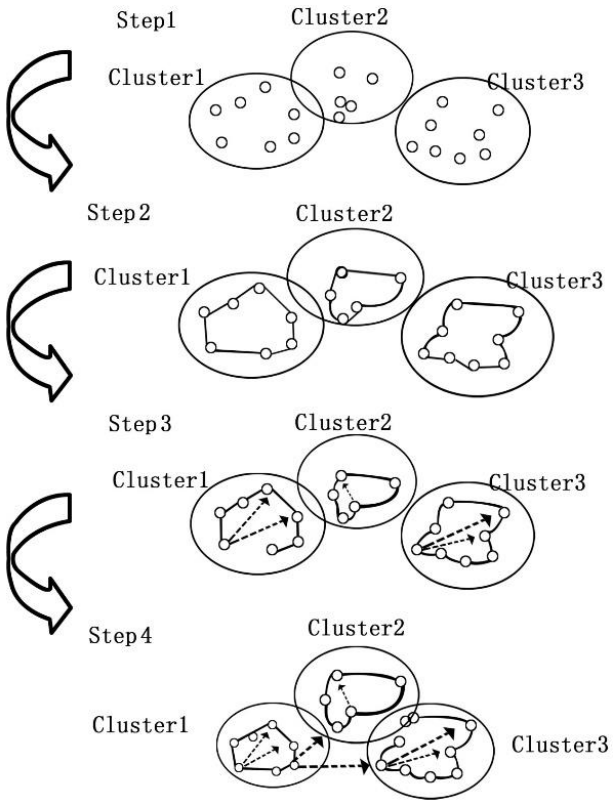

FIGURE 3 Dynamic neighbourhood topology schematic diagram

First divide the overall particle swarm into C sub-species with LA-FCM algorithms, as Step 1. It's noticed that the fussy clustering is aimed at particles, while Part 2 is aimed at all dimensions. The process objects are different. Then create circle topology structure and inner random topology structure in step 2 and step 3 to guarantee the complete validity of the neighbourhood optimization information. Third, have the regular algorithms optimization, when the particle searching space decrease gradually, that is $diversity < d_{\min}$, create the external random topology structure between species to ensure the information sharing.

Thus, the speed evolution formula based on the neighbourhood topology is:

$$v_{id}(k+1)=wv_{id}+c_1 rand_{id}(\bullet)\left[p_{id}(k)-x_{id}(k)\right]+ c_2 rand_{id}(\bullet)\left[l_{gd}(k)-x_{id}(k)\right], \quad (18)$$

where $l_{gd}(k)$ is the current position of the optimal particle in the neighbourhood. The method mainly modified the third part of the evolution formula. It abandons learning the swarm optimal value of the total algorithms and pays more attention to the mutual learning between neighbourhood particles.

5 The simulation analysis of large-scale parameter optimization

We selected 20 standard functions for testing. The large-scale optimization functions are two types: separable functions and inseparable function. When

$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n)= \left(\arg\min_{x_1} f(x_1,\ldots),\ldots,\arg\min_{x_n} f(\ldots,x_n)\right). \quad (19)$$

Every independent variable $x_1,\ldots,x_n$ is independent with each other, then its separable function and easily solved. When there are m variables to the most are dependent, its call m-inseparable function. When every random two variables are dependent with each other, it is called inseparable function.

The 20 standard functions are divided four class:

1 separable functions.

2 partially inseparable function, in which there are fewer parts are relevant.

3 partially inseparable function, in which there are large parts are relevant.

4 completely inseparable function.

All the experiments repeat for 20 times. The testing initial conditions: Testing Dimensions=1000, Cluster Number=10, Iteration Times=3000000, Sub-species Iteration Times=2000, Particle Dimensions=100, Acceleration Coefficient $c_1=1.49, c_2=1.49$, Inertia Weight $\omega=0.8$. The convergence curve of mentioned SP-FCM is shown in Figure 4.

According to the Figure 4, the mentioned SP-FCM algorithms achieved better results. It is also much closed to the best result with stronger stability. In a word, the mentioned SP-FCM algorithm gets the most optimal value of the 20 testing functions and has a better optimization result.

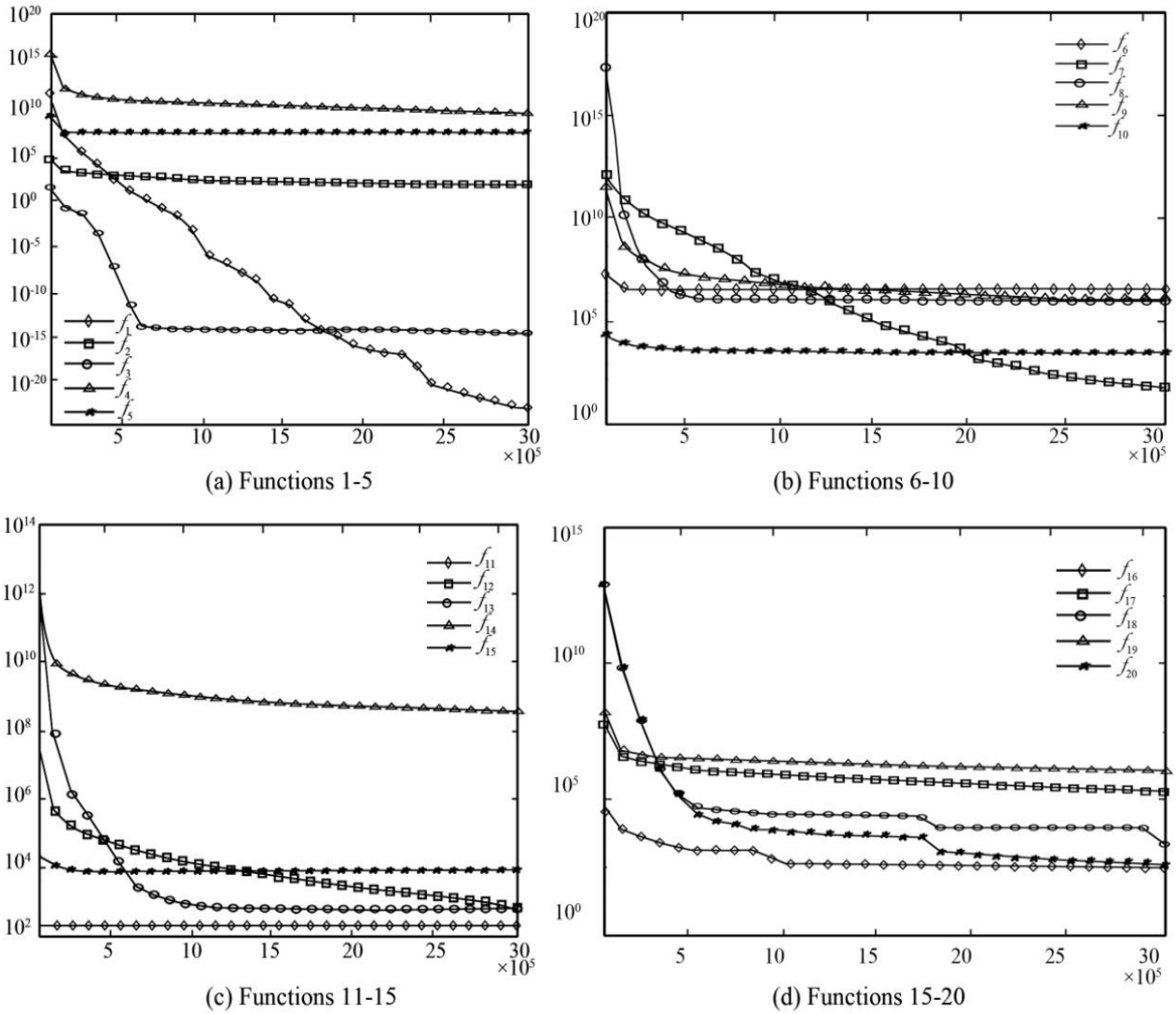

FIGURE 4 Convergence curves of the proposed SP-FCM algorithm

6 Conclusions

In actual industrial process, the control system is much more complicated, the parameters to optimize is more and more, in which case the normal optimization often lose the validity. This article came up with the particle swarm large-scale parameters optimization based on the Fussy C-Means Clustering to solve the curse of dimensions. But the Fuzzy C-Means clustering algorithm still has the sensitivity to the initial points. In this case, this article firstly put forward the Fussy C-Means two-phase clustering algorithms based on linear distribution to get better initial points by means of linear distribution strategy, and then substituted them into the global circulation. It has discovered that the mentioned algorithm had a better clustering result by simulation analysis. And then, this article puts forward a new SP-FCM algorithm on the basis of current literature.

Experiments show the algorithm can effectively increase the searching ability and clustering results of fuzzy clustering algorithms. It combines SFLA and PSO by setting a searching granularity coefficient. Thus, it can find the optimal initial cluster centre and avoid the disadvantages of FCM by dint of the advantages of the two. After that, it adopts the basic FCM algorithm, and it achieved the data property weight and initial cluster centre to guide the overall cluster progress, which efficiently improve the calculate speed and cluster precision.

Acknowledgments

The authors would like to thank the anonymous reviewers of this paper for their carefully reading of the manuscript as well as their many constructive comments. This paper is supported by the NSFC authority-Shenhua group "Coal Joint Fund" Project (Program No. U1361121), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2012JM8093) and Key Project of Chinese Ministry of Education (Program No.211179).

References

[1] Kennedy J, Eberhart R C, Shi Y H 2009 Swarm Intelligence *Singapore: Elsevier*
[2] Yousfi C, Tournier R 2011 Steady state optimization inside model predictive control *Proceedings of American Control Conference* Boston USA 2011 1866-70
[3] Scokaert P O M, Mayne D Q 2009 *IEEE Transactions on Automatic Control* **43**(8) 1136-42
[4] Bemporad A, Borrelli F, Morari M 2012 *IEEE Transactions on Automatic Control* **47**(12) 1974-85
[5] Tang P L, de Silva C W 2012 *IEEE Transactions on Control Systems Technology* **14**(4) 707-18
[6] Zhang T J, Feng G, Lu J H 2011 *IEEE Transactions on Fuzzy Systems* **15**(4) 686-98
[7] Lu C H, Tsai C C 2010 *IEEE Transactions on Industrial Electronics* 2010 **55**(3) 1366-75
[8] Munoz de la Pena D, Christofides P D 2010 *IEEE Transactions on Automatic Control* **53**(9) 2076-89
[9] Torrico B C, Roca L, Normey-Rico J E, Guzman J L 2011 *IEEE Transactions on Control Systems Technology* **18**(6) 1430-9
[10] Qin S J, Badgwell T A 2009 A survey of industrial model predictive control technology *Control Engineering Practice* **11**(7) 733-64
[11] [Grosdidier P, Froisy J B, Hamann M D 2008 The IDCOM-M controller *Proceedings of IFAC Workrhop on Model Based Process* Atlanta USA 31-6
[12] Scokaert P O M, Mayne D Q 2008 *IEEE Transactions on Automatic Control* **43**(8) 1136-42

Authors

Xuhui Zhang, born in October, 1972, Shaanxi province, China

Current position, grades: professor and dean of Mechanical Engineering, Xi'an University of Science and Technology, China.
University studies: PhD in Instrument Science and Technology (2009) at Xi'an Jiaotong University.
Scientific interest: Intelligent detection and control, non-destructive testing and evaluation, condition monitoring and fault pattern recognition.
Publications: 70 papers, 5 books.
Experience: more than 20 years.