Local search ability of frog leaping algorithm in fuzzy controller parameters optimization

Yifei Chen*, Sen Xu

School of Information and Engineering, Yancheng Institute of Technology, Yancheng 224051, Jiangsu, China

Received 1 June 2014, www.cmnt.lv

Abstract

To solve the problem of slow searching and poor searching ability in tuning PID controller parameters and optimizing fuzzy controller parameters with basic frog leaping algorithm, this paper proposes an improved shuffled frog leaping algorithm based on PID fuzzy controller parameters optimization. First by adding the self-adaptive learning factor, this new algorithm accelerates the convergence speed, expands the search area of the individuals, and maintains the diversity of population, which extends the search ability to a certain extent. And then by introducing the constriction factor and acceleration factor in the particle swarm optimization algorithm, the updating strategy is improved to speed up the search speed of the algorithm, and at the same time to ensure the convergence of the algorithm. Simulation results show that, the improved shuffled frog leaping algorithm proposed by this paper has excellent performance, and is suitable to PID fuzzy controller parameters optimization.

Keywords: PID fuzzy controller parameters optimization, improved shuffled frog leaping algorithm, self-adaptive learning factor

1 Introduction

PID control is a feedback control mechanism, which adopts proportion to reduce the deviation, the integral to eliminate the static error, and the differential to predict the future [1]. Due to its clear principle, simple structure, easy realization, it is widely applied in control field. The key of PID control is choosing proper parameters to adapt to different control object, but for these time-varying and large delay control objects, the control effect is not very ideal and it is also very difficult to achieve the control requirements, the main reason of which is that the control parameters of PID controller does not have the online self-tuning capability [2].

As for the problem that systems with strong nonlinear or uncertainty do not have ideal effect with the use of PID, some domestic and foreign research scholars have done rather deep research [3]. Shi Weifeng et al. designed a PID controller based on CMAC. Since CMAC is a kind of neural network based on local learning with fast learning speed, it is suitable for real-time control [4]. As for the pure lag industrial object, Wang Yaonan et al, proposed a control method of PID parameters self-tuning based on RBF network, which adopted a new learning algorithm integrated LMS algorithm with gradient method, and this control method combined with Smith predictor would obtain certain control effect when applied to pure lag industrial object [5]. For the large time-delay and time-varying object, Ji Chunguang et al, combined the Smith compensation control theory and methods of PID parameters self-adjusting by neural network, and proposed a self-tuning Smith PID control algorithm based on BP neural network[6]. In recent years, it has become the main development trend of intelligent control that using genetic algorithm to optimize weight coefficient of the neural network [7]. The key to the successful application of genetic algorithm in neural network -PID is whether it can achieve real-time control, but for the fast object, there are some difficulties at present [8]. As the same with the PID controller, though genetic algorithm-neural network-PID controller could solve the control problem of time-varying object well, it is still not suitable to control of the large time-delay and rapid response nonlinear object [9]. Nowadays the neural network intelligent control has been got considerable development and progress both in theory and application. But how to integrate the PID controller with intelligent control better together so that the PID controller can be well applied to engineering practice is still an important subject of research and exploration for control field. In this field, many researchers are fusing all kinds of intelligent control technology with conventional PID control method together and have proposed various forms of intelligent PID controller [10].

Aiming at the defects of poor adaptability of PID fuzzy controller, this paper puts forward an improved shuffled frog leaping algorithm based on PID fuzzy controller parameters optimization, and optimizes the search ability and search speed of basic shuffled frog leaping algorithm.

* ***Corresponding author's*** e-mail: chenyf1968@sohu.com

2 Shuffled frog leaping algorithm

2.1 ALGORITHM MODEL

Shuffled frog leaping algorithm (SFLA) algorithm is a bionic optimization algorithm with global collaborative search. This algorithm is inspired by the foraging behaviour of frog: a group of frogs live in a place distributed many stones, and the frogs seek for the food by jumping through the different stones. Each frog improves his ability to search for food by looking for different stones, and frogs communicate mutually to realize information sharing.

For the M -dimensional optimization, assuming the population size of the frog is S , the frog i represents a solution in the solution space of $U(i)=\{U_{i1},U_{i2},...,U_{ij},...,U_{iM}\}$, among which, $1\le i\le S$, $1\le j\le M$. Calculate the fitness value of each frog in the population, and then sort them according to the fitness values in descending order. The frog population is divided into m groups, with each group contains n frogs, which satisfy the relation: $S=m\times n$. At the same time, divide the sorted frogs into ethnic groups according to the following rules: the first frog is divided into the first ethnic group, the second one divided into the second group, and the like, the frog m divided into the group m . Then, the frog $m+1$ is divided into the first group, and the frog $m+2$ divided into the second group, and like so to continue the cycle distribution until all the frogs were assigned. The frog with the best and worst fitness value in the group are respectively set as P_b and P_w , and the frog with the best fitness value in the population is set as P_g . Before meeting the predetermined local iterations times, perform local search in each ethnic group and realize the optimization of P_w through the way of P_w learning from P_b . After completing the local search in each ethnic group, the frogs will jump among the ethnic groups and mix into a new population, so a global search is also completed.

2.2 ALGORITHM FLOW

Shuffled frog leaping algorithm (SFLA) can be basically divided into 3 processes: global search, local search and mixed operation. The global search flow is as follows:

1) Parameters of algorithm. The number of ethnic groups is m , the maximum number of iterations of local search is $J_{\max}$. The maximum number of iterations of global search is $Q_{\max}$. The number of frogs contained in each ethnic group is n . The size of frog population is $S=m\times n$. The dimension of solution space of the optimization problem is M , and upper and lower bounds of the search space are respectively recorded as H and L.

2) Population initialization. In the M dimensional feasible solution space, randomly generate S frogs, $U(1),U(2),...,U(S)$, and the frog i is presented as $U(i)=\{U_{i1},U_{i2},...,U_{ij},...,U_{iM}\}$, among them, $1\le i\le S$, $1\le j\le M$, $U_{ij}\in[L,H]$.

3) Fitness function. Through the fitness function $f(i)$ to evaluate the degree of performance of each frog $U(i)$.

4) Division of frog ethnic groups. The frogs are ranked: according to the fitness function $f(i)$, sort the frogs according to the fitness value in descending order, and generate the arrays $P=\{U(i),f(i),i=1,2,...,S\}$, i.e. the frog $U(1)$ is the best individual in the frog population, and is denoted as the global extremum $P_g=U(1)=\{U_{11},U_{12},...,U_{1j},...,U_{1M}\}$. The frogs are grouping into ethnic groups according to Equation (1).

$$Z_k(i)=U(k+m\times(i-1)) \; . \tag{1}$$

Among them, $i=1,2,...,n$, $k=1,2,...,m$, distribute the sorted frogs into m groups $Z_1,Z_2,...,Z_m$, with each group contains n frog; for example, $m=3$, $n=2$, namely, $S=6$, so the frog $U(1)$ is entered into group Z_1, $U(2)$ entered into Z_2 , $U(3)$ enter into Z_3 , $U(4)$ entered into Z_1 , $U(5)$ entered into Z_2 , $U(6)$ entered into Z_3 .

Execute the evolution in each ethnic group. In ethnic groups, the frog is affected by the individual culture of other frogs' in the same ethnic group, through evolution, each frog approaches towards the local extremum of their group.

The local search flow of SFLA is as follows:

1) Initialize $L_m=0$, L_m ranging from 0 to m , is used to count the population; initialize $J_n=0$, J_n ranging from 0 to $J_{\max}$, reflects the number of iterations of ethnic local search. The best and worst individuals within ethnic group are respectively expressed as
$P_b=\{U_{b1},U_{b2},...,U_{bj},...,U_{bM}\}$,
$P_w=\{U_{w1},U_{w2},...,U_{wj},...,U_{wM}\}$.

2) $L_m=L_m+1$.

3) $J_n=J_n+1$.

4) If $J_n\le J_{\max}$, on the condition of meeting the following Rules 1 and 2, update the current position of ethnic group P_w according to Equation (2) and (3) to enhance the fitness value of P_w . The updating rules and formulas are shown as follows:

Rule 1: Set the moving step length of the worst frog individual is D , $D=\{D_1,D_2,...,D_i,...,D_M\}$, $i\in[1,M]$, $D_i\in[-D_{\max},D_{\max}]$. If $D_i<-D_{\max}$, D_i is set to $-D_{\max}$; if $D_i>D_{\max}$, D_i is set to $D_{\max}$;

Rule 2: The updated frog individual is $\hat{P}_w=\{\hat{U}_{w1},\hat{U}_{w2},...,\hat{U}_{wj},...,\hat{U}_{wM}\}$, $j\in[1,M]$, $\hat{U}_{wj}\in[L,H]$. If $\hat{U}_{wj}<L$, D_i is set to L ; if $\hat{U}_{wj}>H$, D_i is set to H .

The moving step length of frog is calculated by

$$D = rand() \times (P_b - P_w) . \quad (2)$$

The calculation Equation of P_w is:

$$\hat{P}_w = P_w + D . \quad (3)$$

Among them, $rand()$ is a random number between 0 to 1, indicating the degree of approximation of P_w toward P_b;

5) If the above updating strategy could improve the adaptive value of P_w, which means it could produce a better solution, substitute the new generation of frog for the original frog, namely $P_w = \hat{P}_w$; otherwise, substitute P_g for P_b, and return to step (4) to re-execute the frog updating strategy. If the update is finished, but the fitness value has not been improved, it randomly generates a new solution to replace the original;

6) If $J_n < J_{\max}$, execute step (3);

7) If $L_m < m$, execute step (2); otherwise, the frog jumps in the inter-ethnic groups to communicate culture, and execute the mixed operation, re-converge and resort to perform the global search.

The flow of mixed operation of SFLA algorithm is as follows:

1) After the local search are executed in all ethnic groups, the frogs jump and move among the ethnic groups, so all the frogs reassemble into a population;

2) Then add 1 to the current global iteration number, if $I < Q_{\max}$, return to the step (3) in global search process, and continue to execute the fitness function, the frog ethnic groups division, local search and mixed strategy;

3) If the termination condition is satisfied, output the global optimal solution P_g. Generally, the defined global maximum iteration number $Q_{\max}$ is chosen as the termination criterion situation.

3 Improvement of shuffled frog leaping algorithm

3.1 OPTIMIZATION OF SELF-ADAPTIVE LEARNING FACTOR

Because of the frog individuals' memory function, it can remember its last update step and historical optimal value of neighborhood individual, so during the iterative process P_w, not only learns from P_b, but also from P_g, and continues the inertia step of last update part as well as learns from the historical optimal value of neighborhood individual in memory at the same time. As with the iterations increases, the effect of learning factor on individual update strategy weakens in linear trend. This update strategy accelerates the convergence speed, at the same time, expands the search area of the individual, and maintains the diversity of population, which extends search ability of the algorithm to a certain extent. The specific update strategies is shown as Equations (4-6).

$$Dis(t+1) = W(R_1 Dis(t) + R_2 His(P_w)) + R_3(P_b - P_w) , \quad (4)$$

$$W = W_e + (W_s - W_e)\frac{T_{\max} - t}{T_{\max}} \text{ g}, \quad (5)$$

$$P_w(t+1) = P_w(t) + Dis(t+1) . \quad (6)$$

In Equations, R_1, R_2, and R_3 are all random number between 0 to 1. Dis is the moving step of P_w. t is the current number of iterations. $T_{\max}$ is the total number of iterations. W is the weight factor. W_s and W_e are respectively the initial value and end value of weight factor, the value of them in this paper were respectively 0.9 and 0.4. $His(P_w)$ is the historical optimal value of individual neighborhood.

3.2 OPTIMIZATION OF CONSTRICTION FACTOR

This paper introduces a new accelerated search parameter c to improve the search speed of SFLA algorithm, but the value of c selected through experiment, has no generality, and cannot guarantee the convergence of the algorithm. The constriction factor x could ensure the convergence of particle swarm optimization algorithm. Therefore, in order to improve the moving speed of the worst individual P_w toward the best individual P_b within subgroups and the optimal individual P_g within the whole population, and at the same time to guarantee the convergence, the constriction factor x and the acceleration factor c_1 and c_2 of particle swarm optimization algorithm are introduced into the basic SFLA algorithm to improve the update strategies. So that the worst individuals P_w learn updating toward the best individual P_b within subgroups according to Equation (7), and towards the global optimal individuals P_g according to Equation (8), thereby improving the learning ability of P_w toward P_b or P_g and speeding up the search speed of the algorithm, at the same time, ensuring the convergence of the algorithm. The value of constriction factor x and the acceleration factor c_1 and c_2 are given according to Equation (9) usually, $c_1 = c_2 = 2.05$, then $x = 0.729$.

$$Dis(t+1) = x \cdot rand() \cdot c_1 \cdot (P_b - P_w) , \quad (7)$$

$$Dis(t+1) = x \cdot rand() \cdot c_2 \cdot (P_g - P_w) , \quad (8)$$

$$x = \frac{2}{\left|2 - \varphi - \sqrt{\phi^2 - 4\phi}\right|}, \phi = c_1 + c_2 > 4 . \quad (9)$$

If the worst individual P_w learns from the optimal individual P_g within subgroups or the global optimal individual P_g without progress, a new individual will be generated randomly within search area to replace P_w. Although this update method increases the diversity of the population, but it do not use the original individual information, which leads to the convergence of algorithm slow down. In order to make full use of the useful information in the original individual, if the worst individual P_w learns from the optimal individual P_g within subgroups or the global optimal individual P_g without progress, it will search according to Equation (10) in a small radius near their own position. If the fitness of $newP_w$ is better than P_w, $newP_w$ will replace P_w; conversely, regenerate state $newP_w$ randomly according to Equation (10) to judge whether the progress condition is met; after repeating *num* times, if the progress condition is still not met, a randomly generated individual within the search scope will replace the original worst individual P_w.

$$newP_w = P_w + (2rand() - 1)Step \tag{10}$$

Among them, *Step* presents the step length of individual randomly moving, $rand()$ is a random number within the region $[0,1]$. The step length of the worst individual self-learning factor are adjusted dynamically according to Equations (11-12).

$$Step = Step \cdot a + Step_{\min}, \tag{11}$$

$$a = \exp(-30 \cdot (t / T_{\max})^2). \tag{12}$$

Under normal circumstances, the initial value of $Step$ is $X_{\max}/16$, ($X_{\max}$ is the maximum value in the search scope), $Step_{\min} = 0.002$, t is the current iteration number, $T_{\max}$ is the maximum iterations number, and the range of s in this paper is $[1,30]$.

4 Algorithm simulation

In PID control, the tuning and optimization of parameters K_P, K_I and K_D greatly affect whether the system optimization design is good or bad, it can be said that the tuning for the three parameters of K_P, K_I and K_D are the most important part of PID control system design.

There has been many kinds of methods for the problem of PID controller parameters optimization, some need to establish mathematical model, some do not rely on the models, and the optimization methods used in this paper belongs to the latter. The basic shuffled frog leaping algorithm and the improved shuffled frog leaping algorithm are used to optimize the PID controller parameters to find out the appropriate K_P, K_I and K_D. And the comparative analysis of the optimization results are made to test the performance of the improved shuffled frog leaping algorithm for this kind of problem.

Basic shuffled frog leaping algorithm and the improved shuffled frog leaping algorithm to optimize the PID controller parameters fitness change curves are shown as follows:

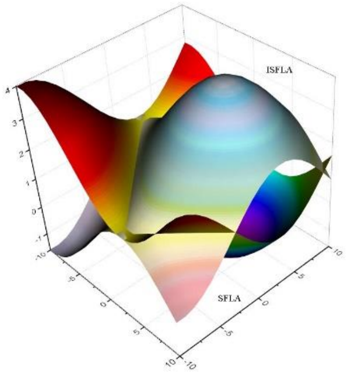

FIGURE 1 Two algorithms to optimize PID controller parameters fitness value curve

Improved shuffled frog leaping algorithm to optimize PID controller parameters step response curve is shown as follows:

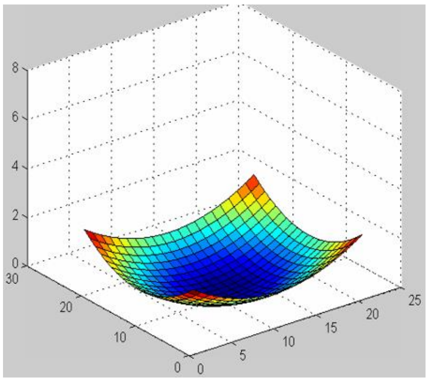

FIGURE 2 Improved algorithm to optimize PID controller parameters step response curve

It can be seen from Figure 1 and Figure 2 that, in solving PID controller parameter tuning problem, both improved shuffled frog leaping algorithm and basic shuffled frog leaping algorithm have very good performance. Compared with basic shuffled frog leaping algorithm that iterated 100 times, the improved shuffled frog leaping algorithm iterated only 50 times, but had equivalent performance, which proved the improved shuffled frog leaping algorithm proposed in this paper had an excellent performance and to be suitable to PID fuzzy controller parameters optimization.

5 Conclusion

Due to the characteristics of clear principle and simple structure, PID control is still widely applied in the industrial field. The conventional PID controller requires that the controlled objects be time-invariant, and a precise mathematical model could be established, however, in practical application, these conditions are very harsh because it is difficult to establish a mathematical model for large numbers of controlled objects, and the characteristics of time-invariant, large delay are very prominent. This paper puts forward a fuzzy controller parameters optimization strategy based on improved shuffled frog leaping algorithm, and simulation results have shown that this kind of algorithm performs well in the optimization of PID fuzzy controller parameters.

Acknowledgments

This work was supported by: 1) The national natural science fund project Based on the matrix of low rank approximation mass text clustering approach to integration research (No. 61105057); 2) 2014 cooperative innovation fund project of Jiangsu Province: Development and application of the Internet of things in the Garden (No. BY2014108-22)

References

[1] Tu H, Chen H 2014 Optimization of PID controller parameters for turbine speed based on Multi-objective genetic algorithm *Journal of Wuhan University of Technology* **36** 67-71 (*in Chinese*)

[2] Yu Y 2014 PID Tuning of Time Delay System Based on Spectral Discretization Methods *Control Engineering of China* **21** 27-31

[3] Chen D 2014 Bacteria foraging optimization algorithm and application of PID parameters optimization *Chinese Mechanical Engineering* **25** 59-64

[4] Wang H 2013 Application of Improved PSO in PID Parameter Optimization of Quadrotor Aircrafts *Electronics Optics & Control* **20** 82-6

[5] Fan J 2013 PSO-GA combination algorithm to optimize the PID parameters and design of visualization platform *Mechanical Design and Manufacturing* **8** 8-11

[6] Guo Z 2013 Application of Improved genetic algorithm in Brushless DC motor speed control system PID parameters optimization *Modular Machine Tool & Automatic Processing Technology* **7** 90-1

[7] Zhang T 2013 PID Parameters Tuning for Load/Unload Manipulators Based on Genetic Algorithm *Machine Tool & Hydraulics* **41** 16-9

[8] Xie P 2013 A simple PID parameters optimization method for low order processes with time delay *Automation Instrumentation* **34** 12-5

[9] Tang J, Li W 2013 PID parameters optimization of NC machine tool table AC servo system based on Genetic algorithm *Modern Manufacturing Engineering* **2** 49-52

[10] Li D, Jing J 2013 Tuning and self-adaption of PID parameters based on immune particle swarm optimization algorithm *Automation Instrumentation* **34** 65-7

Authors

Yifei Chen, 20.05.1968, China.

Current position, grades: associated professor at school of information engineering from Yancheng Institute of Technology, China.
University studies: Master's degree in school of mechatronic engineering from Mining and Technology University, China in 1995.
Scientific interest: computer control.

Sen Xu, 28.01.1983, China.

Current position, grades: associated professor at school of information engineering from Yancheng Institute of Technology, China.
University studies: PhD degree in computer science and technology from Harbin Engineering University, China in 2010.
Scientific interest: machine learning, pattern recognition and data mining.