Cloud Adaptive Chaos Particle Swarm Optimization Algorithm for Economic Load Dispatch of Power System

Zhucai QIN

Department of Automation, Wuhan University of Technology, Wuhan, 430070, China

email:468926608@qq.com

Keywords: Economic Load Dispatch; Cloud Adaptive; Chaotic Variation; Particle Swarm Optimization Algorithm

Abstract. A cloud adaptive chaos particle swarm optimization algorithm is proposed for economic load dispatch problems of power system, which has the characteristics of nonlinear, non-convex and discontinuous. Normal cloud generator was used to adaptively adjust the inertia weight of each particles, so as to optimize their optimization direction and improve the convergence speed of the algorithm; and the chaotic variation operation was introduced to adjust the particle's positions, so as to improve the diversity of the solution and avoid falling into local optimum. Simulation of 6 unit system demonstrates that the proposed algorithm has high accuracy and quick speed used in economic load dispatch of power system.

Introduction

Economic load dispatch of power system (ELD) refers to distribute the load for each generator unit to minimize the total cost, while satisfying the power supply and operational characteristics, which has great significance to improve the economy and reliability of the system [1]. In recent years, optimization method based on swarm intelligence has been gradually used in the ELD problems, such as neural network algorithm, simulated annealing algorithm, the genetic algorithm[2,3], etc. Optimization of these algorithms for ELD problem has played a certain role, but the convergence speed and optimization accuracy still needs to be improved.

Particle swarm algorithm[4] is a new intelligent optimization algorithm. It has the advantage of easy to implement, fast convergence speed and few parameters when compared with other algorithms, but has the defect of easily falling into local optimum at the same time. Therefore, this paper proposes a chaos particle swarm optimization algorithm based on cloud adaptive (CACPSO）for ELD, the algorithm uses the normal cloud generator to adjust the particle's inertia weight, while introduces the chaotic variation, so as to improve the algorithm's convergence speed and accuracy.

The mathematical model of ELD problem

ELD problem is refers to optimize the load distribution for each generator unit then obtain the minimum total power generation cost under the condition of system operation constraints, its ideally goal function model can be approximated as quadratic function:

$$\begin{cases} \min F = \sum_{i=1}^{N} F_i\left(P_i\right) \\ F_i\left(P_i\right) = a_i P_i^2 + b_i P_i + c_i \end{cases} \tag{1}$$

Where F_i is the total power generation cost; N is the generator sets; P_i is the active power of generator i; $F_i\left(P_i\right)$ is the consumption characteristic of generator i; a_i, b_i, c_i are the constant consumption characteristics of generator i.

In actual operation process, the consumption characteristic curve reflects to bump up, namely the valve point effect, which has great effect on the solution accuracy, the consumption characteristic expression changed as follows after considering the effect of broad point:

$$F_i\left(P_i\right) = a_i P_i^2 + b_i P_i + c_i + E_i \tag{2}$$

$$E_i = |g_i \sin(h_i(P_i - P_{i\min}))| \tag{3}$$

Where E_i is the change of consumption characteristic caused by valve point effect; g_i, h_i are the constant characteristics of valve point effect; $P_{i\min}$ is the minimum active power.

The constraint conditions that mainly need to be considered in ELD of power system are power balance constraints and motor operating constraints:

$$\begin{cases} \sum_{i=1}^{N} P_i = P_{Load} + P_{Loss} \\ P_{i\min} \le P_i \le P_{i\max} \end{cases} \tag{4}$$

Where P_{Load} is the total load; P_{Loss} is the total network loss; $P_{i\min}, P_{i\max}$ are the upper and lower limits of generator active power respectively.

Cloud adaptive chaos particle swarm optimization algorithm for ELD

Particle swarm algorithm is a simulation of the birds foraging process. Each feasible solution of the optimization problem represents a bird on the search space, called a particle. All particles have a fitness that determined by the function to be optimized, including individual optimal and global optimal, particles converge to the optimal solution by tracking and studying these two optimal values continuously. Each particle updates according to the following formula:

$$\begin{cases} v_{id}^{t+1} = \omega v_{id}^{t} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 (g_{id}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \end{cases} \tag{5}$$

Where $i = 1, 2, \cdots, N,\ d = 1, 2, \cdots, D$, N and D are the total particle number and dimension respectively; ω is the inertia weigh; c_1 and c_2 are the learning factors; r_1 and r_2 are random numbers which are distributed evenly between 0 and 1; v_{id}^{t} and x_{id}^{t} are the velocity and position of particle i respectively;

Cloud model is a mathematical method that describe the uncertain relationship, it realizes the mutual transformation between the qualitative and quantitative by combining the fuzziness and randomness[5]. Cloud has three digital characteristics: expectations(Ex), entropy(En) and hyper entropy(He), they reflect the quantitative characteristics of qualitative concept. Normal cloud generator reflects the mapping from qualitative to quantitative, according to the three digital characteristics of cloud to generate cloud droplets, specific implementation process is divided into the following three steps:

1) Input the three parameters of cloud: E_x, E_n and H_e .

2) Generate random numbers E_n' and x_i with expectation E_n and variance H_e by using normrnd function, namely $E_n' = normrnd(E_n, H_e)$, $x_i = normrnd(E_x, E_n)$.

3) Calculate $u_i = e^{-\frac{(x-E_x)^2}{2(E_n')^2}}$ and generate the cloud droplet of cloud $i : (x_i, u_i)$

In particle swarm algorithm, inertia weight ω has great importance on particle's search ability, selecting appropriate ω can effectively improve the algorithm's convergence speed and search accuracy. This paper plan to adjust it dynamically by using the fuzzy and random of cloud, take the ratio of global optimal value and current particle's fitness θ as divided interval and then divide the particle population into three parts. The specific methods are as follows:

1) $0 < \theta < \alpha$: These particles belongs to the group that far from global optimal, namely poor fitness particles, their inertia weights should be step up to improve global search ability, take $\omega_{\max} = 0.9$ ·

2) $\alpha < \theta < \beta$: These particles belongs to the group that performance in ordinary, their inertia weights can be adjusted according to the normal cloud generator based on cloud model:

$$\begin{cases} E_x = 5\times\omega_{\max} - \omega_{\min};\ E_n = 5;\ H_e = 0.5; \\ E_n = normrnd(E_n, H_e); \\ \omega_i = \omega_{\max} - \omega_{\min}\times\exp(-(f(i)-E_x)\hat{}\,2/(2\times E_n\hat{}\,2)) \end{cases} \tag{6}$$

3) $\beta < \theta < 1$: These particles belongs to the group that near to global optimal, namely excellent fitness particles, their inertia weights should be reduced to improve local search ability, take $\omega_{\min} = 0.4$.

The basic idea of chaos algorithm is to transform variables from chaos space to solution space, it has the characteristics of global gradual convergence and easy to jump out of local optimal. Make full use of the randomness and ergodicity of chaos to have chaos mutation operation on poor particles with a certain probability, the most common of which are generated by using Logistic chaotic mapping equation, Logistic equation is a typical chaotic systems:

$$y_{n+1} = \mu y_n(1-y_n) \qquad \mathrm{n} = 0,1,2\ldots \tag{7}$$

Where μ is a control parameter, usually values of 4. When arbitrary initial value y is taken in [0,1],it can iterative out a deterministic chaotic sequence $y_1, y_2, y_3\cdots$ according to equation (7).

Basic steps of the improved algorithm for ELD

1) Read the original data, including parameters and constraints.
2) Initialize the particle population and calculate their fitness according to the objective function, then update the individual optimal and global optimal.
3) Adjust the inertia weight with cloud according to formula (6); then update each particle’s position and speed according to formula (5) and obtain the latest individual optimal and global optimal.
4) Select 15% poor fitness particles of the group to have chaotic variation.
5) Judge whether the iteration number meet the limit. If yes, stop the iteration and putout the optimal solution ; otherwise, return to step (3) and continue the iteration.

Tab. 1 Coefficients of generator cost function

Unit Number	a_i	b_i	c_i	$P_{i\min}$ / MW	$P_{i\max}$ / MW
1	0.0070	7.0	240	100	500
2	0.0095	10.0	200	50	200
3	0.0090	8.5	220	80	300
4	0.0090	11.0	200	50	150
5	0.0080	10.5	220	50	200
6	0.0075	12.0	190	50	120

Simulation analysis

Take 6 unit power system (IEEE30 node system) as example in literature [6], ignore the network loss but consider the valve point effect, the total load of generator is 1263MV, the energy dissipation characteristic equation parameters of each unit are shown in table 1. Set algorithm parameters as follows: maximum number of iteration is 200, population size is 40, dimension is 6, inertia weight $\omega_{\max}$ is 0.9, $\omega_{\min}$ is 0.4, the learning factor c_1 and c_2 are both 2.10. Compare the results with PSO and CPSO[7] as it shown in table 2:

Tab. 2 Generator power output and total generation cost of 6 unit

Algorithm	P_{G1}	P_{G2}	P_{G3}	P_{G4}	P_{G5}	P_{G6}	Total Power	Total Cost
PSO	452.73	173.54	258.76	148.16	154.07	75.93	1263.19	15258.36
CPSO	443.32	175.48	259.13	146.27	160.36	78.60	1263.16	15245.37
CACPSO	435.17	180.32	256.45	148.59	161.76	80.81	1263.11	15242.59

As it can be seen from table 2, the total cost of the system is 15242.59($/h) after optimized by the improved algorithm, which have reduced 15.77 ($/h) and 2.78 ($/h) respectively compared with PSO and CPSO, it illustrated that the improved particle swarm algorithm is more efficient used in ELD problem than other algorithms.

Figure 1 shows the convergence curve comparison of three kinds of algorithms in the process of optimization. Obviously, the CACPSO that added chaotic variation and cloud adaptive operation can avoid particle swarm falling into local optimum more effectively, and convergence to the optimal solution faster at the same time.

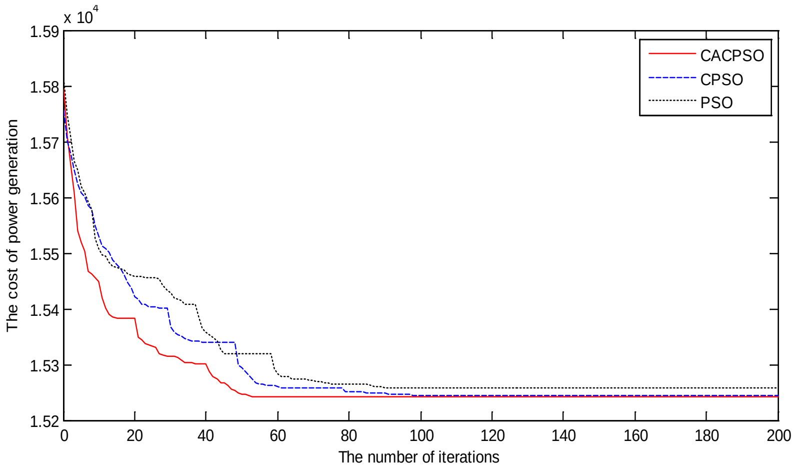

Fig.1. ELD search curve of 6 unit

Conclusion

This paper proposed an adaptive chaotic particle swarm optimization algorithm based on cloud for ELD problem. Normal cloud generator is adopted to adjust each particles' inertia weight to satisfy their different demands of global and local search ability, while the chaotic variation is introduced to avoid falling into local optimal. Results of calculation examples show that the proposed algorithm has higher convergence precision and faster searching speed in solving the ELD problem.

References

[1] Tang Wei,Li Dianpu.Chaotic optimization for economic dispatch of power systems[J]. Proceedings of the CSEE,2000,20(10):36-40.

[2] HE Da-kuo, WANG Fu-li, et al.Application of genetic algorithm on economic dispatch of power systems[J].Journal of System Simulation, 2007(4):890-892.

[3] MAO Ya-lin,ZHANG Guo-zhong.Economic load dispatch of power system based on chaotic simulated annealing neural network mode1[J].Proceedings of the CSEE, 2005, 25(3):65-70.

[4] Yang Bo,Zhao Zunlian,Chen Yunping,et al.An improved particle swarm optimization algorithm for optimal power flow problem[J]. Power System Technology,2006, 30(11) :6-10.

[5] Lu Shijun,Li Xiang,Lin Yanting.Analysis on substation operating costs based on cloud model[J].Power System Technology,2010,34(6):205-209(in Chinese).

[6] Ross D W, Kim S. Dynamic economic dispatch of generation[J]. IEEE Trans. Power App.Syst. 1980, 99(6): 2060-2068.

[7] Su C T,Lin C T.New approach with a Hopfield modeling framework to economic dispatch[J]. IEEE Trans on Power Systems,2000,15(2):541-545.