

Application of Improved Adaptive Particle Swarm Optimization Algorithm in Reactive Power Optimization

Xiangdong Jiang

Benxi Power Supply Branch
State Grid Liaoning Electric Power Supply Co. Ltd
Benxi, China
e-mail: xiangdongjiang@126.com

Zhendong Liu

Benxi Power Supply Branch
State Grid Liaoning Electric Power Supply Co. Ltd
Benxi, China
e-mail: LZD1992@63.com

Ming Zhu

Benxi Power Supply Branch
State Grid Liaoning Electric Power Supply Co. Ltd
Benxi, China
e-mail: bxwl_650928@126.com

Abstract—In this paper, the Particle Swarm Optimization (PSO) algorithm converges too fast, easy access to local convergence, leading to convergence accuracy is not high, to study the particle swarm algorithm improvements. Some parameters in the traditional PSO algorithm need to be predefined by tests, so the practicability of algorithms is restricted. For this reason, an improved adaptive particle swarm optimization (IAPSO) algorithm is proposed and applied to reactive power optimization of Power system. The establishment of a comprehensive consideration of the practical constraints and reactive power regulation means no power optimization mathematical model proposed by improved adaptive particle swarm optimization algorithm for solving the problem of reactive power method to determine the optimal reactive power optimization program. According to PSO handling characteristics of the variable capacity, particle swarm optimization algorithm applied to a continuous emphases on control variables to optimize regulation, reactive power optimized for efficient operation provides a practical and reliable foundation.

Keywords- *particle swarm optimization; adaptive; reactive power; power system ; optimize accuracy*

I. INTRODUCTION

Reactive power transmission and distribution network optimization is very important to power system security and economical operation. It is a multi-variable, complex constraints, strong coupling and mixed-integer nonlinear optimal control problems, including not only the continuous control variables, including discrete control variables, the classic reactive power optimization algorithm is sometimes not fully find globally optimal solution to meet the requirements. In recent years, research in all aspects of artificial intelligence methods have achieved rapid development, application in power systems is relatively broad, and achieved good results.

Recently, multi-objective optimization approaches for reactive power control have become popular [1-7]. But, the

attention has been focused upon power losses and voltage deviation. Relatively, Particle swarm optimization (PSO) as more classic swarm intelligence optimization algorithm has its own characteristics, the reactive power optimization is also applied to different functions [8-10]. PSO as an optimization algorithm based on population, the concept is simple in principle, the search speed, can effectively deal with continuous variable optimization problem. Although PSO has been applied to several power system problems, such as reactive power and voltage control, optimal power flow, dynamic security border identification and state estimation [11-14]. In this paper, Research and analysis of the particle swarm optimization algorithm is applied to reactive power optimization problem solving specific methods and general steps to improve the PSO were mainly used to deal with reactive power optimization problem of discrete control variables and continuous control variables. Finally, the system reactive power optimization tests, so that the improved algorithm is more global optimization capability to verify the feasibility of the method used to solve the optimal allocation of reactive power problem.

II. BASIC PARTICLE SWARM OPTIMIZATION

PSO is a population-based search algorithms, the mathematical description is: Assuming the population in a n-dimensional search space by m particles S , The population is expressed as $S = (X_1, X_2, X_3, \dots, X_m)$, particle i is expressed as $X_i = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{in})$, $i = 1, 2, 3, \dots, m$, this particle into the objective function can calculate its fitness value, which itself experienced the best location in mind that the best position to adapt particle value of $P_i = (P_{i1}, P_{i2}, P_{i3}, \dots, P_{in})$, $i = 1, 2, 3, \dots, m$, Also known as P_{best} , And all the particles in the population experienced the best position denoted by

$P_g = (P_{g_1}, P_{g_2}, P_{g_3}, \dots, P_{g_n})$, $g \in \{1, 2, 3, \dots, m\}$, Also known as G_{best} , Particle i velocity is expressed as $v_i = (v_{i_1}, v_{i_2}, v_{i_3}, \dots, v_{i_n})$, $i = 1, 2, 3, \dots, m$. For each generation, the d dimension of the particle i velocity and position according to the equations as follows:

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1 (P_{id}^k - X_{id}^k) + c_2 r_2 (P_{gd}^k - X_{id}^k) \quad (1)$$

$$X_{id}^{k+1} = X_{id}^k + v_{id}^{k+1} \quad (2)$$

Where c_1 and c_2 are Learning factors, it Usually the number of normal; r_1 and r_2 are both uniformly distributed random number in the range of [0,1]; the k is the number of iterations.

Particle velocity update as in (1) consists of three parts: The first part is the current speed of particles, known as memory items, reflecting the memory capacity of the particles. The second part is the distance between the current position of the particle i and its own position among the best in history, known as self-awareness items, updated to reflect the movement of the particles from their own learning and thinking part of the information that affect their behavior on the particle Next. The third part is the distance between the current position of the particle i and the best position among the population, known as social cognition items, reflecting the sharing of information between the particles and some co-operation. And the speed and position of the particles is usually restricted within a certain range, to prevent the particles unlimited updated beyond the boundary range.

III. IMPROVED ADAPTIVE PARTICLE SWARM OPTIMIZATION

A. Inertia weight particle swarm optimization

Particle Swarm Optimization algorithm can update the particle velocity and position through the iterative calculation from (1) and (2), which has the ability to adjust the global search and local search, but for different problems, different stages of the search require different capabilities, which need to be adjusted in local search and global search algorithm weights to balance local and global exploration and development capabilities. By introducing the inertia weight in the velocity update Equation weight can be attained, thereby updating the Equation then becomes:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (P_{id}^k - X_{id}^k) + c_2 r_2 (P_{gd}^k - X_{id}^k) \quad (3)$$

$$X_{id}^{k+1} = X_{id}^k + v_{id}^{k+1} \quad (4)$$

In (3), the inertia weight ω keep the particles inertia, which determines the current speed of the particle impact velocity of the next moment, acting in the first term of the

inertia of each particle is given the whole space of the expanded search capabilities, namely global search the ability to adjust the size inertia weight ω adjustment algorithm then played a global search and local search ability weights role.

For larger problem space large complex systems, in order to make the searching speed and accuracy to achieve a better balance, common practice is to give it a higher global search capability in the early search algorithm so that it can be appropriate individuals of the population in the late search algorithm gives it a higher local search ability to improve its performance optimization and convergence precision. So, at different stages of setting different algorithms inertia weight to make adjustments, giving greater inertia weights to strengthen the global search in early iterations, reducing inertia weight value in later iterations, to improve the local search ability and convergence precision.

In the early research applications, the use of more linear decreasing inertia weight strategy, that inertia weight algebraic ω decreases with an increase in the linear evolution, usually linearly decreasing from 0.9 to 0.4. Experiments show that this strategy compared to the original optimization method has obvious improvement, but for solving complex problems, the effect is not very obvious, and the actual search process PSO algorithm is relatively complex and non-linear, linear decreasing inertia weight strategy is difficult to reflect the actual search optimization process, which affects the search performance of the algorithm. Since then appeared random inertia weight strategy, the fuzzy controller dynamic inertia weight strategies and adaptive inertia weight strategy and some inertia weight change strategy study, to a certain extent, improve the PSO algorithm performance.

B. Improved Adaptive Particle Swarm Optimization

In particle swarm iterative process, based on the convergence requirements, speed particle populations will increase with the number of iterations decreases, if the speed particle populations close to zero, in the face of population under optimal circumstances, the population of particles will focus on the optimal position and stop moving, the diversity of the population of particles will gradually be lost, resulting algorithm may eventually fall into local optima.

Through the study of particle swarm optimization principles in the analysis of the speed of the particle populations on the basis of the impact on the algorithm, we use to dynamically change its evolution inertia weight method based on population particle velocity, it has adaptive strategy can effectively maintain the diversity, balancing algorithm global search and local search capabilities population, improve the adaptability of the algorithm, which converges to avoid local optima.

By particle swarm particle velocity and position of populations to continuously update the iterative search for the optimal solution strike problem, the speed populations of particles in each iteration are changing, the population of particles in the first generation of average speed k can be expressed as :

$$v_a^k = \left(\sum_{i=1}^m \sum_{j=1}^n |v_{ij}^k| \right) / (m \times n) \quad (5)$$

Where v_a is the average speed of the population of particles, v_{id} is the population of particles in each dimension of the individual speed, m is the number of populations of particles, n is the dimension of the individual particles, k is the number of iterations.

For the population of the particle velocity, if the velocity of the particles is large, it means it has a strong global search capability, if the speed of particles smaller, then the particles have a strong local search ability. In this paper inertia weight change strategy as follows: Firstly, the speed set point population each generation of particles, then the speed value to compare the actual population of particles, depending on the size of the change in the value of the dynamic inertia weight. When the speed of the actual population of particles is less than the set value, increasing the value of the corresponding inertia weight; particle velocity when the actual population is greater than the set value, reducing the value of the corresponding inertia weight, specific practices can be implemented in accordance with the following Equation:

$$\omega(k+1) = \begin{cases} \omega(k) \times r, v_a^k < v_e^k \\ \omega(k) / r, v_a^k > v_e^k \\ \omega(k), v_a^k = v_e^k \end{cases} \quad (6)$$

Where v_a is the average speed of the actual population of particles, v_e is the population of particles speed set point, r is weight coefficient variation of inertia weight, ω is inertia weight, k is the number of iterations.

Population particle velocity setpoint in accordance with the following Equation index variations:

$$v_e^k = v_0 e^{-(2k/(T_{\max} - T_0))^2} \quad (7)$$

Where v_e set value for the population particle velocity, v_0 is set to the initial population of the particle velocity, T_{\max} is the maximum number of iterations, T_0 for local search algebra, k is the number of iterations.

In an iterative process, the range of ω is $[\omega_{\min}, \omega_{\max}]$, when the range is outside the boundary, the corresponding boundary value. The initial inertia weight is taken as ω_{\max} , in order to strengthen the capacity of local search algorithms in later iterations within the T_0 - generation iteration, the inertia weight is taken as ω_{\min} .

Inertia weight variation coefficient r values to the right, if the value is too large, the inertia weight variation is large, the speed of change in the population also increased, increasing the diversity of the population of

particles, but to reduce its local search capability; if the value is too small, the inertia weight less weight change, population change in velocity is reduced accordingly, population increase local search capability, but the diversity of population decrease. According to the experimental analysis, r take 1.05 more appropriate.

IV. SPECIFIC OPERATIONS IMPROVED ADAPTIVE PARTICLE SWARM OPTIMIZATION OF REACTIVE POWER

Improved adaptive particle swarm optimization for reactive power optimization problem to solve, first population of particles that reactive power optimization control variables and the parameters are initialized, the trend calculations by the fitness value of each population of particles is calculated according to the objective function, and Depending on the size of individual particles updated population and global best position to adapt to the optimal value, and then update the population according to the formula particle velocity and position as well as the value of inertia weight, repeated iterations until it reaches the end of the convergence condition optimization, optimal control variable.

A. Individual particle structure

The location populations of individual particles of reactive power optimization of control variables in this article reactive power optimization control variable for the generator terminal voltage, load tap transformer ratio and switching shunt capacitors to optimize the characteristics of the algorithm based on particle swarm, in this paper, to optimize the value, the dimension of individual particles is continuously controlled continuously variable control variable number generator terminal voltage.

B. Population particle initialization

Depending on the reactive power optimization problem to determine the number of variables that control the dimension of the particle population, complexity and general experience combined with the problem of determining the size of the population. PSO selected examples in this article in the population size is 20. Of the population of particles is initialized, initialize the initial value should be evenly distributed as possible, in favor of a global optimization algorithm, and fast convergence. Using (8) randomly generated initial value of the population:

$$x_{id} = x_{id}^{\min} + rand() * (x_{id}^{\max} - x_{id}^{\min}) \quad (8)$$

Where x_{id} is the value of the population of particles i in the d -dimensional control variables, x_{id}^{\max} and x_{id}^{\min} are the upper and lower limits, $rand()$ is a random number uniformly distributed within the range of $[0,1]$.

C. Fitness function

Fitness function is to evaluate the merits of the main basis for the particle population, according to reactive power optimization problem to determine the objective function, usually active network systems loss minimum objective function. By tide system calculated the network variables such as voltage and power, active power loss can

get the system to obtain the objective function value as well as the fitness function values.

D. Individual particle evaluation and optimal value update

For individual particle population, calculated by the trend of individual fitness value, according to the fitness value of individual particles were evaluated. First, the fitness value of individual particles optimum fitness value of its historical comparison, if the individual particles in this iteration better fitness value, the update history of the individual particles of optimum fitness value and the best location P_{best} . After all in this iteration particle populations have completed the above calculations, all the individual particles of the population already have this optimal fitness value of individual history, the history of individual particles in all populations and population optimal fitness value of particles optimal fitness value comparison, if you have better fitness value, the update history optimal particle populations and the fitness value of the particle population optimal position G_{best} .

E. Updating the velocity and position of individual particles and the boundary set

Speed position individuals of the population of particles according to the update (3) and (4) to carry out, while the velocity and position of the boundary is set, and to prevent the control variables beyond the scope of the high speed flying over the optimal solution. Boundary position is in the range boundary of each control variable, ranging from $[x_{min}, x_{max}]$. The maximum speed is set to v_{max} , the velocity in the range of $[-v_{max}, v_{max}]$. Maximum speed setting affects performance optimization particles, if v_{max} too large, the particles may fly optimal solution; If v_{max} too small, the particles may not be a global search, easy to fall into local optimum, usually the control variable of v_{max} is set to the value of 20% -30% range. If the individuals of the population of particles outside the range of speed and position in an iterative process, it is defined on the corresponding boundary.

F. Adaptive change Of inertia weight

Inertia weight changes by the (6) based on the average speed of the particle populations in the iterative process of dynamic change, ranged [0.4-0.95]. The initial inertia weight is taken as ω_{max} , in solving the pre maintain a high inertia weight, so that the population of particles has a strong global search capability, with the increase of iteration, the inertia weight decreased, the particle velocity gradually declined, after the iterative algorithm within T_0 generations, the inertia weight is set to the minimum, to strengthen the local search ability populations particles.

V. CONCLUSION

The basic principles of particle swarm optimization algorithm is dealt with in this paper, applied research and analysis of the reactive power optimization problem. Particle swarm optimization algorithm based on the

specific circumstances and characteristics of reactive power optimization, optimization algorithm is proposed based on improved adaptive particle swarm optimization for solving reactive power program, to a certain extent, improve the search performance optimization solution and convergence speed. According to PSO handling characteristics of the variable capacity, particle swarm optimization algorithm applied to a continuous emphasis on control variables to optimize regulation, reactive power optimized for efficient operation provides a practical and reliable foundation.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contribution of co-workers and reviewers' comments.

REFERENCES

- [1] Y. Li, Y.J. Cao, Z.Y. Liu, Y. Liu and Q.Y. Jiang, "Dynamic optimal reactive power dispatch based on parallel particle swarm optimization algorithm," *Computers and Mathematics with Applications*, vol. 57, pp. 1835-1842, Dec. 2009.
- [2] O.M.Nezami, A.Bahrampour and P.Jamshidlou, "Dynamic Diversity Enhancement in Particle Swarm Optimization Algorithm for Preventing from Premature Convergence," *Procedia Computer Science*, vol. 24, pp. 54-65, Aug. 2013.
- [3] H.Gao, S.Kwong, J.J.Yang and J.J.Cao, "Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation," *Information Sciences*, vol. 250, pp. 82-112, Nov. 2013.
- [4] W.H.Lim and N.A.M.Isa, "Two-layer particle swarm optimization with intelligent division of labor," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2327-2348, May 2013.
- [5] H.W.Dommel and W.F.Tinney, "Option Power Flow Solution," *IEEE Trans.on Power Apparatus and Systems Magazine*, vol. 87, pp. 1866-1875, Dec. 1968.
- [6] H.Weil, H.Sasaki and R.Yokoyama, "An Application of Interior Point Quadratic Programming Algorithm to Power System Optimization Problems," *IEEE Trans. on Power Systems*, vol. 11, pp. 260-266, Nov.1996.
- [7] K. N. Reddy and V. Agarwal, "Utility Interactive Hybrid Distributed Generation Scheme With Compensation Feature," *IEEE Trans. on Energy Conversion*, vol.22, pp. 666-673, Dec. 2007.
- [8] K. R. C. Mamandur and R. D. Chenoweth, "Optimal Control of Reactive Power Flow for Improvements in Voltage Profiles and for Real Power Loss Minimization," *IEEE Trans. on Power Apparatus and Systems*, vol. 100, pp. 3185-3194, Apr. 1981.
- [9] D.P.Bruns, G.R.Newcomb and S.A.Jr Miske, "Shunt capacitor bank series group shorting (CAPS) design and application," *IEEE Trans. Power Delivery*, vol. 16, pp. 24-32, May. 2001.
- [10] K. Aoki, M. Fan and A. Nishikori, "Optimal VAR Planning by Approximation Method for Recursive Mixed-Integer Linear Programming," *IEEE Trans. on Power Systems*, vol. 3, pp. 1741-1748, Apr. 1988.
- [11] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc IEEE Int Conf Neural Networks*, vol. 24, pp. 1942-1950, Dec. 1995.
- [12] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans Power Syst*, vol. 15, pp. 1232-1341, Apr. 2000.
- [13] M. A. Abido, "Optimal power flow using particle swarm optimization," *Int J Electr Power Energy Syst*, vol. 24, pp. 563-571, Nov.2002.
- [14] I. N. Kassabalidis, M. A. El-Sharkawi, R. J. II Marks, L. S. Moulin and A. P. Alves da Silva, "Dynamic security border identification using enhanced particle swarm optimization," *IEEE Trans Power Syst*, vol. 17, pp. 723-732, Nov.2002.