

## Optimizing Available Transfer Capability Based on Chaos Cloud Particle Swarm Algorithm with Golden Section Criteria

Hongsheng Su<sup>1,2</sup> and Shuangshuang Wang<sup>1</sup>

<sup>1</sup>*School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China*

<sup>2</sup>*Key Laboratory for the Highland Traffic Information Engineering and Control in Gansu province, Lanzhou 730070, China  
shsen@163.com*

### Abstract

Available transfer capability (ATC) is an important index to determine power delivery contracts and manage transmission congestion. To aim at the flaws of low accuracy and premature convergent in each ATC power flow optimization algorithm, in this paper, an optimal power flow model is proposed for ATC under the static security constraints, which is called as chaos cloud particle swarm optimization algorithm based on golden section evaluation criteria (CCGPSO), wherein the active power maximum of all load nodes in receiving area is taken as objective function, and AC flow current method is applied for power flow calculation. To balance the local and global searching ability of particle swarm, golden section judge principle is introduced to divide the full particles into three sub-swarms according to their fitness level, that is, standard particles swarm with the higher fitness values, and chaos cloud particles swarm with lower fitness values, and as well as cloud particles swarm with moderate fitness values, and each sub-swarm is operated and updated by different algorithms, that is, for the standard particles, the basic PSO is applied to perform mode updating, and for the chaotic particles, fine search in a certain range is used in the neighborhood of the current optimal solution after each step of chaotic search, and for the cloud particles, X conditional cloud generator is used to adapt inertia weight, dynamically. The algorithm effectively solves the flaws of easily falling into local optimum in basic PSO and large amount of calculation in chaos optimization, and can make entire particle swarm search for all solution spaces. Compared with the simulation results of cloud PSO and chaos PSO algorithms in IEEE-30-node test system, the investigations show that CCGPSO is quite effective, and more suitable to solve large-scale non-linear multi-constraint optimization problems.

**Keywords:** Available Transfer Capability (ATC), particle swarm optimization (PSO), golden section, Chaos, Cloud

### 1. Introduction

Power transmission grids available transfer capability (ATC) is not only one of the important parameters to influence power transactions, but also is a significant technical indicator to monitor the power grid operation status. ATC is denoted as the remaining transfer capacity to be able to be used for commercial transactions on the basis of existing transmission contracts in the practical physical transmission grids [1]. Modern electrical power systems have already developed into large-scale AC-DC interconnected systems, and power transactions are becoming more and more frequent, such that higher demands require to be ensured on power quality, stability and economic operation of power grids. Under the condition that the premise of system safety operation is guaranteed, the regional power grid companies want to take advantage of the existed transmission network to deliver more power to achieve the optimized allocation of global resources.

Hence, for a large-size complex interconnected power system, it is of quite significance to better coordinate the relationship between the security and economic benefits.

The optimizing algorithm of the existing ATC power flow can be divided as the two types: one is the probabilistic algorithm, and the other is the deterministic one. The former mainly applies probabilistic theory and statistics analysis to determine the ATC in power transmission systems [2], whose main advantageous lies in that many uncertainties can be taken into account, but it is difficult for those power systems with its states continuously changing or random faults frequently occurring. The later calculates transmission capacity between two regions as the system is being in a certain operating state[3]. Deterministic ATC methods can meet the rapid and accurate calculating requirements under electricity market environments, and mainly includes DC power flow, and continuous power flow, and as well as optimal power flow method, and *etc.* [4]. Since optimal power flow method possesses stronger processing capacity for the constraints, it can describe ATC issue into a pure nonlinear programming mathematical problem.

To improve accuracy and speed of ATC calculation, many scholars have proposed many methods to implement the improvement on ATC algorithm, and obtain some achievements. In [5], a continuous power flow(CPF) method is proposed to calculate ATC, which effectively overcomes the shortcomings not to consider reactive power flow, and voltage limit, and voltage collapse, but the results acquired are often too conservative due to CPF changes only along the fixed direction. In [6], genetic algorithm (GA) is proposed to establish ATC model, although the crossover operation of GA can improve the global searching ability, but the variation operation increases the diversity of population so that the ability of local searching is affected. In [7], an ATC estimation is proposed based on chaos interior point algorithm, which improves the global searching ability and convergence speed, but the chaos algorithm only searches local solutions, repeatedly. In [8], ATC model is constructed based on improved artificial fish algorithm (AFSA), which effectively overcomes the premature convergence shortcomings by introducing genetic operator to improve the convergence performance of standard AFSA. Clearly, different algorithms possess their respective advantages and disadvantages, and the algorithm to apply single search mechanism is simple, but more difficult to achieve efficient optimization, such that there are some defects existed in processing some large-scale multi-constraint non-linear optimization problems, such as robustness, and easy to fall into local optimum, which leads to that the calculation results are of too conservative and a serious waste of power resources. Therefore, to integrate different algorithms together and take each advantage is an inevitable development trend in solving ATC problem.

In this paper, we present a chaotic cloud particle swarm algorithm [9], and then it is applied to solve ATC issue from optimal power flow model under static security constraints[10]. Firstly, based on particle swarm algorithm (PSO), we apply the ergodicity feature of chaotic optimization to search entire spaces to escape from falling into local extremum. Then, according to particle swarm fitness value, we apply golden section grouping criteria to divide the whole particle swarm into three sub-swarms, that is, standard particles, chaotic cloud particles, and cloud particles. To aim at every sub-swarm, different processing operation and update mode is implemented, which not only reduces the calculation redundancy, but also increases in optimal speed and precision. Finally, the chaotic cloud particle swarm algorithm based on the golden standard is simulated in IEEE-30 node system to calculate available transfer capability, and the simulation results show that the proposed algorithm is more suitable for processing such large scale nonlinear optimization problems.

## 2. ATC Mathematical Model

The investigations aim on transfer capability is to calculate the maximum power which can be transmitted in electricity trading among specific areas on the basis of power flow. Given system operating state, ATC refers to that the maximum transmission increment that does not violate the system operational limits, that is, the existing power transaction of the system is not affected, which considers that the active power output of the generators in non-sending area and the loads in non-receiving area keep constant, the sending area generator active output and receiving area active load increase at the same time. Clearly, the mathematical model of ATC is a nonlinear continuous optimization problem. In this paper, we adopt optimal power flow model based on the static security constraints, and take active output maximum of all load nodes in receiving area as the objective function, which can be described by

$$A_{ATC} = \max(\sum_{i \in \Gamma_{lm}} \Delta P_{Li}) \quad (1)$$

And then equal constraints can be expressed by

$$\begin{cases} P_{Gi} - P_{Li} - V_i \sum_{j=1}^n V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0 \\ Q_{Gi} - Q_{Li} - V_i \sum_{j=1}^n V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) = 0 \end{cases} \quad (2)$$

And inequal constraints can be presented by

$$\begin{cases} P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \\ Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \\ P_{Li}^{\min} \leq P_{Li} \leq P_{Li}^{\max} \\ Q_{Li}^{\min} \leq Q_{Li} \leq Q_{Li}^{\max} \\ V_i^{\min} \leq V_i \leq V_i^{\max} \\ -\overline{S}_{ij} \leq S_{ij}^E \leq \overline{S}_{ij} \end{cases} \quad (3)$$

In (1),  $\Delta P_{Li}$  represents load active increment of the node  $i$ , and  $\Gamma_{Lm}$  represents receiving area load nodes collection. In (2) and (3),  $P_{Gi}$ ,  $Q_{Gi}$  are active and reactive power of the generator, and  $P_{Li}$ ,  $Q_{Li}$  represent active and reactive power of load nodes, and  $V_i$ ,  $\theta_i$  are the nodes voltage magnitude and phase angle, and  $S_{ij}$  is transmission line heat stability constraint from endpoint  $i$  to  $j$ .

## 3. PSO Algorithm

Particle Swarm Optimization (PSO) is one kind of evolution technology based on swarm intelligence behaviors, and has been broadly applied in power system active power distribution and state estimation [11], and the reactive power optimization and voltage control [12], and *etc.* PSO algorithm is derived from birds foraging in the process of migration and swarm behavior, wherein each optimization problem can be viewed as a bird in the search space, and each bird flies by a certain speed in the search space, and the speed can be dynamically adjusted by the individual flight and group flight experience. In searching space, the birds are abstracted as particles without quality and volume, whereas each particle remembers and follows the current optimal particle in solution space. Each particle's fitness value is determined by objective function.

For each iteration, each particle updates its velocity and position according to the following formula.

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (4)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (5)$$

where  $X_i=(X_{i1},X_{i2},\dots,X_{in})$  is the current position of particle  $i$ , and  $V_i=(V_{i1},V_{i2},\dots,V_{in})$  is the current speed of particle  $i$ , and  $P_i=(P_{i1},P_{i2},\dots,P_{in})$  is the experienced optimal position of particle  $i$ , and  $P_g=(P_{g1},P_{g2},\dots,P_{gn})$  is the optimal position of whole particle swarm, and  $\omega$  is inertia weight coefficient, and  $c_1, c_2$  are accelerated factors and generally set from 1.5 to 2.0 according to experience, and  $r_1, r_2$  are uniformly distributed random numbers in range of zero to one. Let  $f(x)$  be the minimum of the aim function, then the optimal position  $P_i$  of the particle  $i$  and  $P_g$  of the full particle swarm are determined by the following formula.

$$P_i = \begin{cases} P_i, f(P_i) < f(X_i(t+1)) \\ X_i(t+1), f(P_i) \geq f(X_i(t+1)) \end{cases} \quad (6)$$

$$P_g \in \{P_0, P_1, \dots, P_m\} \} f(P_g) \\ = \min \{f(P_0), f(P_1), \dots, f(P_m)\} \quad (7)$$

Seen from (4), we can know that the first item on the right of equation is the particle's speed before updating, it reflects the particle's memory ability. The second item is the cognition item, and it represents the distance between current position and its optimum particle position. The third item is the social part, and presents a distance between current position and the best position of swarm, and reflects the information sharing and mutual cooperation among groups. Inertia weight  $\omega$  largely determines the balance between PSO local search ability and the global search ability, such that it generates great influence on the convergence of the algorithm. Concretely speaking, the larger  $\omega$  is, the stronger global search ability is, and the weaker local search ability is, and conversely otherwise.

#### 4. Chaos Optimization Algorithm

Chaos is a universal phenomenon in nonlinear systems, from macroscopic view, which shows chaotic, random, and is very sensitive to initial conditions, but from microscopic observation, it possesses ordered regularity. The chaotic motion possesses the characteristics of ergodicity, and randomness, and regularity, such that it can traverse all states and do not repeat in a certain range according to its own laws. Since chaos method does not require optimization problem to meet the conditions of continuity and differentiability, and can traverse all states, in a certain extent it can overcome the shortcomings of traditional optimization methods.

To generate chaotic phenomenon, the logistic model is introduced into standard particle swarm below.

$$x_{i+1} = \mu x_i (1 - x_i) \quad (8)$$

where  $\mu$  is called logistic parameter with the scope of [3.57,4], and as  $x_i$  belongs to (0, 1) and inequals to 0.25, and 0.5, and 0.75, the system is in the chaotic region, particle trajectories exhibit chaotic characteristics[13]. For  $x_i \in (a_i, b_i)$  beyond the scope of the taking value above we can then perform a round trip mapping by

$$cx_i = (x_i - a_i) / (b_i - a_i) \quad (9)$$

$$x_i = a_i + cx_i \cdot (b_i - a_i) \quad (10)$$

Thus we can apply chaotic particle swarm global ergodic characteristic to make the whole particle swarm search all solution spaces not to stay in local optimum of the extreme point by letting each particle perform chaos ergodic motion after every time iteration.

## 5. Cloud Model Algorithm

The cloud model is an uncertainty conversion model expressed by qualitative knowledge and quantitative expression, and its essence is the concept of cloud droplets, and it better reflects the relationship between the randomness and fuzziness. Let  $U$  be a quantitative domain represented by numeric value, and  $C$  be a qualitative concept of  $U$ , if the quantitative value  $x \in U$  is a random realization of qualitative concept  $C$ , and the certainty degree of  $x$  to  $C$  is a random number possessing some stable tendency, and then distribution of  $x$  in the field  $U$  is called cloud, and recorded as cloud  $C(x)$ , wherein every  $x$  is called a cloud drops [14]. As  $U(x)$  follows normal distribution, it is called normal cloud model, the whole concept expressed by it can be represented integrally by the expected value  $E_x$ , entropy  $E_n$  and hyper entropy  $H_e$ . Expected value  $E_x$  means distribution of cloud droplets in the domain space, and usually, its  $x$  value corresponds gravity center of cloud. Entropy  $E_n$  is a measure of the ambiguity of the qualitative concept, and it reflects the numerical-value range that can be accepted by concept. The hyper entropy  $H_e$  is dispersion degree of entropy. Given a group of cloud droplet as samples  $(x_i, m_i)$  in accordance with normal distribution, the algorithm produces three digital characteristics  $(E_x, E_n, H_e)$  that are described by the cloud theory. As three-eigenvalue is given, it is called X conditional clouds generator, whose generation steps are as follows.

(1) Generate  $E_{ni}' = G(En, He)$ , where  $E_{ni}'$  is a normal random number of its expectation value being  $En$  and standard deviation being  $He$ ;

(2) Generate  $x_i = G(Ex, En)$ , where  $x_i$  is a normal random number of its expected value being  $Ex$  and standard deviation being  $En$ ;

(3) Compute

$$\mu_i(x_i) = \exp\left[-\frac{(x_i - Ex)^2}{2(E_{ni}')^2}\right] \quad (11)$$

and  $(x_i, \mu_i(x_i))$  is then defined as a cloud drop.

(4) To repeat the steps above, until the adequate cloud drops emerge.

## 6. CCGPSO Algorithm Based on Golden Section Criteria

In basic PSO algorithm, although random initialization process can ensure the uniform distribution of initial solutions, it can not guarantee the quality of individual particles, such that one part particles fly away from the initial solution. Along with all particles fly toward the optimal solution in the PSO optimization process, more and more particles lose diversity so that the later convergence speed becomes quite slow. To resolve the issue, we introduce chaos into the particle swarm algorithm. Firstly, a set of chaotic variables is produced with the same number of optimized variables, and then chaos is introduced into optimization variables to appear chaotic state similar to carrier, at the same time ergodic range of chaotic motion is enlarged to the range of optimization variables, then chaotic variables can be directly used to search optimal solution.

However, a lot of chaos transformation and inverse transformation would greatly increase the calculation amount such that the calculation speed of the algorithm becomes very slow. Consider that the cloud algorithm possesses good global search capability, and production process of cloud generator is fast and convenient, hence, in this paper, chaotic particle swarm optimization algorithm is combined with cloud theory, and the golden section evaluation criteria is introduced, simultaneously, which is based on fitness evaluation criteria [15]. According to the fitness value, particle is divided into three parts, each part conducts different processing operation and updating mode to improve the algorithm optimizing speed and accuracy, and simultaneously it can overcome the redundancy problem of chaos algorithm to the maximum extent.

Firstly, we calculate the average fitness value  $f_{avg}$  of all particles, and find out the optimal fitness  $f_{min}$ , and the worst fitness  $f_{max}$ , and two assessment requirements  $f_{golden1}$  and  $f_{golden2}$  are established in accordance with the golden section, that is

$$f_{avg} = \frac{1}{n} \sum_{i=1}^n f_i \quad (12)$$

$$f_{range} = f_{max} - f_{min} \quad (13)$$

$$f_{golden1} = 1.618 f_{avg} \quad (14)$$

$$f_{golden2} = f_{min} + f_{range} / 1.618 \quad (15)$$

The golden section evaluation criteria are described by

$$f_1 = \begin{cases} f_{golden1}, f_{golden1} \geq f_{golden2} \\ f_{golden2}, f_{golden1} < f_{golden2} \end{cases} \quad (16)$$

$$f_2 = \begin{cases} f_{golden1}, f_{golden1} < f_{golden2} \\ f_{golden2}, f_{golden1} \geq f_{golden2} \end{cases} \quad (17)$$

The particles whose fitness value are higher than  $f_1$  are called standard particles, they are closer to the optimal solution, and so they can update their speed and displacement follow the basic particle swarm algorithm. The particles whose fitness value are lower than  $f_2$  are called chaos cloud particles, since they are far from the optimal solution, such that they need to be operated by chaotic particle swarm calculation first and then cloud calculation to traverse all solution spaces. The particles whose fitness value between the two populations above are called cloud particles, their distance from the optimal solution are moderate, and thus we use X condition cloud generator to make each particle dynamically adapt inertia weight as follows.

$$E_x = f_1 \quad (18)$$

$$E_n = (f_1 - f_{min}) / c_1 \quad (19)$$

$$H_e = E_n / c_2 \quad (20)$$

$$E_n' = normrnd(E_n, H_e) \quad (21)$$

$$\omega = 0.9 - 0.5 \times e^{\frac{-(f_i - E_x)^2}{2(E_n')^2}} \quad (22)$$

The step of the CCGPSO is below.

1) To input the original data, and obtain the system nodes and branches information, and acquire the range of the control variables, and set the maximum number of iterations.

2) Initialize population: To generate particle swarm and its initial position and velocity randomly, to get individual optimal value and global optimal value by calculating each particle's fitness in current position.

3) Chaos initialize population: to randomly generate  $n$  vectors range between 0 to 1, that is  $x_1=(x_{11},x_{12},\dots,x_{1n})$ , according to the logistic equation  $x_{i+1}=\mu x_i(1-x_i)$  to obtain  $n$  chaos variables, and the chaos variables range is enlarged to the corresponding range of the optimization variables.

4) To perform flow calculation for each initialization particle, respectively, wherein balance node is selected in sending area, to obtain state variables of the system to solve the fitness value, and then to get individual extremum and global extremum.

5) To calculate average fitness value  $f_{avg}$  of the entire particle swarm, and establish two evaluation rules  $f_{golden1}, f_{golden2}$ , based on the golden section evaluation criteria, the particle swarm is divided into three parts: standard particles, cloud particles, chaotic cloud particles. If the particle belongs to the standard particle, and then turn to step 6), and turn to step 7) for chaos cloud particle, and turn to step 8) otherwise.

6) To update the displacement  $x_{id}^{k+1}$  and velocity  $v_{id}^{k+1}$  of the particle, and set inertial parameter  $\omega$ , and amend speed and displacement of cross-border particles state variables in accordance with the standard particle swarm.

7) To perform chaos optimization for optimal location  $P_g$ . Applying the formula  $x_i=(x_{11},x_{12},\dots,x_{1n})x_i=(p_{gi}-a_i)/(b_i-a_i)$  to map the optimal position to the defined domain [0,1] of logistic equation where  $a_i$  and  $b_i$  is the range of optimization variables, and then according to the logistic equation produce  $m$  chaotic variables, after several iterations, each chaotic variables re-transformed into the optimization variables by conducting reverse transformation and obtain the  $m$  variables.

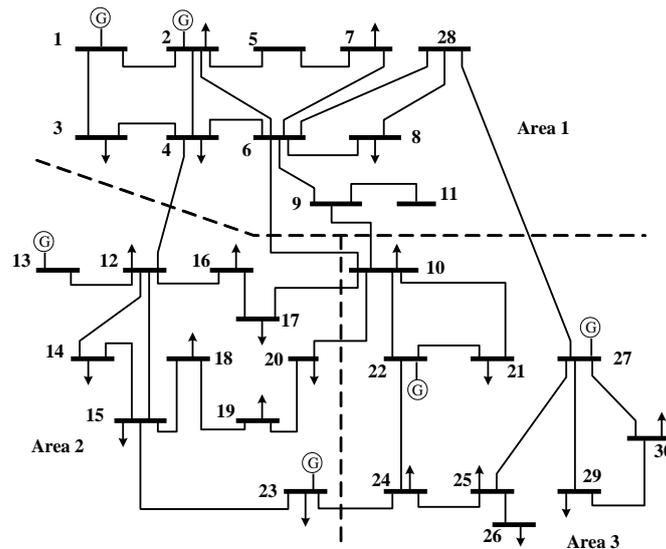
8) To make use of X condition cloud generator to adapt inertia weight dynamically, and displacement and velocity, then calculate fitness.

9) To calculate the fitness value, and compare it with the target value corresponding to the current individual optimal value, and to select the minimum value of the individual optimal solution of all particles as particle swarm global optimal solution  $P_g$ .

10) If the current iterations reach the maximum, and then to turn to the step 9) and output target function value, and otherwise set the iterations  $k=k+1$  and return to step 4).

## 7. Examples Analysis

In this paper, IEEE-30 node system is selected as test object to verify the effectiveness of the proposed method. The system possesses 41 branches, and 6 generators, and as well as 22 load nodes. It is divided into three regions with each region possessing two generators, seven interconnection lines. Figure 1 shows IEEE-30 nodes system, wherein node 1 is a balance node, it is used to balance the whole network power, and the nodes 2, 13, 22, 23, 27 are PV ones. The remaining are PQ nodes.



**Figure 1. IEEE 30-node System**

In this example, the parameters are selected as follows: generation node active power is 100MW, and the original variables take the flow value under basic state. In PSO, particle swarm size is 60, and the maximum iterations is 100, inertia weight  $\omega_{max} = 0.9$ ,  $\omega_{min} = 0.4$ , and learning factors  $c_1=c_2=2.0$ . In chaos algorithm, logistic parameter  $\mu$  equal 4. In cloud algorithm, let global optimal position be expected value  $E_x$ , and current fitness variance  $\delta^2$  be entropy  $E_n$ , and  $E_n/10$  be hyper entropy value  $H_e$ . Table1 shows the calculating results of each optimizing algorithm.

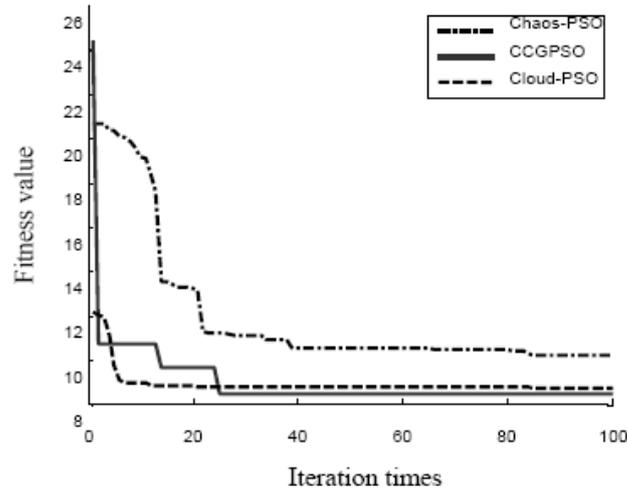
**Table 1. A Comparison Among Three Optimizing Algorithms**

sending area -receiving area	ATC/MW		
	Chaos-PSO	Cloud-PSC	CCGPSO
1-2	103.25	105.52	110.53
2-1	32.34	54.15	57.83
1-3	104.75	102.59	103.07
3-1	53.17	75.01	92.85
2-3	32.21	33.23	44.98
3-2	53.07	63.61	76.54
total	378.79	434.11	485.8

It is noted that the results in the Table1 are average value after 20 times iterations. From Table1, ATC value from region 1 to region 3 by CCGPSO is slightly smaller than the value obtained by cloud particle swarm optimization, but the gross of regional optimal solutions by it is obviously superior to the value acquired from another two methods. This indicates that, for same transmission network as well as the allocation of resources, CCGPSO is able to utilize existing resources to deliver more power, and its advantage is quite prominent. Take ATC from 3 to 1 for an example, practically, the two methods of chaotic particle swarm and cloud algorithm get into premature convergence during ATC calculation.

**Table 2. A Comparison Among Three Optimizing Algorithms from Area 2 to Area 3**

algorithms	ATC <sub>2-3</sub> / MW		
	average value (MW)	sample variance (MW)	iteration times
Chaos-PSO	28.30	7.30	40
Cloud-PSO	30.71	6.27	9
CCGPSO	41.23	3.59	25



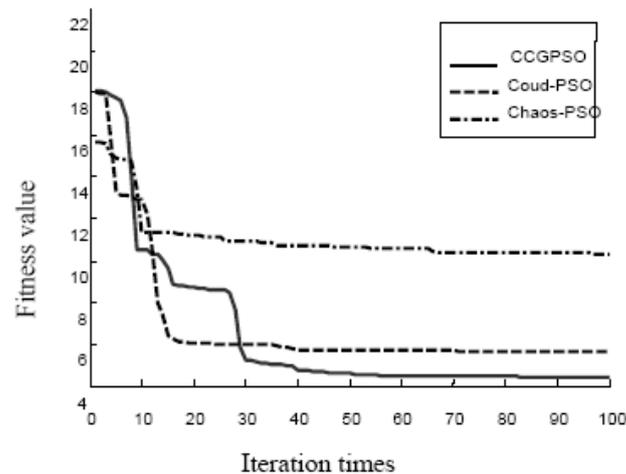
**Figure 2. ATC Convergence Curve from Area 2 to Area 3**

Table 2 lists the related numerical values for ATC calculation from area 2 to 3, and as well as the corresponding convergence properties, whose corresponding simulation curve is shown in Figure 2.

From Table 2, we can know that variance of CCGPSO via 25 times calculation is smaller, which shows that the algorithm possesses strong stability. From the iterations view, CCGPSO requires 25 times iterations, whereas Cloud-PSO is 9 times alone, it is due to the fusion of three optimization algorithms that makes CCGPSO possesses better global search and the ability to overcome local extremum so as to increase the calculation amount, and that the iterations is slightly extended.

**Table 3. A Comparison Among Three Optimizing Algorithms from Area 2 to Area 1**

algorithms	ATC <sub>2-1</sub> / MW		
	average value (MW)	sample variance (MW)	Iteration times
Chaos-PSO	32.31	11.13	10
Cloud-PSO	49.78	10.66	38
CCGPSO	54.43	5.67	30



**Figure 3. ATC Convergence Curve from Area 2 to Area 1**

Table 3 lists the related numerical of ATC from regions 2 to 1, and the corresponding convergence properties curve is shown in Figure 3.

From Table 3, we can see that variance of CCGPSO is smaller than the one of the others. This shows that the algorithm has stronger stability. CCGPSO has more long iterative time than chaos-POS, this is because in search later period particle swarm is divided into three populations, and the operation process becomes slightly complicated so as to make iterations increase. However, CCGPSO obtains maximum average value through 30 times calculations, this indicates that CCGPSO possesses higher global search ability and good accuracy.

The convergence curves of three algorithms in Figure2 and Figure3 show that, at the beginning of iteration, CCGPSO has larger slope, which indicates that CCGPSO algorithm has continued and rapid convergence ability, but after a while it trapped into local extremum. In this moment the introduction of golden section grouping criteria make particle swarm quickly jump out of local minimum, and start to group optimization, and each population continue to search optimal solution along the direction of the global optimal solution. Although this process elapses time slightly long, the convergence value is significantly better than the Cloud-PSO algorithm and Chaos-PSO algorithm, this also shows it possesses higher accuracy. Cloud-PSO converges faster, less the number of iterations, but the horizontal part of the line in Figures shows that search entries into a phase of stagnation, *i.e.*, easily into a precocious, optimization accuracy is inferior to CCGPSO. The convergence extremum from Chaos-PSO algorithm is far less than CCGPSO, but it has lower accuracy.

## 8. Conclusion

The chaos cloud particle swarm optimization algorithm based on the golden section evaluation criteria presented in this paper effectively overcomes the flaws of traditional PSO such as premature and slow convergence, such that the capabilities to searching optimization are dramatically improved in ATC, which promotes its further applications. The method proposed firstly divides the total particle swarm into three-subswarm applying golden section principle based on their fitness values, that is, standard particle subswarm, and chaos-cloud particle one, and as well as cloud particle one, and then implements different operations and updating modes for each sub-population. Namely, for the standard particles, the basic PSO is applied to perform velocity adjustment and position updating, and for the chaotic particles, the fine search in a certain range is used in the neighborhood of the current optimal solution and then cloud calculation to traverse all

solution spaces, and for the cloud particles, X conditional cloud generator is used to adapt inertia weight, dynamically. This algorithm can effectively balance the relationship between the global and the local searching, and solves the problems of easily falling into local optimum in basic PSO, and as well as repeatedly search part of solutions in chaos optimization, and is verified in IEEE 30-node system. The algorithm proposed in this paper possesses higher accuracy in ATC calculation and makes full use of power resources, such that it possesses more prominent practical significance in the power system ATC optimization.

## Acknowledgments

This work was financially supported by the Key Laboratory for the Highland Traffic Information Engineering and Control in Gansu province of China, under the Scientific Project: “The Dynamic Reliability Evaluation and Predictive Maintenance Strategy of EMU Brake Based on Multivariate Integrated Random Walking”; No. 20161104.

## References

- [1] H. Haitao, Z. Hua and Z. Lizi, “Available transfer capability research based on modified particle swarm optimization algorithm”, Proceeding of the CSEE, vol. 26, no. 20, (2006), pp. 45-49.
- [2] G. Yajing, Z. Ming and L. Gengyin, “Available transfer capability based on Markov chain and enumeration method”, Proceeding of the CSEE, vol. 26, no. 19, (2006), pp. 41-46.
- [3] W. Cuiru, Y. Hejin and H. Zhiqiang, “A modified particle swarm optimization algorithm and its application in optimal power flow problem”, September; the Fourth IEEE International Conference on Machine Learning and Cybernetics, Guangzhou, China, (2005).
- [4] S. He, J. Y. Wen and E. Prempain, “An improved particle swarm optimization for optimal power flow”, PowerCon 2004, Singapore, (2004) December.
- [5] G. C. Ejebe, J. G. Waight, M. Santos-Nieto and W. F. Tinney, “Fast calculation of linear available transfer capability”, IEEE Transactions on Power Systems, vol. 15, no. 3, (2000), pp. 1112-1116.
- [6] B. Mozafari, A. M. Ranjbar and A. R. Shirani, “A comprehensive method for available transfer capability calculation in a deregulated power system”, Proceedings of IEEE International Conference on Electric Utility Deregulation, Hong kong, China, (2004) May.
- [7] H. Xin and L. Guoqing, “Calculation of available transfer capability based on chaos-interior point algorithm”, Power System Technology, vol. 35, no. 11, (2011), pp. 130-135.
- [8] Z. Hua, L. Wei, Z. Lizi, Y. Jun and H. Hongwei, “Available transfer capability calculation based on modified artificial fish school algorithm”, Power System Technology, vol. 32, no. 5, (2008), pp. 84-88.
- [9] Z. Chaolong and Y. Chunri, “Particle swarm optimization algorithm based on chaos cloud model”, Journal of Computer Applications, vol. 32, no. 7, (2012), pp. 1951-1954.
- [10] W. Feng and B. Xiaoming, “Optimal power flow based transfer capability calculation”, Proceeding of The CSEE, vol. 22, no. 11, (2002), pp. 35-40.
- [11] Y. Xiaohui, W. Cheng and Z. Yongchuan, “Particle swarm optimization in the application of power system”, Power System Technology, vol. 28, no. 19, (2004), pp. 14-19.
- [12] W. Chao, “The research on the power distribution network reactive power planning based on the chaotic hydrometeor particles swarm algorithm”, Shanghai Jiao Tong University, (2011).
- [13] L. Junmin and G. Yuelin, “Chaos particle swarm optimization algorithm”, Computer Research and Development, vol. 28, no. 2, (2008), pp. 322-325.
- [14] L. Deyi, M. Haijun and S. Xuemei, “Membership cloud and membership cloud generator”, Computer Research and Development, vol. 32, no. 6, (1995), pp. 15-20.
- [15] W. Xingqiong, Z. Yongquan, H. Huajuan and L. Dexiang, “Adaptive particle swarm optimization algorithm based on cloud theory”, Computer Engineering and Applications, vol. 45, no. 1, (2009), pp. 48-50.

## Authors



**Hongsheng Su**, obtained his Master in Traffic Information Engineering and Control, Lanzhou Jiaotong University in 2001. He acquired his PhD in Power Systems and Its Automation, Southwest Jiaotong University. Now he is serving as a full-time professor at school of Automation and Electrical Engineering, Lanzhou Jiaotong University. His research interest includes System Security and Reliability, Intelligent Control, Power Systems and Its Automation, and *etc.*



**Shuangshuang Wang**, obtained his Bachelor in Electrical Engineering and Automation, Lanzhou Jiaotong University. Now she is serving as a full-time graduate student at school of Power Systems and Its Automation, Lanzhou jiaotong University. His research interest mainly includes swarm intelligence, and reliability maintenance engineering, and *etc.*