MPPT Study on Adaptive Chaos Particle Swarm Optimization Based on Local Shading

Jianhua Zhang[1, a], Zhanbiao Jia[2, b], and Menghui Xuan[2, c]

[1]*School of electrical engineering, Zhengzhou University, Zhengzhou 450000, China*
[2]*School of electrical engineering, Zhengzhou University, Zhengzhou 450000, China*
[a]*petermailsm@163.com*, [b]*zhanbiaojia@126.com*, [c]*menghuixuan@gs.zzu.edu.cn*

Keywords: MPPT Study, Local Shading, Particle Swarm Optimization

Abstract: Under complex shading conditions, the pv array outputs p-u image and it has multiple peaks, which may lead to the traditional maximum power point tracking algorithm, falling into the local extreme value and the tracking time being too long. For this reason, particle swarm optimization (PSO) is used to find the maximum power point under the shading condition. This paper is on the basis of the traditional particle swarm optimization algorithm, combining with chaos optimization, using linear synchronous learning factor change and adaptive inertia weight, and puts forward a new kind of self-adaptive chaotic particle swarm optimization algorithm (SA-CPSO). Compared with the PSO, this method can find the maximum power point of the system more quickly and accurately. The tracking time is only about 20% of the tracking time of PSO, moreover, the simulation model was built on Matlab/Simulink to verify its rapidity and accuracy.

1. Introduction

With the constant consumption of fossil energy, new energy sources such as pv power generation emerge. Due to the interconnection of pv power generation, how to maintain the pv system at the maximum power output (MPPT) is particularly important.

So far, there are many ways to implement MPPT, such as constant voltage method, disturbance observation method (P&O), conductance increment method, etc. However, such traditional methods may fall into local extremum under complex shading conditions. Therefore, some intelligent algorithms, such as particle swarm optimization algorithm, fuzzy control algorithm, genetic algorithm and artificial neural network algorithm, are proposed.

In this paper, through the establishment of a pv system model, and then compare the traditional PSO and adaptive chaotic particle swarm optimization (SA-CPSO), the method was proved not only has the ability of global search, but tracking time is short and has the small oscillation.

2. Pv system modeling

Pv cells actually use their own pv effect to generate electromotive force. Its equivalent circuit is composed of constant current source and some resistors.According to the V-I image of the photovoltaic cell, we can get the basic output characteristic formula:

$$I = I_{sc}[1-C_1(e^{U/(C_2 U_{OC})}-1)]$$
$$C_1=(1-I_m/I_{sc})e^{-U_m/(C_2 U_{OC})} \tag{1}$$
$$C_2=(U_m/U_{oc}-1)/\ln(1-I_m/I_{sc})$$

Including: Um, Uoc, Im, Isc are performance reference values, C1 and C2 are intermediate parameters.

Considering that there are still errors in the real environment, some performance correction methods can be adopted to reduce the errors caused by environmental changes.For example, when the environment changes, we can modify the above four reference values to obtain a more accurate mathematical model of photovoltaic cells.

$$D_I = S/S_{ref}[1+a(T-T_{ref})]$$
$$D_U = [1-c(T-T_{ref})]\ln(e+b(S-S_{ref}))$$
$$I'_{sc}=I_{sc}D_I \tag{2}$$
$$I'_m=I_m D_I$$
$$U'_{oc}=U_{oc}D_U$$
$$U'_m=U_m D_U$$

Including: Sref, Tref are reference value for the intensity of sunlight and temperature and they are 1000 w/m^2 and 25℃ respectively; S,T are the actual sun light intensity and temperature; a =0.0025, c=0.00288 and b=0.5 compensate temperature and light intensity respectively as compensation coefficient. The battery performance parameters are Um= 18.47V, Uoc= 23.36V, Im= 2.8A, Isc=3A, Pm= 51.7W. In simulink, with two batteries in series, were measured in the standard conditions (temperature 25 ℃, light 1000 w/m^2) and shade (temperature 25 ℃, light 1000 w/m^2 and 600 w/m^2 respectively) of the output characteristic curve. Such as figure 1 below:

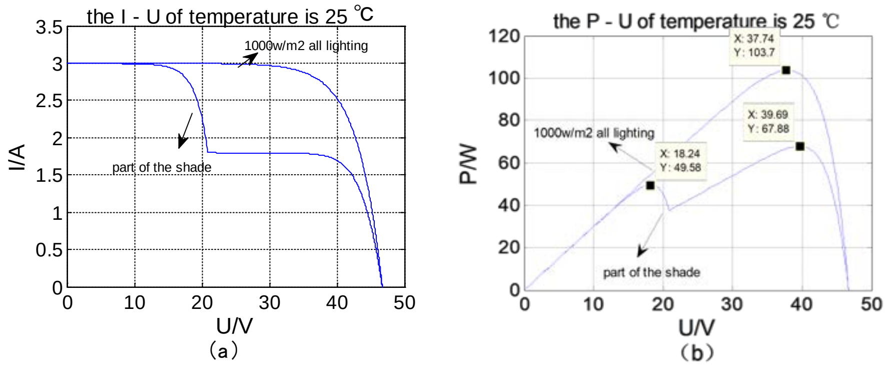

Figure 1. Comparison of standard conditions and partial shading output characteristics

We can see from the picture, when a partial shade, I - U output characteristic is changed from the original smooth to ladder type, P - U is changed from a peak to many peaks. At this point, the traditional optimization method may fall into the local extremum, leading to the failure of optimization.

2.1 Traditional particle swarm optimization

Assuming that there are N particles in the system that extend into the d dimensional space. The position of particle in the d-dimensional space $X^i=(x_{i,1}\ x_{i,2}...x_{i,d})$ is a vector, and the velocity of each particle $V^i=(v_{i,1}\ v_{i,2}...v_{i,d})$ is also a vector. All particles have an adaptive value determined by the optimized function f (X^i), and velocity V^i determines the direction and distance they travel.The system starts with a set of random particles and iterates to find the optimal value.In each iteration, the particle updates itself by tracking two optimal solutions, one of which is the optimal solution found by the particle itself, namely the individual extreme value,pbest,$P^i=(p_{i,1}\ p_{i,2}...p_{i,d})$;The other one is the optimal solution found by the whole population, namely the global optimal solution ,gbest,Pg.When these two optimal values are found, the particle updates its position and velocity according to the following formula.

$$v_{i,j}(t+1)=wv_{i,j}(t)+c_1r_1[p_{i,j}-x_{i,j}(t)] + c_2r_2[p_{g,j}-x_{i,j}(t)] \tag{3}$$
$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1),j=1,2,...d$$

Including: w is the inertial weight, c_1 and c_2 are positive learning factors, r_1 and r_2 are random numbers uniformly distributed between 0 and 1.

2.2 Self-adaptive chaotic particle swarm optimization algorithm

The performance of particle swarm optimization algorithm depends largely on the control parameters of the algorithm, such as particle number, maximum speed, learning factor and inertia weight. When w is large, the global search capability is strong. w is relatively small, the local search capability is strong.c_1 and c_2's influence on the system is mainly manifested on can determine your own experience and peers of flight, which affect the movement of the particles, can be understood as the particles through their own experience and other particles best experience determine the next movement. When c_1 is large, the particle increases the global search speed, while when c_2 is large, the particle will enhance the local optimization ability.

Therefore, when particles search iteratively, we can adjust the above parameters in real time to achieve the output of the maximum optimal value. The updated formula of linear synchronous change of learning factors is:

$$c_1=c_2=c=c_{max}-(c_{max}-c_{min})*t/MaxDT \tag{4}$$

Including: c_{max}=2.1 is the maximum value of learning factors,c_{min}=0.8 is the minimum value of the learning factor,MaxDT=50 is maximum number of iterations,t is current iteration number.

Adaptive inertia weight updating formula:

$$w=\begin{cases} w_{min}-(fv(i)-f_{min})*(w_{max}-w_{min})/(f_{avg}-f_{min}), & fv(i)<=f_{vag} \\ w=w_{max}, & fv(i)>f_{vag} \end{cases} \tag{5}$$

Including: w_{max}=0.9 is the maximum value of inertia weight,w_{min}=0.6 is minimum value of inertia weight, fv (i) is current objective function value, f_{min} and f_{vag} respectively represent the minimum value and average value of the objective function of all particles, w changes automatically with the objective function value.

At the same time, we introduce the idea of "chaotic iteration" in the algorithm. For the desired objective function, in order to reduce the possibility of falling into the local extreme value in the search process, we can treat the search process as the traversal process of chaotic orbit. The specific method is: first, the target value of all particles is found, and 20% of the best particles in the population are retained,Nbest=floor(N*0.2).Chaotic search for the 20%of the best particles in a population.The iterative formula is:

$$cx(dim)=(tmpx(1,dim)-xmin(dim))/(tmpx(1,dim)-xmax(dim))$$
$$cx(dim)=4*cx(dim)*(1-cx(dim)) \tag{6}$$
$$tmpx(1,dim)=tmpx(1,dim)+cx(dim)*(xmax(dim)-xmin(dim))$$

Including: 'dim' is the number of independent variables,'cx' is Chaotic variables,'tmpx' is decision variables.

If the accuracy or number of iterations is reached, the search stops.Otherwise, shrink the search area, randomly generate the remaining 80% of the particles in the population in the shrinking area, update the particle location and speed, and continue to evaluate.Shrinking area is:

$$xmin(s)=max(xmin(s),pg(s)-r*(xmax(s)-xmin(s)))$$
$$xmax(s)=min(xmax(s),pg(s)+r*(xmax(s)-xmin(s))) \quad (7)$$

Including:'s' is the number of independent variables.

Below is the flow chart of SA-CPSO algorithm:

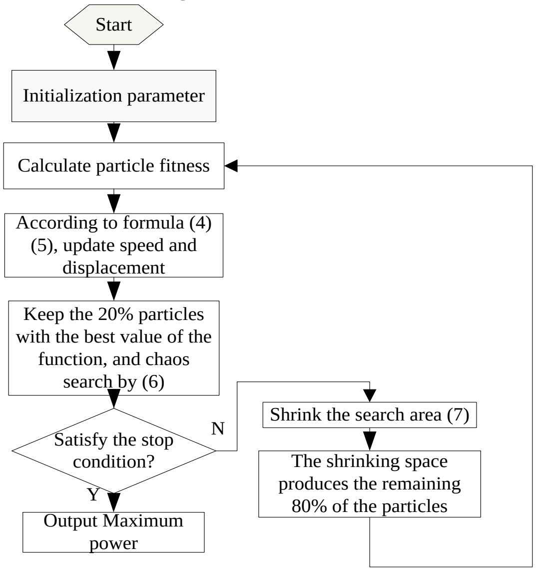

Figure 2. Flow chart of SA-CPSO

3. Analysis of simulation results

In order to verify the optimization effect of SA-CPSO relative to PSO under standard and partial shading conditions, the model was established in Matlab/Simulink， These include battery modules, MPPT control modules, boost circuits, and loads.$C1=C2=1e^{-6}F$,L=0.5H,Rload=200Ω.Under normal conditions and partial shading conditions, PSO and SA-CPSO were respectively used to track the maximum output power of photovoltaic cells, and the observation results were obtained.

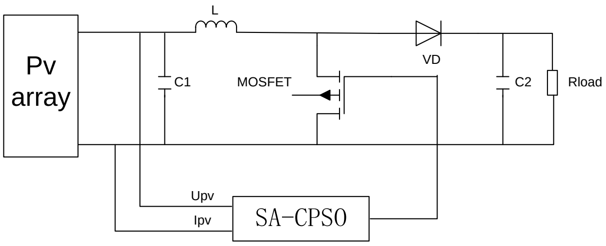

Figure 3. Pv array MPPT system

3.1 Comparison of PSO and SA-CPSO output results under standard conditions

The two photovoltaic cells are normally connected in series, and the P-U diagram is shown in

Figure 2, b. As can be seen from the figure, there is only one peak in the P-U graph, P_{max}= 103.7w.The following figure shows the power results searched by PSO and SA-CPSO methods under standard conditions. As can be seen from the figure, the optimal value can be found in both ways. Among them, PSO takes 0.1524s, and the optimal value searched is 103.5w. SA-CPSO only took 0.03744s, and the optimal value was 103.6w.Thus, PSO search time is about 5 times as long as SA-CPSO. Using SA-CPSO greatly reduces search time and improves efficiency.

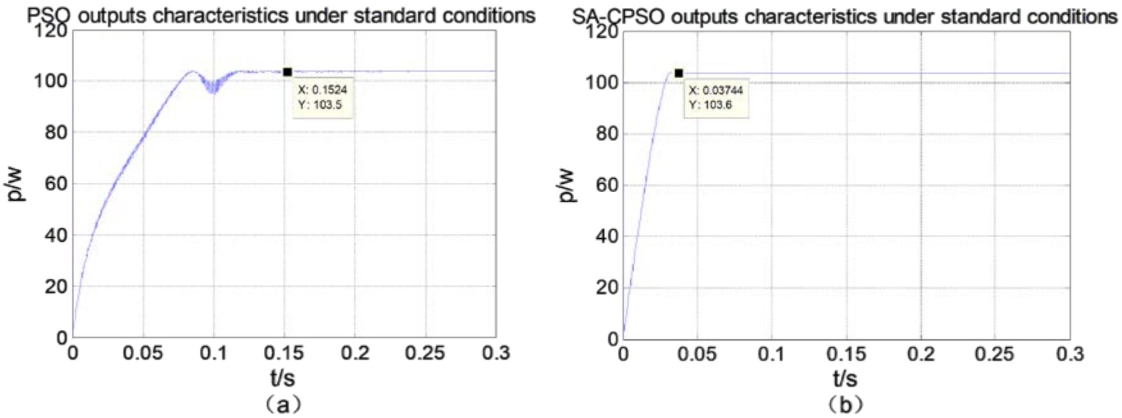

Figure 4. Output characteristics of the two methods under standard conditions

3.2 Comparison of PSO and SA-CPSO output results under partial shading condition

When the light is $1000W/m^2$ and $600W/m^2$ respectively, the two batteries are strung together. The P-U diagram is shown in FIGURE 2, b. As can be seen from the figure, there are two peaks in the P-U graph, P_{max}= 67.88w.In the following figure, power results are searched by PSO and SA-CPSO methods in the experimental simulation. As can be seen from the figure, the optimal value can still be found in both ways. Among them, PSO took 0.105s, and the optimal value was 65.72w. SA-CPSO was only used for 0.0228s, and the optimal value was 67.61w.The SA-CPSO search time is still about 20% of the PSO search time.

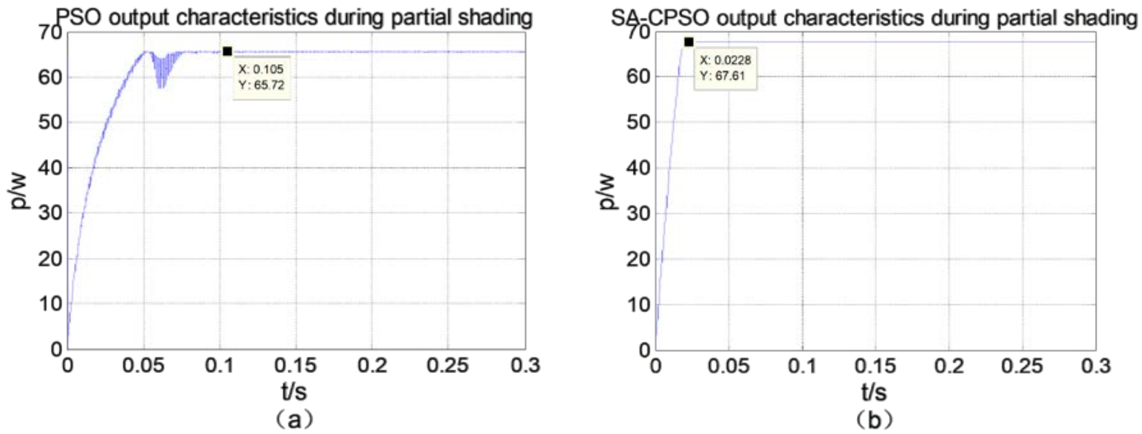

Figure 5. Output characteristics of two methods under partial shading conditions

4. Conclusion

This paper puts forward a method of adaptive chaotic local search of particle swarm(SA-CPSA), it is using chaos optimization and the combination of particle swarm optimization combined with linear synchronous learning factor change and adaptive inertia weight, to adjust the relevant parameters. By establishing relevant models in Matlab/Simulink, the search results of PSO and SA-

CPSA were compared.The results showed that: under the condition of standard and partial shade, SA-CPSO and PSO can achieve good tracking effect, and the SA-CPSO oscillation is small, about 20% of the time only for PSO tracking time, greatly improving the efficiency.

References

[1] Shi Jiying, Ling Letao, Xue Fei, etc. Study on photovoltaic global MPPT based on adaptive population particle swarm, J. Power electronics, 2017, (5):27-30.

[2] Tian Jingwen, Gao Meijuan. Research and application of artificial neural network algorithm, M. Beijing, Beijing University of Technology Press, 2006.

[3] Zhang Huanyue, Yan Yu, Xu Kai. Maximum power tracking strategy for distributed photovoltaic power generation based on PSO, J. Control engineering, 2015, (s1):51-57.

[4] Fu Jiangsheng, Xiao Dashuai, Yang Qinchao. Research on photovoltaic cell simulation model based on Matlab/Simulink, J. Application of electronic components, 2012, (2):44-46.

[5] Zhang Yongge, Shi Jiying, Zhang Wen,etc. Simulation study on improved PSO algorithm of pv system MPPT control under complex shading condition, J. Chinese journal of electrical engineering, 2014,34(s1):39-46.

[6] Gong Chun, Wang Zhenglin. Proficient in MATLAB optimization calculation, M. Beijing, Electronic Industry Press, 2016.

[7] Yuan Xiaoling, Chen Yu. The application of adaptive weighted particle swarm optimization algorithm in maximum power point tracking (MPPT) of shadow photovoltaic power generation ,J. China electric power,2013,46(10):85-90.

[8] Zhao Yang. Research on MPPT algorithm of photovoltaic power generation system based on observation of particle swarm fusion variable step disturbanc, D. Taiyuan University of Technology, 2017.

[9] Wang Qiye. MPPT control strategy of pv grid-connected system based on improved PSO algorithm, J. Power electronics,2017,(7):30-32.