Application of Improved Cat Swarm Optimization in MPPT Control of Photovoltaic Arrays

GUO Xiu-cai; XU Dan
Xi'an University of Science and Technology, Yanta Middle Road 58, Xi'an, Shaanxi Province, China
454862378@qq.com; 15398022836

Abstract. In the photovoltaic power generation system, the P-U output curve of the system has a multiple-peak phenomenon due to the influence of complicated factors such as partial shading. The traditional single-peak optimization based on the cat swarm optimization cannot effectively track the MPP point in the case of multiple peaks. This paper proposes a new improved cat swarm optimization, which is applied to the tracking of photovoltaic maximum power point. The method introduces the degree of aggregation, evolutionary speed factor, mutation operator and inertia weight factor. For the traditional cat swarm optimization (CSO), it is easy to fall into a local optimum. When the initial particle distribution is uneven or too concentrated, a narrow search space will be formed. The problem was simulated in the environment of MATLAB/SIMULINK, and the correctness of the method was verified by comparison and analysis with the traditional CSO.

1. Introduction
In the process of the development of human society, energy plays an extremely important role. Solar energy, as the most primitive energy source on the earth, is not only safe, clean, and inexhaustible, it can be converted into various forms of energy. [1] Power generation is one of the forms of conversion. However, the photoelectric conversion efficiency of PV cells is very low. Therefore, it is our expectation to convert the maximum efficiency of solar energy into electrical energy. At present, improving the output power efficiency of PV cells is the most direct and effective way. The maximum power point tracking (MPPT) technology is generally used to control the PV array in real time. Under uniform light conditions, the PV cell's power-voltage (P-V) characteristic exhibits a single peak. In non-uniform light conditions, the PV cell's power-voltage (P-V) characteristic exhibits multiple peaks. For multiple-peak cases, particle swarm optimization, genetic algorithm, and CSO are commonly used. This paper proposes an improved CSO. In this algorithm, the change of the inertia factor is affected by the running status of the algorithm.

2. Cat Swarm Optimization
CSO is a global optimization algorithm. The basic idea of algorithm design stems from the observation of the feline movement behavior. Although many cats survive in different environments, their behaviors have the same features.[2] In daily life, if you look closely at the cat, you will find that the cat is in a state of idleness and observance most of the time, but the cat is highly alert and maintains a high degree of vigilance even at rest; They have a strong sense of curiosity about moving targets, once they find that prey is being tracked, and they quickly catch prey. The cat's behavior is divided into two modes. [3] One is the mode when the cat is lazy and waits, called the search mode, and the second is...
the tracking behavior when the cat finds prey, which is called the tracking mode.

2.1. Improved Cat Swarm Optimization

CSO takes into account both the search mode and the tracking mode. Through the cooperative application of the two modes, the performance has been improved, but there are also deficiencies. At the end of the iteration, when the global optimum is gradually approached, the diversity of the population is reduced, and it is easy to fall into a local optimum, resulting in low tracking accuracy. It is difficult to quickly and accurately track the maximum power point. Therefore, this paper proposes an improved CSO, dynamic change of inertia factor, and introduction of mutation operator. In this algorithm, the change of the inertia factor is affected by the running status of the algorithm. It is determined by the evolutionary speed of the cat group and the degree of aggregation of the cat.

2.1.1. The evolution of cats and the concentration of cats

The improved CSO first randomly initializes a group of cats, and it finds the global optimal value \( x_{best}(t) \) in the search mode, and adjusts the speed and position of each cat in the tracking mode according to the optimal solution. The adjustment formula is as follows:

\[
\begin{align*}
    v_i(t+1) &= v_i(t) \times w + c \times \text{rand} \times [x_{best}(t) - x_i(t)] \\
    x_i(t+1) &= x_i(t) + v_i(t+1)
\end{align*}
\]

\( v \) is the speed of the cat; \( X \) is the current position of the cat; \( \text{rand} \) is the random number generated between (0, 1); \( c \) is called the learning factor, usually \( c=2 \); \( w \) is the inertia factor, which ranges from 0.1 to 0.9. \( v_i(t+1) \) is Cat speed after update, \( w \) is the inertia weight factor. \( x_i(t+1) \) represents the position of the i cat after the update.

In this paper, the degree of fitness represents power, where the power is a positive number that is consistently greater than zero. The global optimum depends on the change in the individual's optimal value, and it also reflects the movement of all cats in the cat group. In the iterative process, the global optimum of the current iteration is always better than or at least equal to the global optimum of the last iteration. Specifically, if the optimization goal is to find maximum:

\[
\text{max}[ \text{fitness}(t)] \geq \text{max}[ \text{fitness}(t-1)]
\]

Definable:

\[
h = \frac{\text{max}[ \text{fitness}(t-1)]}{\text{max}[ \text{fitness}(t)]} \quad (0 < h \leq 1)
\]

Let \( h \) be called the evolutionary speed factor. This parameter takes into account the running history of the algorithm and also reflects the evolutionary speed of the cat swarm. That is, the smaller the value of \( h \), the faster the evolution. After a certain number of iterations, the value of \( h \) remains 1 and the algorithm is determined to be stagnant or found optimal solution.

\[
\text{max}(\text{fitness}(t-1)) \text{ and } \text{max}(\text{fitness}(t)), \text{ respectively represent the maximum fitness of the two iterations before and after.}
\]

Another factor that affects the performance of the algorithm is the degree of aggregation of the cat. In the algorithm, the global optimal value is always superior to the current fitness value of all individuals. If \( f_{avg} \) is the average fitness of all cats. During the maximal search process:

\[
f_{avg} = \sum_{i=1}^{N} \text{fitness}(i) / N
\]

\[
\text{max}[ \text{fitness}(t)] \geq f_{avg}, \text{ Definable: } s = \frac{f_{avg}}{\text{max}[ \text{fitness}(t)]}, 0 < s \leq 1
\]

Let \( s \) be called the cat aggregation factor. It reflects the current degree of aggregation of all cats and also reflects the diversity of cats to some extent. The larger the value of \( s \), the greater the aggregation of the cats and the smaller the diversity of the cats. When \( s=1 \), all cats in the cat group have the same identity. If the algorithm falls into a local optimum, the result is not easy to jump out of the local pole.
2.1.2. Dynamic grouping rate
The grouping rate of the traditional CSO is a fixed value, so that the cat's dynamic characteristics could not be better reflected. The algorithm assigns more proportions of tracking cats at the beginning to improve the global search mode, thereby increasing the convergence speed of the algorithm. The search cat that distributes more algorithms later can make the algorithm jump out of the local optimal solution. The updated formula is as follows:

\[ MR = MR_{\text{max}} - (MR_{\text{max}} - MR_{\text{min}}) \times \frac{i_{\text{iter}}}{T} \]  

(2-7)

Where \( i_{\text{iter}} \) represents the current number of iterations and \( T \) represents the maximum number of iterations. \( MR_{\text{max}} \) and \( MR_{\text{min}} \) represent the maximum and minimum grouping rates, respectively.

2.1.3. Inertia weight factor
It is found that the algorithm has a strong global search capability when the inertia weight \( w \) is large, and the algorithm tends to search locally when \( w \) is small. In fact, the size of \( w \) should change with the evolution of cats and the degree of gradual aggregation of cats, \( w \) can be expressed as a function of \( h \) and \( s \): \( w = f(s, h) \). If the cat population evolves faster, the algorithm can continue to search in a larger search space, and the cat can maintain a wide range of optimization. When the evolution of the cat population slows down, the value of \( w \) can be reduced so that the cat group searches in a small space to find the optimal solution faster. If the cat is more dispersed, it is not easy for cats to fall into a local optimal solution. With the increase of the degree of aggregation of the cat group, the algorithm is likely to fall into a local optimum. At this time, the search space of the cat group should be increased, and the global search ability of the cat group should be improved.

In summary, \( w \) should increase as the cat's degree of aggregation increases, and as the evolution speed decreases, it can be expressed as:

\[ w = w_{\text{ini}} - h \times w_{h} + s \times w_{s}, w_{\text{ini}} \text{ is the initial value of } w, \text{ generally } w_{\text{ini}} = 1. \]  

(2-8)

Since \( 0 < h \leq 1, 0 < s \leq 1, w_{\text{ini}} - w_{h} < w < w_{\text{ini}} + w_{s} \).

Based on the above discussion, the improved CSO proposed in this paper dynamically adjusts \( w \) according to the values of \( h \) and \( s \) during operation, thereby improving the performance of the algorithm.

2.2. Application of Improved CSO in Local Shading PV MPPT
This paper builds a simulation model in MATLAB/SIMULINK to verify the simulation and uses the improved CSO to simulate the partial obstruction of the PV system MPPT. First, select three PV modules in series under stable lighting conditions. The parameters of the PV modules are:

- \( U_{\text{oc}} = 21.4V, U_{m} = 17.3V, I_{sc} = 3.8A, I_{m} = 3.65A \).
- The light intensity of the two batteries is: \( S_{1} = 1200W/m^{2}, S_{2} = 800W/m^{2}, S_{3} = 400W/m^{2} \).
- Environment temperature is: \( T = 25^{\circ}C \).

The simulation model is shown in Figure 2.1.
The algorithm is implemented as follows:

1. Initialize the various parameters of the cat group. Including setting the initial population size of the cat to N, the initial grouping rate \( MR_{\text{max}} \), \( MR_{\text{min}} \) and the maximum number of iterations T, randomly initialize the position of each cat, between \( [x_{\text{min}}, x_{\text{max}}] \), initialize the speed of each cat, at \( [v_{\text{min}}, v_{\text{max}}] \).

2. Calculate the fitness of all cats in the cat swarm based on the fitness function and record them, and fitness is represented by \( \text{fitness}(i) \). At the same time, compare and record the positions of the cats with the greatest fitness.

3. According to the distribution of the cats by MR, MR indicates the ratio of the cats to the cats in the tracking mode. The MR update formula is (2-7).

4. Calculate the degree of aggregation of the cat. The degree of aggregation reflects the current degree of aggregation of the cat. The calculation formula is (2-5) and (2-6).

5. Calculate the cat's evolutionary speed factor. The evolutionary speed factor reflects the speed of the algorithm. Its calculation formula is (2-3) and (2-4).

\[
\text{max} (\text{fitness}(t - 1)) \text{ and } \text{max} (\text{fitness}(t)), \text{ respectively represent the maximum fitness of the two iterations before and after, h represents the evolutionary speed factor.}
\]

6. Search mode, for each cat copy a certain individual and copy the cat into the memory pool SMP, the number of cats copied by each cat is determined by the size of its own fitness, the higher the fitness, the greater the number of cats copied. The formula for the number of copies of each cat is as follows:

\[
N_{\text{copy}} = \frac{\text{fitness}(i)}{\sum_{i=1}^{n} \text{fitness}(i)} \times N_{\text{copy\_sum}} \tag{2-9}
\]

\( N_{\text{copy}}(i) \) indicates the number of copies of the i-th cat, \( \text{fitness}(i) \) indicates the fitness of the \( N_{\text{copy\_sum}} \) indicates the total number of cats responsible.

7. Perform the mutation operator and give each copying cat a random perturbation in its original position. The cat enters a new position, replaces the old position with a new position, calculates the
fitness value of each cat in the new position, and adapts to each cat and its copying cat. The maximum value replaces the original cat's position and completes the update of the output power.

8. In the tracking mode, the entire cat group has experienced the best position currently, namely, the best $x_{best}(t)$ solution currently searched, and the speed of each cat is $v_i$, $w$ is the inertia weight factor. Then each cat is according to the formulas (2-1), (2-2) and (2-8) to update:

9. Record the best-adapted cat in the conservation population.

10. It is determined whether the termination condition is satisfied, and if the output optimal solution is satisfied, the loop starting from step 3 is not satisfied.

The output characteristics of three series arrays with different light intensities are shown in Figure 2.2. It can be seen from the figure that the $P-U$ curve has three peak points. The operating voltage and power of the maximum power point is only one, 90.88W.

![Fig. 2.2 Output characteristics of three series models under different light intensities](image1)

In order to verify the effectiveness of the algorithm, in the following, the basic CSO and the improved CSO are separately simulated to obtain the maximum power point tracking results of the two algorithms as shown in Figure 2.3 and Figure 2.4.

![Fig.2.3 Improved CSO tracks the results](image2)
From the picture, it can be seen that the maximum power value tracked by the improve CSO is 90.85W, and the actual value is 90.88W. The error between the two is 0.03W. And the maximum power value tracked by the basic CSO is 90.25W, and the actual value is 90.88W. The error between the two is 0.63W. Comparing Fig.2.3 and Fig.2.4, it can be seen that the time taken by the improved CSO to track to the MPP points is approximately 0.03 s, while the time taken by the basic CSO to track the MPP points is approximately 0.08 s. It can be seen that the improved CSO has higher tracking accuracy and faster convergence time than the basic CSO.

3. summary
By comparing the two algorithms, one can clearly see: The improved CSO tracks the maximum power point of the PV array faster and more accurately than the basic CSO. It is verified that the algorithm in this paper can successfully track the global maximum power point in the case of partial shading.

References
[1] Ren Z H. New Adaptive Particle Swarm Optimization Algorithm with Dynamically Changing Inertia Weight[J]. Computer Science, 2009.
[2] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. Information Processing Letters, 2016, 85(6):317-325.
[3] Zhao M, Pan J S, Chen S T. Compact Cat Swarm Optimization Algorithm[M]// Security with Intelligent Computing and Big-data Services. 2018.
[4] Bahrami M, Bozorg-Haddad O, Chu X. Cat Swarm Optimization (CSO) Algorithm[J]. 2018.