Power System Optimization Based on Multi-agent Structure and Lion Swarm Optimization

Ying Guo and Mingyan Jiang
School of Information Science and Engineering, Shandong University, Qingdao 266237, China
Corresponding Author: jiangmingyan@sdu.edu.cn

Abstract. To improve the global exploration capability of lion swarm optimization, this paper presents a novel lion swarm optimization based on multi-agent structure. The new algorithm combines the lion swarm optimization and the multi-agent system in order to improve efficiency and accuracy. It could use the group information and environment information to determine the search strategy in the iterative process. In solving the economic load distribution problem of power system, the lion swarm optimization based on multi-agent structure has significantly improved performance compared with lion swarm optimization in terms of efficiency and robustness. The experimental results on power system prove the effectiveness of this new algorithm.

1. Introduction

Economic load dispatch (ELD) [1,2] is one of the typical optimization problems in power system planning and operation scheduling. In recent years, intelligent optimization algorithms have been applied to ELD problems because of their intelligence, parallelism and robustness, which have good adaptability and strong global search ability. Among them, common intelligent optimization algorithms include artificial bee colony algorithm [3], artificial fish swarm algorithm [4] and particle swarm algorithm [5-9]. Lion swarm optimization (LSO) [10] is a new swarm intelligence optimization algorithm that mimics lion behaviour. LSO was proposed by Shengjian L et al in 2018 through long-term observation of lion hunting behaviour. The basic LSO has advantages of fast convergence speed, high optimization accuracy and good stability. Through the optimization results of some test functions, it can be seen that the lion's pride algorithm can achieve good optimization results for most of the test functions, which is obviously better than other common swarm intelligence algorithms.

In order to achieve a more satisfactory optimization effect of LSO on the optimization of more functions and make LSO better applied in solving ELD problems, this paper combines the competition and cooperation mechanism of the multi-agent system with LSO and proposes a new lion swarm optimization based on multi-agent structure (MALSO). By comparing the three classical test functions with other algorithms in different dimensions, MALSO has stronger global convergence, making particles not easy to fall into the local optimal. Besides, the convergence effect of MALSO is better. Finally, the IEEE40 node system of economic load distribution in power system is simulated, and the results also verify that the MALSO is successful and feasible.
2. Lion Swarm Optimization based on Multi-agent Structure

2.1. Lion Swarm Optimization

The basic idea of the lion pride algorithm is as follows: starting from an initial position to be optimized, the lion that finds the optimal fitness is defined as the lion king. Select a certain proportion of lionesses and cooperate with other lionesses to hunt. When prey is found to be superior to the lion's current position, the lion will take the place of the superior prey. The cubs follow the lioness as she hunts or feeds near the king. When the lion cubs grow into adults, they are driven out by the lion king and become stray lions, and the stray lions move closer to the optimal position in memory. LSO is to search for the optimal value of the objective function through such individuals in accordance with different division of labour and close cooperation.

Suppose N lions form a group in the D-dimensional search space, and the position of the i lion in the D-dimensional space is denoted as \( x_i = (x_{i1}, x_{i2}, ..., x_{iD}) \). Each generation of lions updates its position according to the following equation:

The lion king position which represents the locally optimal, is updated according to the equation:

\[
x_i^{k+1} = g^k(1 + \gamma||p_i^k - g^k||)
\]  

(1)

Lionesses cooperate with each other to update the position. And the equation of lioness is as shown:

\[
x_i^{k+1} = \frac{p_i^k + p_c^k}{2}(1 + \alpha_f \gamma), \quad \alpha_f = \text{step} \cdot \exp(-30(T)^{10})
\]  

(2)

Cub locations are updated as follows:

\[
x_i^{k+1} = \begin{cases} 
\frac{g^k + p_i^k}{2}(1 + \alpha_c \gamma), & q \leq \frac{1}{3} \\
\frac{p_m^k + p_i^k}{2}(1 + \alpha_c \gamma), & \frac{1}{3} \leq q \leq \frac{2}{3}, \quad \alpha_c = \text{step}\left(\frac{T-t}{T}\right) \\
\frac{\text{Low} + \text{High} - g^k + p_i^k}{2}(1 + \alpha_c \gamma), & \frac{2}{3} \leq q < 1
\end{cases}
\]  

(3)

Where \( \gamma \) is the random number generated according to the normal distribution N(0,1); \( p_i^k \) is the historical optimal position of the k -generation of the i lion; \( g^k \) represents the optimal position of the k -generation group; \( p_c^k \) is the historical best location of a randomly selected hunting partner from the k -generation lionesses; \( p_m^k \) is the best position in the k -generation history of the young lion following the lioness \( \alpha_f \) is the disturbance factor of the moving range of the lioness; \( \alpha_c \) is the disturbance factor of the movement range of the young lion; \( T \) is the total number of iterations; \( t \) is the number of contemporary iterations; The probability factor \( q \) is the uniform random value generated according to the uniform distribution U[0,1].

2.2. Lion Swarm Optimization Based on Multi-agent Structure

Agent and multi-agent system [7] are characterized \( \alpha_c \) by openness and flexibility. The structural advantage and interaction ability of the algorithm can be reflected in the new algorithm. In the multi-agent pride optimization algorithm, each Agent is abstracted as a single individual in the pride algorithm. In this way, each individual not only has a strong global convergence ability in LSO. At the same time, each individual can continuously learn experience, interact with other Agent neighbors, and complete the competition and cooperation just like agents in the multi-agent system.

2.2.1 Environment of agents

In the multi-agent system, the environmental information of Agent is one of the necessary factors to solve the problem. With the help of a simple lattice environment [6], each Agent is randomly initialized in an environment with a total lattice number of \( T_{size} \times T_{size} \). And each Agent occupies a grid, and the data in the grid represents the location information in the Agent's environment. \( T_{size} \) is a positive integer, and the total number of cells corresponds to the population size in the LSO. Each
Agent perceives environmental information from its local environment and can make action decisions based on the perceived local environment information.

2.2.2 Action strategies of agents

In the MALSO, each Agent updates its own location. However, the difference with LSO is that the Agent must first compete and cooperate with the neighbor particles in the local environment before updating. In this paper, the number of neighbors is set as 8.

First, each neighbor calculates its own adaptive value according to the objective function of the specific problem. It is assumed that \( \mu = (\mu_1, \mu_2, \cdots, \mu_n) \), and the Agent \( \mu \) has the minimum adaptive value among the eight neighbors of the Agent \( \theta \). If the Agent satisfies the equation (4), it is a high-quality individual; otherwise, it is a low-quality individual.

\[
f(\theta) \leq f(\mu) \tag{4}
\]

If the Agent \( \theta \) is a good individual, its position in the solution space remains the same. Conversely, the position of the Agent \( \theta \) in the solution space is based on the equation (5) make adjustments.

\[
\theta'_k = \mu_k + \text{rand}(-1,1)(\mu_k - \theta_k) \quad k = 1,2,\cdots,n \tag{5}
\]

Where, \( \text{rand}(-1,1) \) is the random number in the interval \((-1,1)\). If \( \theta'_k < x_{k\min} \), then \( \theta'_k = x_{k\min} \); If \( \theta'_k > x_{k\max} \), then \( \theta'_k = x_{k\max} \). \( x_{\min} \) is the lower limit of the feasible solution space of the optimization problem, and \( x_{\max} \) is the upper limit. It can be seen from equation (5), even if the Agent \( \theta \) is an inferior individual, it still fully absorbs the beneficial information of the best neighbor individual Agent \( \mu \) on the basis of retaining its original useful information, further reducing its adaptive value.

Then the multi-agent system is combined with the LSO. After competition and cooperation, each individual modifies its own action strategy and uses equation (1), (2) and (3) to exchange information with the optimal individual in the pride. This overcomes the environmental limitations and low speed efficiency of the useful information transfer of single Agent in the multi-agent system, speeds up the flow of information in the multi-agent system, and improves the convergence speed of the algorithm.

2.2.3 Self-learning mechanism

In order to reduce the computation time of the algorithm, only the optimal individual in each iteration, namely lion king, is generally self-learned. It not only improves the efficiency of the algorithm, but also improves the precision of the search. The basic process is described as follows:

It is assumed that the location of the Agent \( T_{i,j} \) in the solution space is \( T_{i,j} = (t_i, t_2, \cdots, t_n) \), and a micro-local environment with the size of \( T_{\text{size}} \times T_{\text{size}} \) is constructed. The position \( T'_{i',j'}(i',j' = 1,2,\cdots,T_{\text{size}}) \) of each Agent in the micro-environment can be obtained according to the following equation:

\[
T'_{i',j'} = \begin{cases} T_{i,j} & i' = 1; j' = 1 \\ TT'_{i',j'}, \text{else} \end{cases}
\tag{6}
\]

Where, \( TT'_{i',j'} = (tt_{i',j',1}, tt_{i',j',2}, \cdots, tt_{i',j',n}) \), where \( tt_{i',j',k} \) can be obtained from equation (7).

\[
\begin{align*}
\text{If } k &< \frac{R}{r} \\
\text{Then } tt_{i',j',k} &= \begin{cases} x_{k\min}, & \text{If } l_k \text{rand}(1 - sR, 1 + sR) < x_{k\min} \\
x_{k\max}, & \text{If } l_k \text{rand}(1 - sR, 1 + sR) > x_{k\max} \\
l_k \text{rand}(1 - sR, 1 + sR), & \text{else}
\end{cases}
\end{align*}
\tag{7}
\]

Where, \( sR \) is the radius of the local search, and \( sR \in [0,1] \).
3. Lion Swarm Optimization based on Multi-agent Structure for ELD

3.1. Objective Function of ELD
ELD can be summarized as a nonlinear single-objective multi-constraint optimization problem. The model of the objective function is shown in equation (8) to (10).

\[ F = \sum_{i=1}^{N_g} F_i(P_i) + \sum_{i=1}^{N_g} E_i \]  

Among the equation:

\[ F_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i \]  
\[ E_i = |g_i \sin(h_i(P_i - P_{i,min}))| \]

Where, \( F \) is the total power generation cost of the system; \( F_i(P_i) \) is the consumption characteristic of the generator \( i \); \( N_g \) is the total number of generators in the system; \( E_i \) is the characteristic change of consumption caused by valve point effect due to the phenomenon of drawing when the lower inlet valve of the steam turbine is suddenly opened; \( \alpha_i, \beta_i \) and \( \gamma_i \) are the consumption curve coefficients of the first generator; \( h_i \) and \( g_i \) are the valve point effect parameters of generator \( i \); \( P_i \) and \( P_{i,min} \) are the lower limits of the active power and active power of the generator.

3.2. Constraint Conditions for ELD
A) generator operation constraint conditions

\[ P_i \text{ min} \leq P_i \leq P_i \text{ max} \quad i = 1,2,\ldots,N_g \]  

Where, \( P_i \) is the active power of the generator \( i \); \( P_{i \text{ min}}, P_{i \text{ max}} \) is the lower limit and upper limit of the active power of the generator \( i \).

B) power balance constraint conditions

\[ \sum_{i=1}^{N_g} P_i = P_L + P_S \]  

Where: \( P_L \) is the total system load; \( P_S \) is the total network loss of the system.

3.3. Process of MALSO Solving ELD
MALSO method is applied in load distribution problem. Particles in multi-agent environment learn through competition and cooperation, and constantly update individual position and global extremum. The specific process is shown in table 1.
Table 1. Process of MALSO

| Algorithm: MALSO |
|------------------|
| Requires: construct a grid environment of $T_{size} \times T_{size}$; |
| set the generation $k=1$; |
| 1. initialization sets the size of lioness and cub, maximum number of iterations, local search radius and other parameters; |
| 2. calculate the fitness value $F$ for each particle; |
| 3. while (not meet the termination conditions) |
| 4. each lion is assigned 8 neighbours with random spacing |
| 5. for (each member in the scope) |
| 6. compete and cooperate with each neighbour separately; |
| 7. record the respective fitness values; |
| 8. end for |
| 9. if (the best member is better than that in $k-1$ generation;) |
| 10. update the position of each Agent in the solution space; |
| 11. calculate the fitness of each member; |
| 12. if (the best member is better than that in $k-1$ generation;) |
| 13. update the individual extremum of each particle and the global extremum of the population; |
| 14. set $k=k+1$; |
| 15. end while |

4. Experimental Results and Analysis

In order to evaluate the general characteristics and optimization performance of MALSO, three standard test functions were selected to conduct experiments with MALSO algorithm, including Rosenbrock function in equation (13). In this section, MALSO, PSO and LSO are applied to the above three test functions, and the results are summarized and compared. In order to avoid the influence of different initial states, the population and the maximum number of iterations were set as 64 and 500, respectively. Table 2 show the optimization results of the function after each algorithm runs for 20 times, independently on PC MATLAB R2007a of Intel Core i5-8250u processor with Windows10 64-bit operating system, 3.4GHZ main frequency and 8GB memory.

Rosenbrock:

$$f = \sum_{i=1}^{D} [100(x_{i+1} - x_{i}^2) + (x_{i} - 1)^2]$$

The function satisfies that the dimension is 2/30/100, and the values of the variables range from -2.048 to 2.048.

Table 2. Comparison of PSO, LSO, MALSO in Rosenbrock function

| Dimension | Algorithm | Average Time/s | Average Solution | Theoretical Optimal |
|-----------|-----------|----------------|------------------|---------------------|
| 2         | PSO       | 0.5198         | 8.8818e-16       | 0                   |
|           | LSO       | 1.1614         | 6.2226e-04       | 0                   |
|           | MALSO     | 3.61335        | 0                | 0                   |
| 30        | PSO       | 0.5337         | 41.1551          | 0                   |
|           | LSO       | 1.0463         | 28.9317          | 0                   |
|           | MALSO     | 4.2439         | 20.1278          | 0                   |
| 100       | PSO       | 0.8145         | 216.7787         | 0                   |
|           | LSO       | 1.1138         | 98.9206          | 0                   |
|           | MALSO     | 4.8053         | 95.5464          | 0                   |
According to the analysis in table 2, in the optimization process of different dimensions of Rosenbrock function, the optimization accuracy of LSO is better than that of PSO. Both MALSO and LSO can obtain the ideal solution of the function in two dimensions, but the result error obtained by the algorithm in this paper is smaller. In high dimensional space, LSO appears precocity and falls into local optimal value. However, MALSO can rapidly converge, jump out of the local optimal and approach the theoretical optimal value. The conclusion is the same for the Sphere function and the Himmelblau function.

When solving the ELD problem of power system, the system with 40 high-dimensional units was selected, with a total load of 10500MW. The algorithm was set as 500 iterations, and each experiment independently runs for 100 times. The convergence curves of MALSO, LSO and PSO are shown in figure 1. The comparison results of MALSO, LSO and PSO are shown in table 3. The simulation results of IEEE 40-nodes system show that MALSO has fast convergence speed, high quality solution and running speed, and strong stability when solving high dimensional complex systems.

5. Conclusion
The MALSO proposed in this paper is a new intelligent optimization algorithm that combines the competition and cooperation mechanism of the multi-agent system with the LSO. Compared with the PSO and LSO, although the fusion of various mechanisms in MALSO increases its time consumption, the shorter time of PSO and algorithm is at the expense of the convergence accuracy of the algorithm. Taken together, MALSO reflects the superiority of the optimization results and precision. MALSO is applied to solve the optimization problem of high-dimensional complex power system, and it has a good optimization result, which reflects the feasibility of MALSO in practical application.

6. Acknowledgment
This study is financially supported by the National Science Foundation of China (Grant No.61771293)
and Key Project of Shandong Province (Grant No.2019JZZY010111).

7. References

[1] Park J, Lee K, Shin J, et al 2003 Economic Load Dispatch for Non-smooth Cost Functions using Particle Swarm Optimization vol 2 IEEE Power Engineering Society General Meeting (Toronto, Ontario Canada) pp 938-943

[2] Yunhe H, Lijuan L, Xinyin X, et al 2004 Enhanced Particle Swarm Optimization Algorithm and Its Application on Economic Dispatch of Power Systems vol 24 Proceedings of the CSEE pp 95-100

[3] Mingyan J, Dongfeng Y 2014 Artificial Bee Colony Algorithm and Its Application (Beijing: Science Press)

[4] Mingyan J, Dongfeng Y 2012 Artificial Fish Swarm Algorithm and Its Application (Beijing: Science Press)

[5] Chuanjian L and Qianjin L 2011 A Distribution Network Reconfiguration Algorithm Based on Multi-agent System and Minimum Spanning Tree Led by Particle Swarm Optimization vol 39 Power System Protection and Control pp 24-28

[6] Bo Z and Yijia C 2005 vol 25 A Multi-Agent Particle Swarm Optimization Algorithm for Reactive Power Optimization Proceedings of the CSEE pp 1-7

[7] Xianlun T, Heng Z, Jin L, et al 2012 Economic Load Distribution of Power System Based on Particle Swarm Optimization vol 40 Power System Protection and Control pp 42-47

[8] Xuewen Z and Yanjun L 2006 Self-adjusted Particle Swarm Optimization Algorithm Based Economic Load Dispatch of Power System vol 30 Power System Technology pp 8-13

[9] Chengu W, Haiming L, Dong L, et al 2016 Application of Improved Particle Swarm Optimization Algorithm to Power System Economic Load Dispatch vol 44 Power System Protection and Control pp 44-48

[10] Shengjian L, Yan Y, Yongquan Z 2018 A Swarm Intelligence Algorithm-Lion Swarm Optimization. Vol 31 Pattern recognition and artificial intelligence pp 431-441