A Power Generating Distribution Planning Using Swarm Moth-flame Optimization

Trong-The Nguyen[2], Van-Chieu Do[2], Thi-Kien Dao[1*]

[1] School of Computer Science and Mathematics,
Fujian University of Technology, Fuzhou 350118, China
[2]Haiphong University of Management and Technology, Haiphong, Vietnam

vnthe@hpu.edu.vn, chieudv@gmail.com, *jvnthe@gmail.com

Abstract. Power generating distribution planning (PDP) is a critical component of the power system's financial performance. An effective PDP model has been constructed by carefully considering various system operation restrictions typical of multi-constrained nonlinear optimization problems. This work suggests solving the PDP model based on a new swarm-based technique called moth flame optimization (MFO) to achieve energy-saving and cost consumption reduction. In the experimental section, the IEEE-bus benchmark test systems are used to verify the performance of the proposed scheme system. The results show that the proposed scheme can solve the power system PLP problem with good robustness and significant economic benefits.

Keywords: Power distribution planning; Economic dispatch; Swarm moth flame optimization; Optimization algorithm

1 Introduction

Power generating distribution planning (PDP) is a critical component of the power system's financial performance [1]. Because PDP determines each unit's output in a given time under the given load demand [2], in the power system, each generating unit can share the load demand to meet the actual constraints and seek to minimize the total operation cost of the whole system [3]. The PDP is considered a fundamental problem in the operation of modern power systems and plays an essential role in improving the operation economy of power systems [4]. The economic dispatch model of a power system has only the constraints of unit capacity limitations and power balance but also considers many practical nonlinear conditions existing in the power system operation process, e.g., the unit forbidden area, valve point effect, and climbing rate limit [5]. Therefore, the power system's PDP problem is essentially a non-smooth, highly nonlinear multi constraint optimization problem [6]. The traditional classical mathematical algorithm, e.g., Lagrange relaxation method, linear programming (LP) for the complex problem, would have faced challenging time complicated for the computation. The development of swarm computing methods is one of the most effective ways [7][8] can deal with the complex issues of nonlinear like the PDP problem [9][10].

Moth flame optimization (MFO) is a new swarm intelligence algorithm in recent years [11]. Its idea comes from the simulation of the lateral flight mechanism behind the moth's behavior. MFO algorithm has advantages such as model simple and easy to implement. The FMO has been widely used in the engineering field, e.g., analysis of pumping test data, which provides an effective method for accurately estimating confined aquifer parameters, optimizing the Muskingum model's parameters with higher simulation accuracy, etc [12].

This paper introduces solving the PDP model of the power system with the MFO algorithm to overcome the premature and local convergence problems and to improve the global optimization ability. In addition, we also suggest a method combining the balancing unit with the penalty function to deal with power balance equality constraint and unbalanced power allocation problem to improve the algorithm's convergence speed and calculation accuracy.

2 Power Generating Distribution Planning Model

As a nonlinear optimization problem of the PDP model [5], we considered it multiple constraints under the premise of ensuring the safe and stable operation of the system. The output of the unit of system operation is determined by the decision to meet the output plan and minimize the generation cost of the system. Two components in modeling for a system power system are objective function and dealing with its constraints detailed as follows.

Objective function

The unit operation cost would determine the economic distribution of load among operating units is generally expressed as a function of unit output. If the fuel cost function of the power system is regarded as the summation of a series of quadratic polynomials, the PDP's fitness function is presented as follows.

$$\min \sum_{i=1}^{n'} F_i(P_i) = \sum_{i=1}^{n'} (\alpha_i P_i^2 + \beta_i P_i + \gamma_i) \tag{1}$$

Where, $F_i(P_i)$ is the function of fuel/coal cost in *i-th* unit; α_i, β_i and γ_i are the cost function's coefficients; P_i and n' are the i-th unit's output and the total number of units with their operations. In the practical power system, the PDP objective function considering "valve point effect" can be expressed as

$$F_{cost} = \min \sum_{i=1}^{n'} F_i(P_i) = \sum_{i=1}^{n'} \left(\alpha_i P_i^2 + \beta_i P_i + \gamma_i + e_i \left|\sin\left[f_i(P_i^{min} - P_i)\right]\right|\right) \tag{2}$$

Where e_i and f_i are the coefficients reflecting the "valve point effect" of the i-th unit; P_i^{min} is the i-th unit's lower limit output.

Constraints

In order to solve the above problems, a relatively comprehensive PDP model closer to the actual operation will be established by comprehensively considering many practi-

cal constraints in power system operation, including power balance, generation capacity, restricted area, and climbing rate limit conditions.
a) The power balance constraint is

$$\sum_{i=1}^{n'} F_i = P_{load} + P_{loss} \quad (3)$$

Where: P_{load} and P_{loss} are the power system's total load and the network loss, which can be approximately presented as a function of unit output by coefficient matrix **B**

$$P_{loss} = \sum_{i=1}^{n'} \sum_{j=1}^{n'} P_i B_{ij} P_j + \sum_{j=1}^{n'} B_{0i} P_i + B_{00} \quad (4)$$

Where, B_{ij}, B_{00}, B_{0i} are the known transmission loss parameters; P_i and P_j are the output of the i and j units respectively.
b) The generation capacity constraint is

$$P_i^{min} \le P_i \le P_i^{max} \quad (5)$$

Where: P_i^{max} is the i-th thermal power unit's upper limit of the output.
c) The restricted area is

$$\begin{cases} P_i^{min} \le P_i \le P_{i,1}^L \ , \\ P_{i,j-1}^U \le P_i \le P_{i,1}^L \ , \text{ with } j = 1,2,\dots,n_i, i = 1,2,\dots,n' \\ P_{i,n_i}^U \le P_i \le P_i^{max} \ , \end{cases} \quad (6)$$

Where: $P_{i,1}^L$ is the lower limit of the first forbidden area of the i-th unit; $P_{i,j-1}^U$ is the upper limit of the $j-1$ forbidden area of the i-th unit; n_i is the number of the forbidden areas of the i-th unit; P_{i,n_i}^U is the i-th unit's upper limit of the last forbidden area.
d) Climbing rate limit.
When the output increases, it is

$$P_i - P_i^0 \le U_{Ri} \quad (7)$$

When the output is reduced, it is

$$P_i - P_i^0 \le D_{Ri} \quad (8)$$

Where P_i^0 is the output of the previous stage of unit i; U_{Ri} and D_{Ri} are the unit(i) 's maximum upward climbing rate MW / time period and the maximum downward climbing rate MW / period.

3 PDP Optimization based on Moth-Frame Algorithm

3.1 Principle of Moth Frame algorithm

MFO inspired from the moth and fire are two crucial components considered the candidate optimization solution of the problem, moth flies in the decision space, and the fire is the best position that the moth has found up to now[11]. Therefore, the fire is applied to the "wind vane" for moths to search in the searching space. Each moth-frame is considered an agent searching to search around a fire and updates its position when it finds a better solution. The searching moth and fire agents would be presented with a matrix M and F respectively as follows [11].

$$\mathrm{M}=\begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nd} \end{pmatrix}, \quad \mathrm{F}=\begin{pmatrix} F_{11} & F_{12} & \cdots & F_{1d} \\ F_{21} & F_{22} & \cdots & F_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ F_{n1} & F_{n2} & \cdots & F_{nd} \end{pmatrix} \tag{9}$$

Where, n is the number of moths and d is the dimension of the variable. The vector-storing all moth and fire fitness can be expressed as

$$O_M = [O_{M1}, O_{M2}, \dots, O_{Mn}]^T \ , O_F = [O_{F1}, O_{F2}, \dots, O_{Fn}]^T \tag{10}$$

Where: O_M is the fitness matrix of the moth; O_{M1} is the fitness of the first moth; O_{Mn} is the fitness of the nth moth; O_F is the adaptability matrix of the fire; O_{F1} is the fitness of the first fire; O_{Fn} is the fitness of the nth fire.

Each moth updates its position around its corresponding frame fire:

$$M_i = S(M_i, F_j) \tag{11}$$

Where M_i is the position of the ith moth; F_j is the position of the jth fire; S is the logarithmic spiral function. The logarithmic spiral function is

$$S(M_i, F_j) = D_{ij} e^{bt} \cos(2\pi t) + F_j \tag{12}$$

Where b is the constant coefficient for logarithmic spiral function with distance D_{ij}; t is a random number [- 1 to – 2] in the iteration loops. The smaller t is, the closer the search position is to the fire. By taking different values of T, the other position relations between moth and fire can be obtained. The update mechanism of MFO algorithm with abscissa t is a random number in the range of r ~ 1, and the ordinate is the position of the moth. The logarithmic spiral update formula searches new positions around the fire by moth agent and effectively utilizes historical optimization solution as local development ability and exploring the unknown area as global search ability to realize the resolution of the optimization problem. The exploring and exploiting the searchability of the algorithm, the number of agent's fires are used to update the position changing dynamically under iteration times. The number of agents' fires under the current iteration is stated in the following formula.

$$N_{\text{flame}} = N - L\frac{N-1}{T} \tag{13}$$

Where: N_{flame} and N is the number of fires under the current iteration number and population the maximum number; T and L the maximum and the current numbers of iterations; if $\mathrm{N} = \mathrm{n}$, it is the same as the number of moths or the number of the maximum fire updating the position.

3.2 Solution of PDP model Using Moth Fire Algorithm

Agent moth code variables.

The optimal variable of the PDP problem is the active power output of each unit, and the dimension of the problem is equal to the total number of units. Combined with the characteristics of the MFO algorithm and PDP model, each moth represents a candidate solution of the power system load economic dispatch problem, and the position of each moth is a vector composed of the output of each unit.

$$P = \begin{pmatrix} p_1^1 & \cdots & p_i^1 & \cdots & p_{n'}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_1^i & \cdots & p_i^i & \cdots & p_{n'}^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_1^n & \cdots & p_i^n & \cdots & p_{n'}^n \end{pmatrix} \quad (14)$$

Where: P is the output moth matrix of the generator set, and the row vector of the matrix P represents the specific position of each moth.

Only the modified position should be in the area, meeting the conditions for the output capacity constraint and the forbidden area constraint. Due to the limitation of unit capacity and climbing rate, the output capacity of the first unit has upper and lower limits. The lower bound of output is expressed as follows.

$$l_i = \max(P_i^{min}, P_i^0 - D_{Ri}) \quad (15)$$

The upper bound of output is

$$u_i = \max(P_i^{min}, P_i^0 - U_{Ri}) \quad (16)$$

In updating the position, the moth may cross the feasible boundary formed. When one or more units fail to meet the output capacity constraint, the maximum range of the modified constraint is mapped symmetrically to the interior of the boundary according to the number of units violating the boundary constraint. Then the corrected unit output is randomly determined between the limit and the mapping positions within the boundary.

$$P_i = \begin{cases} l_i + R(l_i - P_i) & , P_i < l_i \\ u_i + R(u_i - P_i) & , P_i > u_i \\ P_i & , others \end{cases} \quad (17)$$

Where R is a random number generated by uniform distribution between 0 and 1.

The existence of the forbidden zone makes the decision-making space no longer continuous, and there is an interruption area that initializes the output of a unit randomly between the upper and lower boundaries. The power balance constraint is an equality constraint. The penalty function method is used alone to deal with the conditions. Most of them are infeasible solutions because of the large amount of time spent in the early stage of the algorithm.

New power balance constraint

Some steps consider network transmission loss: Set the number of moths n, the maximum cycle iteration number Q and the acceptable unbalanced power value P_0, and the cycle count $C_0 = 1$. T is the number of units that may have a balancing unit, the initial value is equal to the total number of units n', A random sequence x generated by the ordered sequence {1,2,...,g} is generated. Generate a balancing unit sequence with $(n' - T + 1)$ element from random sequence X. Calculate the network transmission loss P_{loss}, and then approximately determine the output of the balancing unit

$$t_{emp} = P_{load} + P_{loss} - (\sum_{i=1}^{n'} P_i - P_I) \quad (18)$$

Where P_I is the power used to balance the unit. If the output of the balancing unit is within the upper and lower limits of the unit output, check whether it falls in the forbidden area; if it falls in the forbidden area, the output of the balancing unit is set as the nearest boundary value of the prohibited area; if not, the value of temp is directly set as the output value of the balancing unit; If the output of the balancing unit exceeds the unit output, the output of the balancing unit is set as the output value of the balancing unit If t is greater than 0, return to step 3 to re-enter the internal circulation; otherwise, it means that the cycle has completed all units and no balancing unit has been found.

Recalculate the network transmission loss P_{loss}, and use equation (19) to calculate the unbalanced power ΔP. if $C_0 \leq Q$ and $\Delta P > P_0$, $C_1 = C_0 + 1$ and return to step 2 to re-enter the external loop iteration, otherwise, the constraint processing is finished.

$$\Delta P = P_{load} + P_{loss} - \sum_{i=1}^{n'} P_i \tag{19}$$

Fig. 1 shows a flowchart of the moth flame optimization for the power system PLP problem.

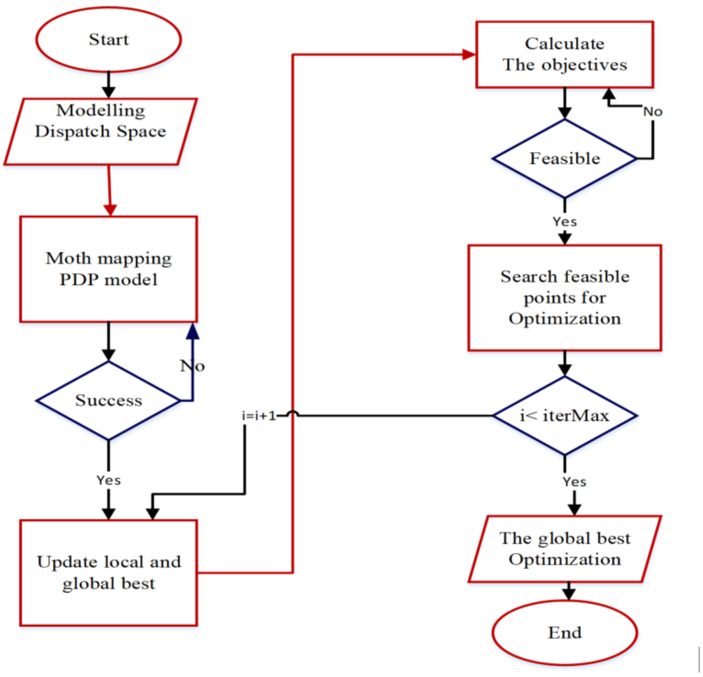

Fig. 1. A flowchart of the moth flame optimization for the power system PLP problem

Algorithm implementation process

The specific process of applying moth to fire algorithm to solve the PDP model of power system is as follows.

Step 1: the moth position is initialized, and all moths are randomly initialized between the upper and lower bounds of the unit output capacity constraint shown in Eqs. (15) and (16).

$$P_i = l_i + R(u_i - l_i) \tag{20}$$

Step 2: unit operation constraint processing, check the position of all moths. If the output of a moth falls in the forbidden area, re-initialize the output of the unit whose particle falls in the banned area at random until the constraint in the forbidden area is met. If the output of some units of a particle exceeds the upper and lower limits, the unit output shall be reassigned according to Eq. (16).
Step 3: power balance constraint treatment. The power balance constraint is treated by the balance unit method. The new moth population can be obtained by combining step 2 and step 3.
Step 4: fitness assessment, using the cost function shown in Eqs. (1) or (2) to evaluate the fitness of all moths. Suppose the balance unit method still does not meet the equality constraints. In that case, the penalty function method shown in Eq. (21) is used to punish the unbalanced power, forcing the moth to fly away from the infeasible region and explore the feasible area. Among them is the total fuel cost (total fitness) considering penalty cost, is the fuel cost function of unit I, and is the penalty factor.

$$F_{cost} = \sum_{i=1}^{n'} F_i(P_i) + k(\sum_{i=1}^{n'} P_i - P_{load} - P_{loss}) \tag{21}$$

Step 5: update the location and fitness of the fire. If it is in the first generation, the initial population of the moth is directly used as the initial population of the fire. In the iteration process, the first m individuals with the best fitness were selected from the fire population and the updated moth population as the new fire population, and the fire fitness was updated.
Step 6: moth position update, moth use Eq. (12) to update its position with reference to the corresponding fire, and the number of fires decreases dynamically according to Eq. (13).
Step 7: determine the optimal launch and termination conditions. If the maximum number of iterations is reached, the iteration will be terminated. The optimal scheduling scheme (the best fire location) and the generation cost (the best fire fitness) of the output economic dispatch problem; otherwise, the iterative process is repeated until the termination condition is satisfied.

4 Simulation Results

In order to verify the reliability and effectiveness of the proposed method, IEEE 6 and 15 units systems [13] are used for testing the proposed performance. The system of 6-units is an ieee30bus system, and the total load demand is 1263 MW. In the model calculation, the climbing rate limit, network transmission loss, and forbidden zone constraints are considered most. The total load demand of 15 unit system is 2630 MW, and the actual constraints such as forbidden area constraint, climbing rate limit, and network transmission loss are taken into account. However, others some other units, e.g., 2, 5, and 6, have three prohibited areas, and unit 12 has two. The decision interval of the system is a nonconvex decision space with 210 convex intervals. The existence of the transmission point effect is considered in the model.

Fig.2 shows the graph of the daily power generating distribution for the test system of 6 units.

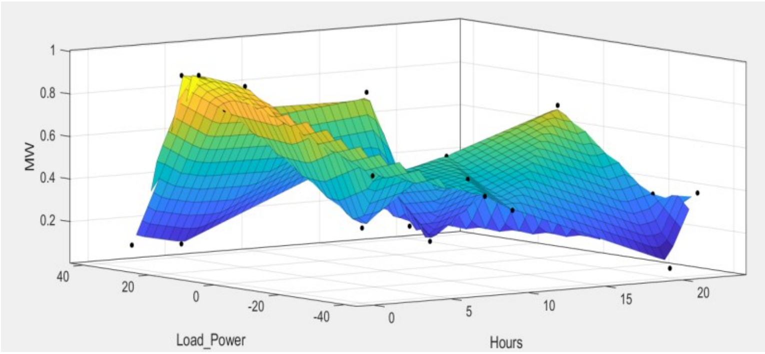

Fig. 2. The graph of the daily power generating distribution for the test system of 6 units.

Table 1. Comparison of results of different algorithms (for IEEE 6 unit system)

Units output	**MFO**	GA[4]	PSO[5]	DE[6]	GWO[9]
P_1	339.91	306.47	356.65	286.44	353.08
P_2	331.18	364.00	329.30	301.02	326.01
P_3	88.31	104.00	102.41	92.96	101.39
P_4	104.00	16.00	97.51	55.28	96.53
P_5	291.21	207.22	198.90	350.88	196.91
P_6	268.36	368.00	244.59	268.50	242.15
Total power/MW	**1275.45**	1276.01	1276.03	1276.95	1276.40
Network loss/MW	**12.45**	12.96	13.02	12.96	12.44
Min cost ($ / h)	**15 433.07**	15 450.00	15 459.00	15 449.77	15 443.82
Max cost ($ / h)	15 443.08	15 492.00	15 524.00	15 449.87	16 449.87
Average cost($/h)	**15 443.07**	15 454.00	15 469.00	15 449.78	15 446.95
Std. deviation($/h)	0.0010	0.0011	0.0012	0.0013	0.582 2
Max deviation($/h)	0.21	42.00	5.00	2.11	0.98

The parameters of the MFO algorithm are set as follows: the number of moth population is 40; when applied to three different examples, the maximum iteration times are 2000 [11]. In Eq. (12), B is a constant 1, t is a random number in the range [- 1 to – 2]. The penalty factor is 500. The maximum number of iterations is 50, and the allowable value of unbalanced power deviation is 0.01 MW.
The obtained results of the proposed scheme are compared with the other previous methods, e.g., GA[4], PSO [5], DE [6], and gray GWO [9]. The number of search agents individual is uniformly set to 40 in all algorithms during the simulation, and the maximum number of iterations is 2000. Each instance is run separately multiple times to ensure the tests' effectiveness, comparability, and robustness. For example, the parameters of setting for 15 machines test system, e.g., system prohibited generating units, the total load demand, and the coefficients B: $[B_{ij}, B_{oi}, B_{oo}]$ power factors.

Tables 1 and 2 show the comparison of the suggested MFO with the GA, PSO, DEand GWO[9] for test systems of the IEEE30 buses with 6 and 15 units, respectively; where the $P_1, P_2, \ldots P_n$ are the generator output power of each branch, n is number of units of the solution. The values in the Tables are the average value, minimum value, maximum value, maximum deviation (the difference between the maximum generation cost and the minimum generation cost) and standard deviation of the generation cost of the 6, and 15 generators systems. It can be seen from Table 1 that for a 6-machine system, the best result is \$15433.07/h, the worst result is \$15453.08/h, the average result is \$15443.07/h, and the standard deviation is 0.0010. The difference between the worst and best results is only 0.01 \$ / h, indicating that MFO has good robustness.

Table 2. Comparison of results of different algorithms (for IEEE 15-unit test system)

Units output	**MFO**	GA[4]	PSO[5]	DE[6]	GWO[9]
P_1	424.88	383.09	445.81	358.05	441.36
P_2	413.97	455.00	411.63	376.28	407.51
P_3	110.38	130.00	128.02	116.20	126.74
P_4	130.00	20.00	121.88	69.10	120.67
P_5	364.01	259.03	248.62	438.59	246.13
P_6	335.45	460.00	305.74	335.62	302.68
P_7	338.10	465.00	202.33	376.97	200.31
P_8	171.00	60.00	220.81	144.09	218.60
P_9	122.87	148.02	152.46	73.77	150.93
P_{10}	89.84	42.97	154.58	142.02	153.03
P_{11}	53.69	73.96	72.45	61.42	71.73
P_{12}	26.19	45.37	52.16	46.55	51.64
P_{13}	31.40	25.00	74.27	65.57	73.53
P_{14}	29.22	37.00	41.06	36.35	40.65
P_{15}	29.69	52.99	44.52	29.66	44.07
Total power/MW	**2660.08**	2662.40	2668.40	2662.29	2660.36
Network loss/MW	**30.08**	32.43	38.28	32.28	30.36
Min cost (\$/h)	**32697.15**	32858.00	33063.54	32 751.39	32732.95
Max cost (\$/h)	**33398.04**	33331.00	33337.00	32945.00	32756.01
Average cost(\$/h)	**32727.95**	33039.00	33228.00	32756.01	32735.06
Std. deviation(\$/h)	**0.0293**	0.09	0.81	0.05	0.36
Max deviation(\$//h)	**0.89**	473.00	273.46	193.61	23.06

It can be seen from Table 2 that for a 15 machine system, the minimum, maximum, and average cost of the 50 test results of the MFO algorithm are 32697.15 \$ / h, 33398.04 \$ / h, and 32727.95 \$ / h, respectively, and the standard deviation is 0.0293. Compared with other algorithms, MFO still shows efficient and stable optimization ability, and the maximum deviation is only 0.89 \$ / h, which indicates that the MFO algorithm has good robustness and stability.

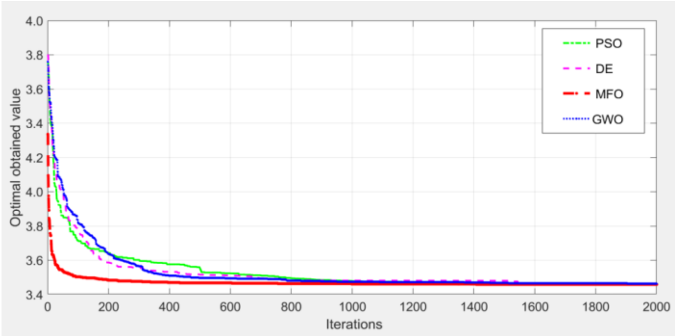

Fig. 3. The comparison obtained result curves of the suggested algorithm with the other algorithms for the test system of 15 units.

Fig. 3 shows the comparison obtained result curves of the suggested FMO with the other algorithms. e.g., DE, PSO, and GWO for the test systems of 15 units. The observed figure shows that the MFO optimization method has better quality performance in convergence speed and time consumption than PSO and GWO methods. In general, we can say that the MFO can solve the PDP problem in the power system with good robustness and significant economic benefits.

5 Conclusion

This paper suggested a new solution to solving the power generating distribution planning (PDP) problem based on a new swarm-based technique called moth flame optimization (MFO) to reduce power loss and cost consumption. PDP model is a critical component of the power system's financial performance, considering various system operation restrictions that are typical multi-constrained nonlinear optimization problems. The experimental results of the suggested method were compared to the findings of other algorithms, the IEEE 30 buses with 6-unit and 15-unit systems. Compared results show that the proposed MFO produced more optimal and stable derivatives that can effectively handle the PDP problem, resulting in significant cost gains and save fuels.

References

1. Tsai, C.F., Dao, T.K., Pan, T.S., Nguyen, T.T., Chang, J.F.: Parallel bat algorithm applied to the economic load dispatch problem. J. Internet Technol. 17, 761–769 (2016). https://doi.org/10.6138/JIT.2016.17.4.20141014c.
2. Nguyen, T.-T., Wang, M.-J., Pan, J.-S., Dao, T., Ngo, T.-G.: A Load Economic Dispatch Based on Ion Motion Optimization Algorithm BT - Advances in Intelligent Information Hiding and Multimedia Signal Processing. Presented at the (2020).

3. Dao, T.K., Pan, T.S., Nguyen, T.T., Chu, S.C.: Evolved Bat Algorithm for solving the Economic Load Dispatch problem. In: Advances in Intelligent Systems and Computing. pp. 109–119 (2015). https://doi.org/10.1007/978-3-319-12286-1_12.
4. Warsono, King, D.J., Özveren, C.S., Bradley, D.A.: Economic load dispatch optimization of renewable energy in power system using genetic algorithm. In: 2007 IEEE Lausanne POWERTECH, Proceedings (2007). https://doi.org/10.1109/PCT.2007.4538655.
5. Safari, A., Shayeghi, H.: Iteration particle swarm optimization procedure for economic load dispatch with generator constraints. Expert Syst. Appl. 38, 6043–6048 (2011).
6. Noman, N., Iba, H.: Differential evolution for economic load dispatch problems. Electr. power Syst. Res. 78, 1322–1331 (2008).
7. Nguyen, T.-T., Wang, H.-J., Dao, T.-K., Pan, J.-S., Liu, J.-H., Weng, S.-W.: An Improved Slime Mold Algorithm and Its Application for Optimal Operation of Cascade Hydropower Stations. IEEE Access. 8, 1 (2020). https://doi.org/10.1109/ACCESS.2020.3045975.
8. Nguyen, T.-T., Pan, J.-S., Chu, S.-C., Dao, T.-K., Do, V.-C.: Improved Performance of Wireless Sensor Network Based on Fuzzy Logic for Clustering Scheme BT - Advances in Smart Vehicular Technology, Transportation, Communication and Applications. Presented at the (2019).
9. Pradhan, M., Roy, P.K., Pal, T.: Grey wolf optimization applied to economic load dispatch problems. Int. J. Electr. Power Energy Syst. 83, 325–334 (2016).
10. Nguyen, T.-T., Pan, J.-S., Dao, T.-K.: A Compact Bat Algorithm for Unequal Clustering in Wireless Sensor Networks, (2019). https://doi.org/10.3390/app9101973.
11. Mirjalili, S.: Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based Syst. 89, 228–249 (2015).
12. Mehne, S.H.H., Mirjalili, S.: Moth-flame optimization algorithm: theory, literature review, and application in optimal nonlinear feedback control design. In: Nature-Inspired Optimizers. pp. 143–166. Springer (2020).
13. Chakrabarti, S., Kyriakides, E., Eliades, D.G.: Placement of synchronized measurements for power system observability. IEEE Trans. power Deliv. 24, 12–19 (2008).