A Power System Economic Load Dispatch Using Jellyfish Search Algorithm

Shi-Jie Jiang[1], Trong-The Nguyen[2], Thi-Kien Dao[1],
Van-Dinh Vu[3], Truong-Giang Ngo[4]

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications,
Fujian University of Technology, Fujian Province 350118, China
[2]Haiphong University of Management and Technology, Haiphong, Vietnam
[3]Information Technology Faculty, Electric Power University, Hanoi, Vietnam
[4]Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam
j530216317@qq.com, vnthe@hpu.edu.vn, jvnkien@gmail.com,
dinhvv@epu.edu.vn,giangnt@tlu.edu.vn

Abstract. Economic load dispatch (ELD) is an important measure to achieve energy-saving and consumption reduction in power systems. This study suggests a solution to the ELD problem based on a new intelligent optimization algorithm called the Jellyfish search algorithm (JSA). Due to the imitation of jellyfish's behavior searching for food in the ocean, the JSA owes the advanced characteristics, e.g., simple structure, fast search, and easy to implement. The ELD problem is mathematically expressed as a typical multi-constraint nonlinear optimization problem that can be dealt with by optimizing the JSA algorithm successfully. In order to verify the feasibility and effectiveness of the proposed scheme, two different case calculation studies are used to test and analyze the optimization performance of the proposed method from the multi-dimensional perspectives of the economy. The validation results show the proposed scheme provides more rapidity, convergence, and robustness than the other comparative methods.

Keywords: Jellyfish search algorithm; Electric power generating plant outputs; Economic load dispatch

1 Introduction

The goal of optimizing the economic load distribution (ELD) problem of the power system is to reasonably allocate a load of each unit in a power system so that the power generation cost is minimized under meeting the load and operation constraints [1][2]. Such problems have the characteristics of high dimensionality, discreteness, and nonlinearity[3]. Therefore, traditional linear programming methods, dynamic programming methods, and other methods are difficult to solve such problems virtually [4]. With artificial intelligence technology development, many intelligent algorithms are generated as called metaheuristics algorithms [5] such as a genetic algorithm [6], particle swarm algorithm [7], differential evolution algorithm [8], etc. These metaheuristics algorithms have been applied to solve the ELD problems successfully [3][9][10].

Jelly-fish Search Algorithm (JSA) [11] is a brand-new intelligent optimization algorithm that imitates the behavior of jellyfish searching for food in the ocean, with-out additional adjustment algorithm parameters, simple structure, fast search, and easy to implement. Because the intelligent algorithm of the metaheuristic method has better robustness, it can also calculate the net loss and valve point effect ignored by traditional methods, thereby improving its accuracy and practicability.

This paper proposes a solution to the ELD problem based on the JSA algorithm. The least-squares programming is used to model mathematical optimization for the objective optimization function as a typical multi-constraint nonlinear optimization problem. Two different case studies of ELD scenarios that are used to validate the proposed schemes optimization performance.

2 Model of The ELD

2.1 Objective function

The mathematical model with the minimum total power generation cost C of the system as the optimal dispatch target[1][2] [3][12] can be expressed as:

$$C = min \sum_{i=1}^{n} F_i(P_i) \tag{1}$$

In the formula, n is the number of grid-connected generators in the system. P_i is the active power of the i-th generator, and $F_i(P_i)$ is the consumption characteristic curve of the i-th generator, that is, the power generation cost function. It can generally be expressed as:

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

In the formula, α_i、β_i and γ_i are the power generation cost coefficients of the i-th generator.

2.2 Objective function’s Constraints

(1) Power balance constraint

$$\sum_{i=1}^{n} (P_i) = P_{Load} + P_{Loss} \tag{3}$$

In the formula, P_i is the active power of the i-th generator; P_{Load} is the total load demand of the system; P_{Loss} is the total network loss of the system, which is generally obtained by the B coefficient method [1][2].

$$P_{Loss} = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i \, B_{ij} \, P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \tag{4}$$

In the formula, B_{ij}, B_{0i} and B_{00} are network loss coefficients, which are generally constants.

(2) Generator output constraint

$$P_i^{min} \le P_i \le P_i^{max} \tag{5}$$

In the formula, P_i^{min} is the lower limit of the active power of the i-th generator; P_i^{max} is the upper limit of the active power of the i-th generator.

(3) Generator ramp rate constraint

When the output increases, it is:

$$P_i - P_i^0 \le UR_i \tag{6}$$

When the output decreases, it is:

$$P_i^0 - P_i \le DR_i \tag{7}$$

In the formula, P_i^0 is the active power of the i-th generator at the previous moment; UR_i and DR_i are the limit of the increase and decrease of the active power of the i-th generator per unit time, respectively.

The generator output constraint and the slope rate constraint can be written as:

$$max\left(P_i^{min}, P_i^0 - DR_i\right) \le P_i \le min(P_i^{max}, P_i^0 + UR_i) \tag{8}$$

3 Jellyfish Search Algorithm

The jellyfish search algorithm (JSA) imitates the behavior of jellyfish looking for food in the ocean[11]. Initially, the jellyfish move in the direction of the ocean current. As time goes by, a jellyfish group will be built, and each jellyfish will move within the group, including active and passive movements. In this process, the time control mechanism will regulate the switching of these movements. After several runs, the jellyfish will bloom. This is the best stage.

Ocean current

There are a large number of plankton in ocean currents, and jellyfish will follow the ocean current for food. The direction of the ocean current is determined by averaging all vectors from each jellyfish's position to the position of the best jellyfish. Its expression is as follows:

$$\overrightarrow{trend} = \frac{1}{N}\sum(X_{best} - e_c X_i(t)) = X_{best} - e_c\frac{\sum(X_i(t))}{N} = X_{best} - e_c\mu \tag{9}$$

In the formula, $\overrightarrow{trend}$ is the direction of ocean currents, X_{best} is the current best jellyfish position; e_c is the attraction coefficient; μ is the average position of all jellyfish. Set $df = e_c\mu$, df is the difference between the best position of the current jellyfish and the average position of all jellyfish, and then the expression can be abbreviated as:

$$\overrightarrow{trend} = X_{best} - df \tag{10}$$

According to the typical distribution assumption, the possibility of all jellyfish exists within the distance $\beta\sigma$ around the average position μ. Therefore,

$$df = \beta \times \sigma \times r \tag{11}$$

In the formula, β is the distribution coefficient, β=3; σ is the standard deviation; r is a random number between [0,1]. Set $\sigma = r \times \mu$, than

$$df = \beta \times r \times \mu \tag{12}$$

Thus,

$$\overrightarrow{trend} = X_{best} - \beta \times r \times \mu \tag{13}$$

Therefore, the update formula for each jellyfish is:

$$X_i(t+1) = X_i(t) + r \times \overrightarrow{trend} \tag{14}$$

Jellyfish swarm

In the jellyfish group, the jellyfish also exercise mainly passive movement (Type A) and active movement (Type B). In the early stage of colony formation, jellyfish mostly move passively. As time goes by, they will slowly change to active movement.

Type A movement is the movement of a jellyfish around its own position, and the corresponding updated position of each jellyfish is given by the following formula.

$$X_i(t+1) = X_i(t) + \gamma \times r \times (Ub - Lb) \tag{15}$$

Type B movement is to randomly select a jellyfish j except for itself, and determine the direction of movement by the vector of the jellyfish i and the selected jellyfish j. If the amount of food at jellyfish j is better than that at jellyfish i, jellyfish i moves to jellyfish j. If the amount of food at jellyfish j is worse than the amount of food at jellyfish i, jellyfish i will stay away from jellyfish j. Its motion direction formula is as follows:

$$\overrightarrow{Direction} = \begin{cases} X_j(t) - X_i(t), f\big(X_i(t)\big) \geq f\left(X_j(t)\right) \\ X_i(t) - X_j(t), f\big(X_i(t)\big) < f\left(X_j(t)\right) \end{cases} \tag{166}$$

Therefore, the updated position formula of each jellyfish is as follows:

$$X_i(t+1) = X_i(t) + r \times \overrightarrow{Direction} \tag{17}$$

Time control mechanism

Because ocean currents contain many plankton, jellyfish will move with ocean currents and slowly gather into jellyfish groups. When ocean currents change, a new group of jellyfish will form. In the jellyfish group, the jellyfish will switch between passive movement and active movement. At first, it was mainly passive exercise, and then gradually changed to vigorous exercise over time. A time control mechanism is added here to regulate the movement of jellyfish. $C(t)$ is a time control function, which provides a random value fluctuating from 0 to 1 over time.

$$C(t) = \left|\left(1 - \frac{t}{T}\right) \times (2 \times r - 1)\right| \tag{18}$$

In the formula, t is the current iteration time; T is the maximum number of iterations; r is a random number between [0,1].

When $C(t) > 0.5$, the jellyfish moves with the ocean current. When $C(t) \leq 0.5$, the jellyfish moves in the group. $1 - C(t)$ is a function that controls the movement of the jellyfish group. When $\left(1 - C(t)\right) > r$, the jellyfish exhibits passive motion. When $\left(1 - C(t)\right) \leq r$, the jellyfish exhibits active movement. Since $\left(1 - C(t)\right)$ increases from 0 to 1 overtime, the passive movement takes precedence over the active workout at the beginning, and then gradually, the functional training is dominant.

4 Jellyfish Search Algorithm for the ELD problem

The ELD is a problem with discrete, multi-constrained, nonlinear, and other characteristics[12]. Its constraints include equality constraints and inequality constraints. All individuals must satisfy constraints in the solution space. Therefore, before calculating the objective function value of each individual, it needs to be processed to satisfy all constraints. In this paper, a penalty function is used to deal with the constraints, and the objective function is rewritten as:

$$C = min \sum_{i=1}^{n} F_i(P_i) + q(\sum_{i=1}^{n} P_i - P_{Load} - P_{Loss})^2 \tag{19}$$

In the formula, q is the penalty factor, which is always 1000 during the simulation. The necessary steps for the ELD problem based on the JS algorithm are as follows:

Step 1. Set the JS algorithm parameters and ELD model coefficients.

Step 2. Initialize the population. Calculate the fitness value of each jellyfish and get the jellyfish in the optimal global position.

Step 3. Determine the time control function, choose to follow the ocean current, or move in the jellyfish group to update the position of the jellyfish.

Step 4. If you are moving in the jellyfish group, select passive or active movement to update the position of the jellyfish.

Step 5. Calculate the fitness value of each jellyfish and update the jellyfish in the optimal global position.

Step 6. If the algorithm does not meet the optimization end conditions, turn to Step3; otherwise, the optimization process ends.

Step 7. Output the optimal solution.

5 Experimental Results

In order to verify the feasibility and effectiveness of the proposed method, two typical ELD problems with different dimensions of six units and 15 units are used to carry out simulation experiments. To fit the actual ELD problem, the simulation examples all consider the influence of power loss. Compare this algorithm with other algorithms, namely particle swarm optimization (PSO)[3], sine cosine algorithm (SCA)[13], and whale optimization algorithm (WOA)[14]. To ensure fairness, in the simulation process, all algorithms uniformly set the number of search agents to 30, and the maximum number of iterations is 1000. After multiple runs, take the average value to get the simulation result.

A. Case study of six units

Taking a 6-unit system as an example, the total load borne by the generator is 1263MW, and the dimension d is taken as 6. The parameters of the 6-unit test system are shown in Table 1.

Table 1. Coefficients setting for a six-unit system

Units	γ \$/MW^2	β \$/MW	α \$	P_{min} MW	P_{max} MW
1	0.0070	7	240.0	100.0	500.0
2	0.0095	10	200.0	50.0	200.0
3	0.0090	8.5	220.0	80.0	300.0
4	0.0090	11	200.0	50.0	150.0
5	0.0080	10.5	220.0	50.0	200.0
6	0.0075	12.2	190.0	50.0	120.0

The power loss factor B is:

$$B_{ij} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.0010 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.0010 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.0150 \end{bmatrix},$$

$$B_0 = 10^{-3}[-0.3908 \ \ -0.1279 \ \ 0.7047 \ \ 0.0591 \ \ 0.2161 \ \ -0.6635],$$

$$B_{00} = 0.056$$

Table 2 shows the comparison results of this method with PSO, SCA, and WOA is a six-unit system.

The solution has six generator outputs, including P1 to P6. In addition, the total cost of power generation, the real power loss, and the respective calculation time.

Table 2. The best power outputs for a six-generator system

Outputs	PSO[3]	SCA[13]	WOA[14]	**JSA**
P_1	464.9059	460.6460	471.2798	446.4659
P_2	186.2893	176.3008	190.9744	181.1464
P_3	240	240	240	265
P_4	149.9982	133.2271	133.6215	150
P_5	168.3705	180.5467	154.1544	157.7778
P_6	85	105.0286	105	93.9918
Total power output (MW.)	1294.5639	1295.7488	1295.0301	**1294.3820**
Total generation cost ($/h.)	15709.6236	15721.4026	15720.2586	**15700.7846**
Power loss (MW.)	31.5639	32.7366	32.0302	**31.3508**
Total CPU times (sec.)	1.7393295	1.3081535	1.5615661	**1.3050705**

Figure 1a describes the comparison of using JSA and other algorithms (PSO, SCA and WOA) to solve the six-unit system scheduling problem under the same conditions.

B. Case study of fifteen units

Table 3 lists the parameters of the fifteen-unit system. The total load borne by the generator is 2630MW, and the dimension *d* is 15.

Table 3. Coefficients setting for a fifteen-unit system

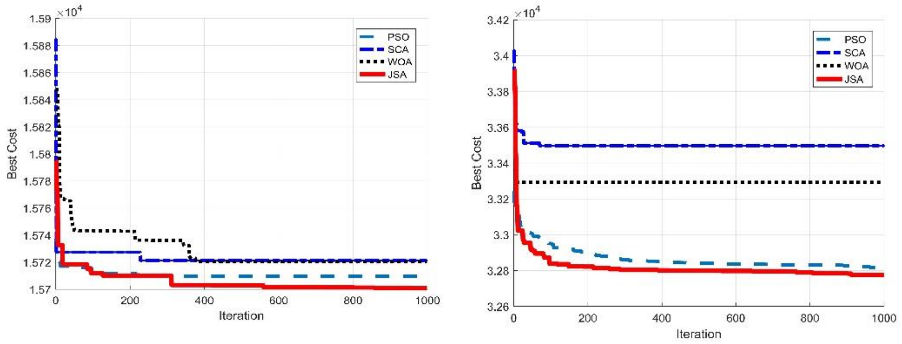
Units	**γ $/MW²**	**β $/MW**	**α $**	**P_{min}MW**	**P_{max}MW**
1	0.000299	10.1	671	150	455
2	0.000183	10.2	574	150	455
3	0.001126	8.8	374	20	130
4	0.001126	8.8	374	20	130
5	0.000205	10.4	461	150	470
6	0.000301	10.1	630	135	460
7	0.000364	9.8	548	135	465
8	0.000338	11.2	227	60	300
9	0.000807	11.2	173	23	162
10	0.001203	10.7	175	23	160
11	0.003586	10.2	186	20	80
12	0.005513	9.9	230	20	80
13	0.000371	13.1	225	25	85
14	0.001929	12.1	309	15	55
15	0.004447	12.4	323	15	55

Table 4 lists the comparison results of the JS algorithm to solve the 15-unit system and other algorithms (such as PSO, SCA, and WOA) under the same conditions. The table contains statistical results of solution, power generation cost, network power loss, and average CPU time.

Table 4. The best power output for a fifteen-generator system

Outputs	PSO[3]	SCA[13]	WOA[14]	**JSA**
P1	453.8913	419.5333	446.3137	455
P2	377.2665	339.4666	373.4275	380
P3	129.9198	111.4374	124.7734	130
P4	129.7971	107.7066	104.9709	129.9865
P5	166.9779	168.9997	157.5359	169.9508
P6	455	430	395	460
P7	429.7784	389.4666	405.9198	430
P8	66.5485	139.7333	134.7823	60
P9	107.6787	143.0415	147.2498	104.8355
P10	113.9893	132.2346	113.0699	131.6601
P11	80	75.3339	70.6940	79.4695
P12	79.9886	67.8400	65	80
P13	25	75.0158	63.7854	25.0435
P14	35.5363	46.4698	44.7592	15
P15	15	46.8933	35.4870	15
Total power output (MW.)	2666.3723	2693.1725	2682.7690	**2665.9459**
Total generation cost ($/h.)	32813.4965	33497.5476	33293.0176	**32773.7831**
Power loss (MW.)	36.0661	63.0077	52.7689	**35.9367**
Total CPU time (sec)	1.8027188	1.2727662	2.0873071	1.3111499

It can be seen from the table that the method used has better quality performance results than other methods in terms of cost, power consumption, and time consumption. Figure 1b describes the comparison of using JSA and different algorithms (PSO, SCA, and WOA) to solve the six-unit system scheduling problem under the same conditions.



a) Six-unit and fifteen -unit system b) Fifteen -unit system scheduling

Fig. 1. The comparison of using JSA and the PSO[3], SCA[13], and WOA[14] methods to the solution to the six-unit and fifteen -unit system scheduling problems under the same conditions

It can be seen from the figure that the quality performance observation results in terms of convergence speed and time consumption show that the optimization method used is superior to other methods.

6 Conclusion

In this paper, we proposed a solution to the economic load distribution problem (ELD) by applying a new metaheuristic algorithm called Jellyfish Search Algorithm (JSA). With the advantages characteristics of the new metaheuristic algorithm of the JSA algorithm, e.g., rapidity, convergence, robustness, and easy implementation, the proposed scheme can have achieved the solution's target. In the experiment section, two calculation examples verify the proposed scheme performance with four angles of economy, rapidity, convergence, and robustness. The preliminary results are compared with other literature (such as Particle-swarm optimization (PSO), Sine-cosine algorithm (SCA), and Whale optimization algorithm (WOA) schemes shows the proposed scheme can effectively consider local search and global search, showing strong competitiveness and effective problem-solving method for the ELD problem.

References

1. Al-Sumait, J.S., Sykulski, J.K., Al-Othman, A.K.: Solution of different types of economic load dispatch problems using a pattern search method. Electr. Power Components Syst. 36, 250–265 (2008). https://doi.org/10.1080/15325000701603892.
2. 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.

3. Park, J.-B., Lee, K.-S., Shin, J.-R., Lee, K.Y.: A particle swarm optimization for economic dispatch with nonsmooth cost functions. IEEE Trans. Power Syst. 20, 34–42 (2005). https://doi.org/10.1109/TPWRS.2004.831275.
4. Rabani, Y.: Linear programming. In: Handbook of Approximation Algorithms and Metaheuristics (2007). https://doi.org/10.1201/9781420010749.
5. Gogna, A., Tayal, A.: Metaheuristics: review and application. J. Exp. Theor. Artif. Intell. 25, 503–526 (2013).
6. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. Computer (Long. Beach. Calif). 27, 17–26 (1994). https://doi.org/10.1109/2.294849.
7. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). pp. 69–73 (1998). https://doi.org/10.1109/ICEC.1998.699146.
8. Storn, R., Price, K.: Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. Science (80-.). 11, 1–15 (1995). https://doi.org/10.1023/A:1008202821328.
9. 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.
10. Nguyen, T.-T., Wang, M.-J., Pan, J.-S., Dao, T., Ngo, T.-G.: A Load Economic Dispatch Based on Ion Motion Optimization Algorithm, Advances in Intelligent Information Hiding and Multimedia Signal Processing. 157, 115-125 (2020).
11. Chou, J.-S., Truong, D.-N.: A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. Appl. Math. Comput. 389, 125535 (2021). https://doi.org/https://doi.org/10.1016/j.amc.2020.125535.
12. 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.
13. Gonidakis, D., Vlachos, A.: A new sine cosine algorithm for economic and emission dispatch problems with price penalty factors. J. Inf. Optim. Sci. 40, 679–697 (2019).
14. Kumar, A., Bhalla, V., Kumar, P., Bhardwaj, T., Jangir, N.: Whale optimization algorithm for constrained economic load dispatch problems—a cost optimization. In: Ambient Communications and Computer Systems. pp. 353–366. Springer (2018).