Journal of Computer and Communications, 2022, 10, 88-103
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

An Improved Differential Evolution Whale Algorithm for Economic Load Distribution

Haiming Li[1*], Chunning Fu[2]

[1]School of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China
[2]School of Computer Science, University of Nottingham, Nottingham Park, Nottingham, UK
Email: *zjxulhm@163.com

How to cite this paper: Li, H.M. and Fu, C.N. (2022) An Improved Differential Evolution Whale Algorithm for Economic Load Distribution. *Journal of Computer and Communications*, **10**, 88-103.
https://doi.org/10.4236/jcc.2022.1010007

Received: April 13, 2021
Accepted: October 24, 2022
Published: October 27, 2022

Abstract

An improved optimization algorithm combining the differential evolution algorithm and the whale algorithm is proposed for the problem of not being able to get rid of the local optimum in the economic load distribution algorithm. The algorithm adopts a nonlinear convergence strategy, a crossover strategy of differential evolution and the introduction of an elimination mechanism, which balances the global search and local exploitation ability of the algorithm and improves the accuracy of the solved optimal solution. The 13-unit and 40-unit systems are selected for economic load distribution calculation, and the experimental results show that the proposed improved algorithm is superior in distributing the economic load of the power system and can effectively reduce the economic cost.

Keywords

Whale Optimization Algorithm, Differential Evolution Algorithm, Elimination Mechanism, Economic Load Distribution

1. Introduction

Economic load dispatch (ELD) refers to the economic distribution of load to all units in the distribution network of a certain area. In other words, the existing resources are used to reliably meet the demand for customers with minimum cost to obtain maximum economic benefits.

Due to the valve point effect of the generating units, the limitation of the distribution network transmission performance differences, and the constraints of the power balance and the operating conditions of the units, the distribution of unit loads often makes the solution region of the optimal solution impossible to widen. Earlier, mathematical conventional optimization methods such as Lagrangian method and dynamic programming method were usually chosen to deal with this type of load distribution problems [1]. Besides, when there are more units in the distribution system, the high-dimensional characteristics exhibited by both ends of the generating units cannot be solved quickly by the basic mathematical optimization methods.

In recent years, many scholars have proposed the use of swarm intelligence algorithms to cope with the difficulties in ELD problems: swarm intelligence algorithms are highly adaptable to the types of objective functions in computation, the uncertainty of computational data and the structure of the search space, and can deal with the effects of grid losses, valve point effects, fuel differences and other factors, making the solution results more in line with engineering reality. Currently, particle swarm optimization (PSO) [2], genetic algorithm (GA) [3], cluster evolutionary algorithm (CEA) [4], differential evolutionary algorithm (DE) [5], etc., have all achieved good results in the field of optimization. However, due to the inherent computational complexity of the ELD problem makes the group intelligent optimization algorithm has the problem of easy to fall into the local optimum or slow convergence, and some of the algorithms have too many parameters, the parameters interfere with each other and other problems, it is difficult to adapt to the flexible and variable problem model, these will to a certain extent affect the quality of the group intelligent algorithm to solve the actual ELD problem, so there is still a need to seek high accuracy, few parameters, good robustness and high efficiency. The algorithm with good robustness and high efficiency is still needed.

Whale Optimization Algorithm (WOA) [6] has been successfully applied in recent years to short-term power load forecasting [7], production optimization of circuit breakers [8], robust multi-user detection [9], capacity optimization configuration of micro-grid composite energy storage system [10], and optimal scheduling of grid reactive power [11], etc. Although the WOA algorithm has a simple structure and a novel and unique way of finding the optimal solution, it is still unsatisfactory in terms of search speed and accuracy of searching for the optimal solution as other traditional optimization algorithms. The DE algorithm is a typical algorithm with good global convergence and simple operation steps, but the selection of the initial population of the algorithm is unstable, and the improper selection of parameters will cause the algorithm to fall into local weakness at the early stage. However, the selection of the initial population of the algorithm is unstable, and improper selection of parameters will cause the algorithm to fall into local optimum in the early stage.

To this end, the whale optimization and differential evolution algorithms are considered to be improved separately, and then the advantages of each of the two algorithms are selected and combined into an improved differential evolution whale optimization algorithm (Improved Differential Evolution Whale Optimization Algorithm, IDEWOA). The predation and bubble net strategies in the WOA algorithm are first used to update the population information instead of the variation step in DE. Then, the convergence factor in the algorithm is adjusted to a special nonlinear update strategy to enhance the exploration ability of the algorithm. Finally, the crossover and selection links of the differential evolution algorithm are introduced to enhance the population diversity and prevent the premature degradation of convergence accuracy in the process of solving the optimal solution. Meanwhile, the elimination mechanism is adopted in the crossover link to improve the algorithm's efficiency in finding the optimal solution, thus proving the superiority of the algorithm in solving ELD problems.

Based on the above study, the power system loads of Unit 13 and Unit 40 are selected in the paper and simulated and tested with the improved new algorithm for their optimal distribution, and finally the experimental results are compared and analyzed with the standard whale optimization algorithm and standard differential evolution algorithm to verify the superiority of the optimization algorithm.

2. Economic Load Distribution Model

2.1. Objective Function

The solution of the objective function in the economic load distribution model of the power system is actually the minimum value of the generation cost of the entire power system units:

$$\min F = \sum_{i=1}^{m} F_i\left(P_i\right) = \sum_{i=1}^{m} a_i P_i^2 + b_i P_i + c_i \tag{1}$$

In the formula, F is the cost of generating unit; $F_i(P_i)$ is the cost characteristic function of unit i; P_i is the operating power of unit i; a_i, b_i, c_i are consumption characteristic parameters.

2.2. Valve Point Effect

The valve point effect will cause a discontinuity on the consumption curve of the generator set, which is mainly due to the pulling phenomenon caused by the sudden start of the unit's gas valve during operations. The inclusion of valve point effect in the calculation is beneficial to improve the accuracy of the optimal solution of the load distribution problem, so the mathematical model after the introduction of valve point effect is:

$$F_i\left(P_i\right) = a_i + b_i P_i + c_i P_i^2 + E_i \tag{2}$$

$$E_i = \left| d_i \sin\left(e_i\left(P_i^{\min} - P_i\right)\right)\right| \tag{3}$$

E_i is the cost impact caused by the valve point effect of the i-th unit; d_i and e_i are the characteristic constants of the unit; $P_i^{\min}$ is the minimum operating power of the i-th unit.

2.3. Constraints

2.3.1. Power Balance Constraint

$$\sum_{i=1}^{m} P_i = P_D + P_L, i = 1, 2, \cdots, m \tag{4}$$

where, m is the sum of generating units in the system; P_D is the target value of the load in the power system; P_L is the loss value of the power grid. Under a certain distribution network range, when the load in the system is distributed centrally and the coverage is small, the network loss is negligible, so the formula (5) is adjusted to the following form:

$$\sum_{i=1}^{m} P_i = P_D, i = 1, \cdots, m \tag{5}$$

2.3.2. Unit Operation Constraints

$$P_i^{\min} \le P_i \le P_i^{\max}, i = 1, 2, \cdots, m \tag{6}$$

where $P_i^{\max}$ and $P_i^{\min}$ respectively represent the maximum and minimum operating power of the i-th unit, and P_i is the corresponding effective power of the i-th unit.

In summary, the mathematical model of the ELD problem considering the valve point effect, power balance constraint and operating power constraint is:

$$\min F = \sum_{i=1}^{m} F_i\left(P_i\right) = \sum_{i=1}^{m} a_i + b_i P_i + c_i P_i^2 + \left| d_i \sin\left(e_i \left(P_i^{\min} - P_i\right)\right)\right| \tag{7}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{m} P_i = P_D \\ P_i^{\min} \le P_i \le P_i^{\max} \end{cases} \tag{8}$$

It can be seen from the above formula that the mathematical model of the ELD problem contains constraints of nonlinear equality and inequality constraints, the function variables are not differentiable and discontinuous, and the range of feasible solutions sought is non-convex. In order to solve the ELD problem more effectively, this paper proposes to improve the original two algorithms and combine them into a new optimization algorithm.

3. Improved Differential Evolution Whale Optimization Algorithm

3.1. Classic Whale Optimization Algorithm

The WOA algorithm is inspired by the hunting behavior of humpback whales. The main steps of the algorithm are encircling the prey, bubble netting and searching for new prey in three steps, and the algorithm's optimization search process is as follows:

1) Surround the prey

In the stage of encircling the prey, the individual whales of the current population carry out encirclement actions according to the location of the food. The coordinates of the whale with the best fitness in the current iteration are used as the location of the prey, while other whales continuously update their coordinates to encircle the prey, and their positions are updated. The formula is as follows:

$$X_{G+1} = X^* - A \cdot \left| X_G - C \cdot X^* \right| \tag{9}$$

where, X^* is the coordinates of the individual with the optimal solution for this iteration, X_G is the coordinate of the whale population, and A is the convergence factor, which is derived from formula (10):

$$A = 2a \cdot r - a \tag{10}$$

A varies from 2 to 0 using a linear decreasing strategy. C is a random number representing the oscillatory behavior of whale predation and is given by formula (11):

$$C = 2 \cdot r \tag{11}$$

In formula (10) and (11), r is expressed as a constant of (0,1).

2) Bubble net predation

The predation stage of the bubble net in WOA is mainly composed of two strategies: spiral blowing bubbles and enveloping predation. The spiral motion moves according to formula (12):

$$X_{G+1} = D \cdot \mathrm{e}^{bl} \cdot \cos 2\pi l + X^* \tag{12}$$

In the formula, $D = \left| X^* - X_t \right|$ represents the distance between the individual and the target prey, b is the helix constant, generally taken as 1, used to define the whale movement mode, and the value of l is [−1, 1].

In WOA, the spiral movement of the whale and the encircling predation are carried out at the same time, so it is usually assumed that the whale has half chance to choose one of the ways to approach the prey. The coordinate iterative expression is shown in formula (13):

$$X_{G+1} = \begin{cases} X_G - A \cdot \left| X_G - C \cdot X^* \right| & \text{if } p < 0.5 \\ D \cdot \mathrm{e}^{bl} \cdot \cos(2\pi l) + X^* & \text{if } p \geq 0.5 \end{cases} \tag{13}$$

where p is a random number.

3) Search for new prey

When $|A| < 1$, the whale continues to carry out enveloping search and changes its own coordinates according to formula (9); when $|A| \geq 1$, the whale performs a random global search to prevent local optimality. The coordinate iteration expression is as follows:

$$D' = \left| X_G - C \cdot X_{rand} \right| \tag{14}$$

$$X_{G+1} = X_{rand} - A \cdot D' \tag{15}$$

where X_{rand} represents any whale in the population, and D' is the distance between the whale and the better prey it finds.

3.2. Classical Differential Evolution Algorithm

1) Mutation link

Among all the individuals $X_{i,G} \left(i = 1, 2, \cdots, N \right)$ of the current population G, three different individuals are selected as variant individuals:

$$V_{i,G+1} = X_{r1,G} + F * \left(X_{r2,G} - X_{r3,G} \right) \tag{16}$$

where r_1, r_2, r_3 represent random natural numbers in [1, N], and satisfy $r_1 \neq r_2 \neq r_3 \neq i$, F is the scaling factor, and $F = 0.5$ in this paper.

2) Cross link

The crossover link is to ensure that the crossover population $U_{i,G+1}(i = 1, 2, \cdots, N)$ can contain at least one mutant individual, and the following strategies are used to ensure the contribution rate of the mutant individual $V_{i,G}$:

$$U_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & rand(j) \leq CR \text{ or } j = rnb(i) \\ x_{i,j,G+1} & rand(j) > CR \text{ or } j \neq rnb(i) \end{cases} \tag{17}$$

In the formula, the value range of $rand(j)$ is [0, 1]; $rnb(i)$ represents a random integer in [1, N]; CR is the crossover probability factor, which is used to control the degree of crossover. This paper takes $CR = 0.9$.

3) Select link

After the crossover link, the individual $U_{i,G+1}$ is compared with the current individual $X_{i,G}$ to compare the advantages and disadvantages, and the individuals with better fitness are selected and placed into the next generation population.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_G) \\ X_G & \text{if } f(U_{i,G+1}) > f(X_G) \end{cases} \tag{18}$$

3.3. Improved Differential Evolution Whale Optimization Algorithm

In the paper, DE, which has strong exploration capability, is integrated into WOA to improve WOA's ability to develop the algorithm space. At the initial stage of each iteration, WOA uses bubble net and prey search mechanism to preprocess the variation of the location of population individuals to generate an initialized population, and then DE crosses and selects this population to carry out the population update iteration. The specific improvement method is as follows:

3.3.1. Constraint Processing

Considering the high complexity of the power balance constraint in the high-dimensional case, the paper proposes a strategy that takes both the operation constraint and the power balance constraint into account, as follows:

Step 1 Set the population serial number as $i(i = 1, 2, \cdots, N)$, and N is the number of population.

Step 2 Calculate the sum of power P_{sum} of all units in the current population i and the total load value P_D of the system demand, and calculate the difference. If the difference is not 0, continue to the next step; if it is 0, skip to Step 5.

Step 3 Generate a random serial number r (r is any integer within [1, m]), and set the active power P_i of the unit r under this serial number to $P_i = P_i - (P_{sum} - P_D)$.

Step 4 Check P_i's unit operation constraints, if it exceeds the upper and lower limits, set them as the upper and lower limits. Jump to Step 2. Re-verify whether the difference is 0.

Step 5 $i = i + 1$ Constraint and adjust the units of all populations in turn, until the difference between the P_{sum} of all populations and the total load value P_D of the system demand is 0.

3.3.2. Whale Optimization Algorithm Replaces the Mutation Part of Differential Evolution Algorithm

$$W_{i,G+1} = \begin{cases} X_G - A \cdot \left| X_G - C \cdot X^* \right| & \text{if } p < 0.5 \text{ and } \left| A \right| < 1 \\ X_{rand} - A \cdot \left| X_G - C \cdot X_{rand} \right| & \text{if } p < 0.5 \text{ and } \left| A \right| \geq 1 \\ D \cdot e^{bl} \cdot \cos(2\pi l) + X^* & \text{if } p \geq 0.5 \end{cases} \tag{19}$$

where $W_{i,G+1}$ are a new generation of mutant populations. That is, $W_{i,G+1}$ is used instead of $V_{i,G+1}$ for the next step of crossover operation.

At the same time, improve the mutation operation of the whale optimization algorithm: the mutual coordination of the exploration and exploitation capabilities of WOA guides the whole algorithm. Among them, the exploration capability refers to the ability to search a wider range of regions and jump out of local optimum; the exploitation capability refers to the ability to perform local search on regions near known solutions, which is beneficial to accelerate the convergence speed.

3.3.3. Improvement of Convergence Factor

From literature [8] and literature [12], it can be seen that the convergence factor a in the WOA algorithm has a crucial impact on the exploration and exploitation ability of the algorithm: the larger the convergence factor a, the stronger the global search performance of the algorithm; if the convergence factor a is smaller, the local search performance of the algorithm will be stronger and the convergence rate will be higher. In the standard WOA algorithm, as the iteration progresses, the linearly decreasing method of the convergence factor a allows the algorithm to have better global search capabilities in the early stage, but the convergence speed is too slow; and when the algorithm is about to reach the maximum number of iterations. The convergence factor a approaches the zero point, and its convergence speed has accelerated, but it has not been able to get rid of the dilemma of local optimality, and it has not achieved a good coordination of exploration and exploitation. Therefore, the convergence factor in the original WOA algorithm cannot effectively utilize the superiority of the algorithm.

There are several nonlinear update strategies in existing studies. In the literature [9], the original convergence factor is proposed to be replaced by a trigonometric function with the following formula:

$$a = 2 - 2\sin\left(\frac{\frac{\pi}{2}t}{T}\right) \tag{20}$$

where, t and T represent the current number of iterations and the maximum number of iterations respectively, the same below. Literature [13] proposes the use of segmented convergence factor update strategy, the formula is as follows:

$$a=\begin{cases} 2-\left(\frac{t}{T}\right)^2 & t\le T/2 \\ 1-\frac{2\left(t-\frac{t}{T}\right)}{\frac{T}{2}}+\left(\frac{t-\frac{T}{2}}{\frac{T}{2}}\right)^2 & t>T/2 \end{cases} \tag{21}$$

The literature [14] proposes a convergence factor update strategy in the form of a positive exponential with e as the base, the formula is as follows:

$$a=2-\frac{2\left(e^{\frac{t}{T}}-1\right)}{e-1} \tag{22}$$

This paper proposes an adaptive convergence factor update strategy in the form of a negative exponent with a constant as the base:

$$a=2*\mu^{-\frac{t}{T}} \tag{23}$$

where, μ represents a constant, and its value is adjusted according to the actual situation of the calculation example. After experimental testing, $\mu = 1000$ is the best.

The above four strategy functions are plotted in the same graph as the linear decreasing strategy, as shown in **Figure 1** (take the number of iterations T = 2000).

It can be seen from **Figure 1** that compared with the other strategy functions,

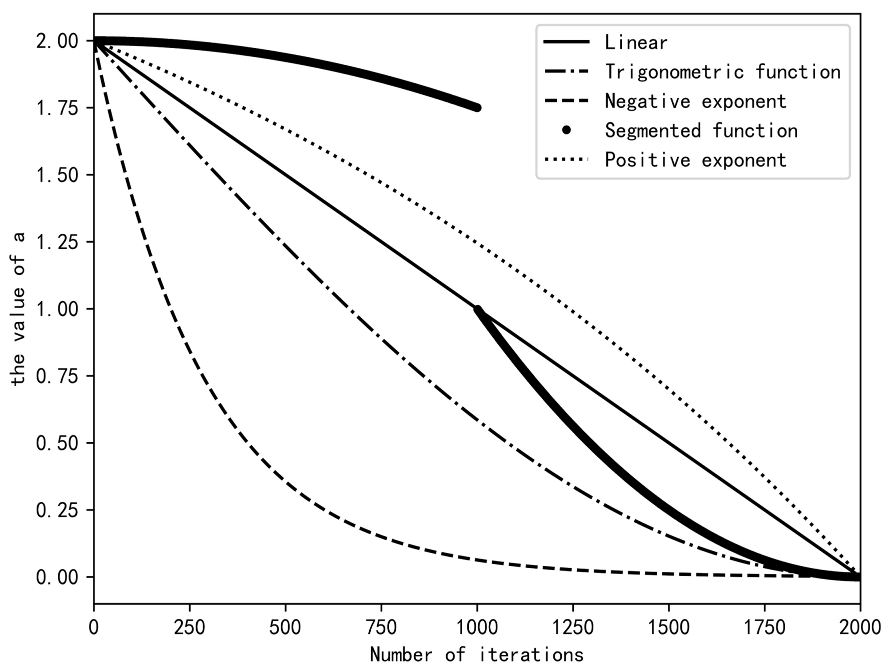

Figure 1. Curve change graph of convergence factor *a*.

the negative exponential decreasing update strategy maintains a larger value of a in the early iterative stage of the algorithm, which gives the algorithm more chances to jump out of the local extremes; in the middle of the algorithm, the function monotonically decreases more than the other four strategies, and the convergence factor a can be plunged to a smaller value, thus effectively ensuring the high efficiency of convergence; while the search for the optimal solution in the later stage. The range of the optimal solution is basically determined, and the convergence factor a in this strategy is in a small value and decreases slowly, so the final convergence accuracy of the algorithm is guaranteed.

3.3.4. Introduction of Cross-Operation and Elimination Mechanism

Improve the crossover operation of differential evolution: The size of the crossover factor CR largely determines the convergence performance of the differential evolution algorithm. It can be seen from formula (17) that the larger the CR value, the higher the contribution rate of V_i to U_i, which means that the population will contain more variant individuals after the crossover operation; the smaller the value of CR, the contribution rate of X_i to U_i, the lower the value, the poorer the algorithm's ability to exploit other regions of the solution, and the poor convergence efficiency, but it is conducive to retain the original individual characteristics (maintaining the diversity of the species), which makes the algorithm have a local solution during higher success rate.

Therefore, if the algorithm want to maintain the population diversity more stably in the early stage, it should start from a small CR value to maintain the diversity and success rate in the early stage, and then gradually increase the CR to enhance the convergence rate in the later stage, so as to effectively reduce the number of falling into local optimum in the whole iteration, which is the more ideal CR change rule.

In addition, this paper proposes to screen mutated individuals in advance in the crossover operation, so as to provide more opportunities for superior individuals in the population; while inferior individuals are given a lower probability of selection and are gradually eliminated. Therefore, this paper proposes an elimination mechanism based on individual fitness (as an example of solving for the minimum):

$$CR = \begin{cases} CR_{\min} + \left(CR_{\max} - CR_{\min}\right) \cdot \dfrac{f_i - f_{av}}{f_{\min} - f_{av}} & \text{if } f_i < f_{av} \\ CR_{\min} & \text{otherwise} \end{cases} \tag{24}$$

$CR_{\max}$ and $CR_{\min}$ are the maximum and minimum values of CR, respectively. At the same time, each iteration calculates the average f_{av} of the individual fitness of all populations and finds the best fitness $f_{\min}$. Taking finding the global minimum as an example, if the average f_{av} of the population fitness is lower than the individual's current fitness f_i, this individual can be regarded as a dominant individual. The crossover rate CR of the individual will be adaptively adjusted according to the degree of close to optimal fitness. If the average fitness f_{av} is higher than the individual's current fitness f_p then this individual can be regarded as a disadvantaged individual. The individual's CR will be reset to $CR_{\min}$. Therefore, the superior individuals will continue to be retained to the next generation, and the inferior individuals will gradually decrease due to the low probability of being selected.

Under this method, individuals with better fitness will be retained, and CR will gradually approach the maximum parameter value $CR_{\max}$, thus conforming to the best change law of CR.

3.3.5. Normal Selection Process

The flow chart of the IDEWOA algorithm for solving the economic load distribution model of the power system is shown in Figure 2.

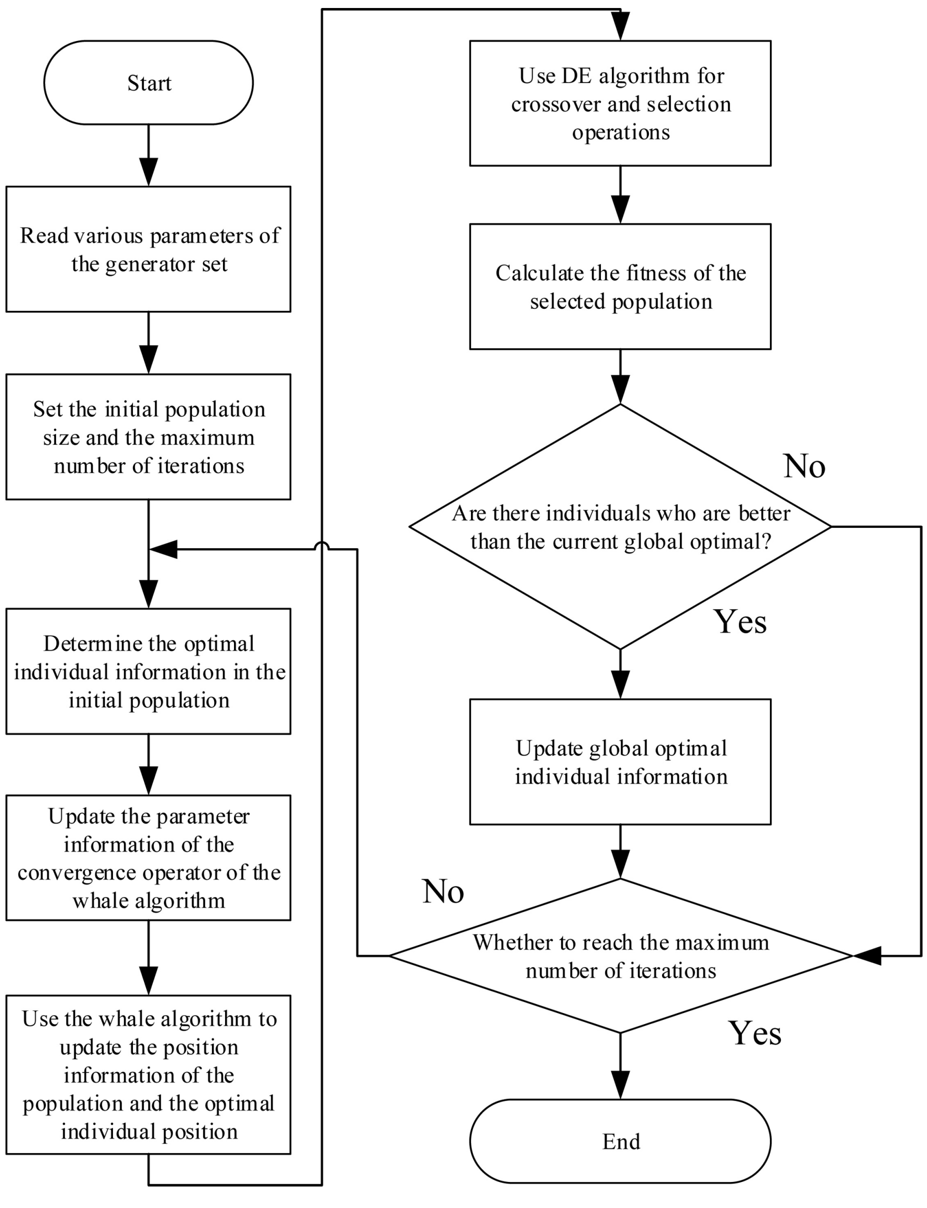

Figure 2. Flow chart of IDEWOA algorithm.

4. Simulation and Analysis

In order to test the ability of the IDEWOA algorithm to optimize the economic load distribution of the power system, the standard DE, standard WOA and IDEWOA algorithm were used to optimize the economic load of the power system of 13 units and 40 units of IEEE, and the valve point effect was introduced. And due to the excessive number of units, grid losses were not considered for the time being, and the basic parameter settings of the system were taken from the literature [15].

In order to avoid the problem of algorithm randomness, the three algorithms are independently executed 40 times. **Figure 3** and **Figure 4** show the convergence curves of the three algorithms for the 2 test cases respectively, **Table 1** and **Table 2** are the statistical results of the various algorithms for the two test cases.

Table 1. Comparison of results of 13 units.

Algorithm	Average value	Max	Min	Standard deviation
PSO [16]	18256.23	18489.98	18095.90	114.00
SA [16]	18160.31	18337.76	18050.25	76.00
CS [16]	18058.90	18108.19	17997.51	25.10
DE	18093.86	18118.19	18027.29	22.66
WOA	18052.73	18097.38	17977.50	42.77
IDEWOA	17972.89	17972.96	17972.84	0.03

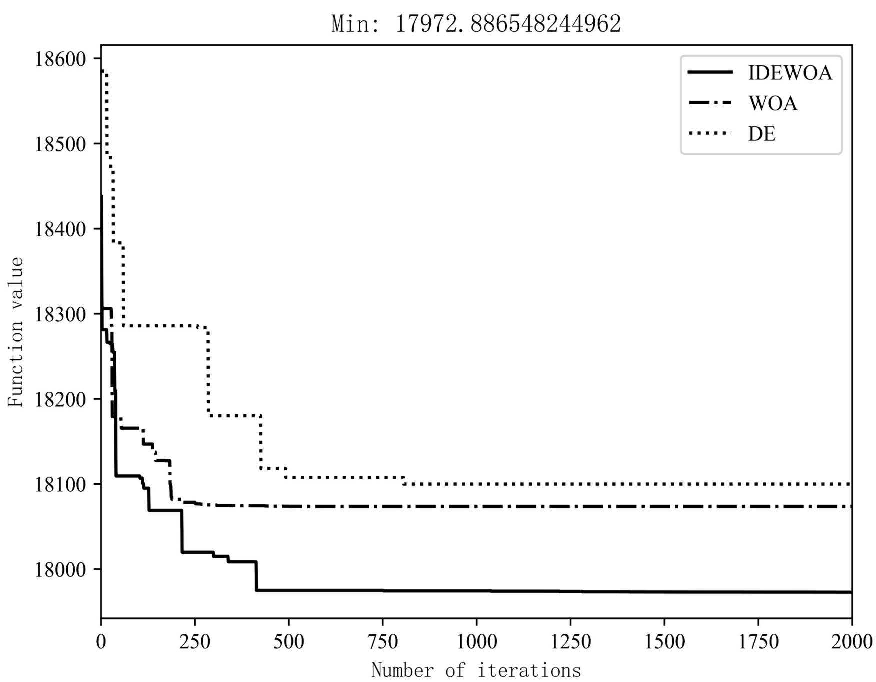

Figure 3. Comparison of operation of 13 units.

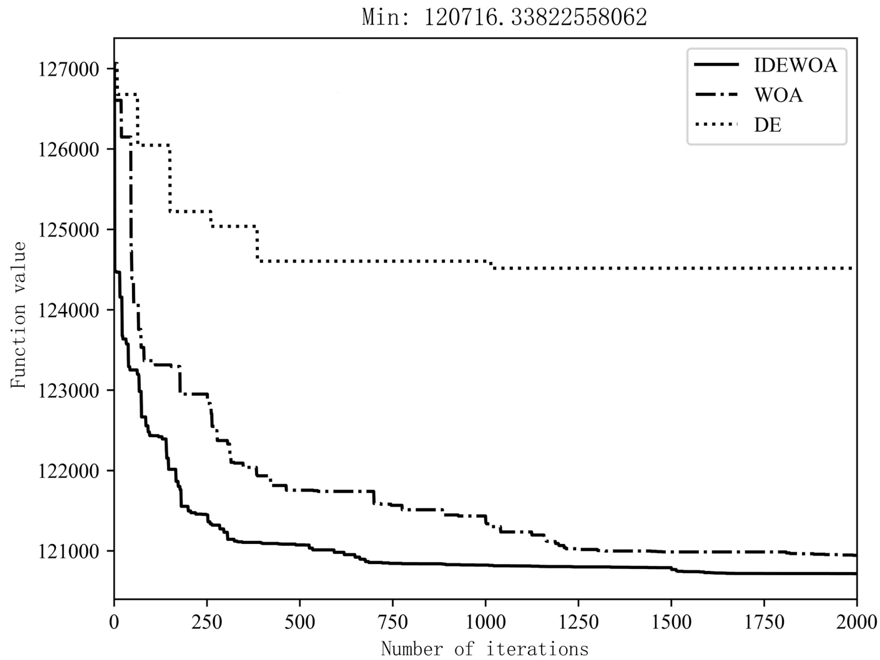

Figure 4. Comparison of operation of 40 units.

4.1. Load Distribution of 13 Units

The required load value for the 13-unit system is $D = 1800$ MW, the dimension is 13, the population size is 65 (generally set to be 5 - 10 times the dimension), $CR_{\max}$ and $CR_{\min}$ are respectively 1 and 0.5, and the number of iterations is 2000.

The simulation results of IDEWOA are compared with other algorithms as shown in **Table 1**. Among them, the DE, WOA and IDEWOA were run 40 times on Python 3.7, and the results of the operations obtained were counted and analyzed. The simulation results of PSO, SA and CS were referred from the study of literature [16]. The comparison of the results of the IDEWOA algorithm with the basic DE and basic WOA algorithms for 13 units is shown in **Figure 3**. The results selected in the figure are the results of the better one run of the IDEWOA algorithm.

From the curves in **Figure 3** and the data in **Table 1**, it can be seen that the IDEWOA algorithm can achieve more superior results than other standard DE, WOA and other similar swarm intelligence algorithms, with significant improvement in convergence speed and higher search accuracy. The data in **Table 1** shows that the IDEWOA algorithm has a running average of 17972.89, which is significantly better than the optimal solutions of other algorithms, and the standard deviation of the experiment is only 0.03, which proves that the solution sought by the algorithm is more stable and more robust. From **Figure 3**, it can be seen that the WOA and DE algorithms are more obvious in their tendency to fall into early maturity in the early stage, while IDEWOA can jump out of the local optimal solution region several times in a short period after improvement, and achieve a solution with high accuracy before the middle of the algorithm. To sum up, IDEWOA can effectively reduce coal consumption and economic cost in the load distribution problem of 13 units.

Table 3 shows the results of the simulation experiments obtained by the IDEWOA algorithm. The optimal solution obtained under this condition is 17972.84.

4.2. Load Distribution of 40 Units

In order to test the stability of the IDEWOA algorithm in a higher-dimensional environment, this paper chooses IEEE 40 units with 300 nodes as the experimental object. In this example, the number of units (dimensions) is large, which increases the difficulty of the solution and extremely tests the algorithm's high-dimensional solution ability. The required load value of the system is $D = 10{,}500$ MW, the dimensionality is 40, and the population size is five times the dimensionality, that is, 200. $CR_{\max}$ and $CR_{\min}$ are respectively 1 and 0.5, and the maximum number of iterations is $t_{\max} = 2000$.

The simulation results of IDEWOA are compared with the other five algorithms as shown in Table 2. The simulation results for PSO, SA and CS are referenced from the study in literature [16]. 40 runs of the IDEWOA algorithm with 40 units are shown in Figure 4 compared with the basic DE and basic WOA. The results of one of good runs of the IDEWOA algorithm are selected in the figure.

From the curve in Figure 4 and the data in Table 2, it can be seen that in the high-dimensional calculation examples, the IDEWOA algorithm has significantly stronger optimization configuration capabilities than other algorithms. The results

Table 2. Comparison of results of 40 units.

Algorithm	Average value	Max	Min	Standard deviation
PSO [16]	126149.57	128477.99	123980.64	1240.00
SA [16]	123588.41	124419.54	122437.56	537.00
CS [16]	122661.26	122945.44	122373.79	165.00
DE	124636.48	125051.93	124240.46	281.88
WOA	120883.26	120988.60	120858.82	38.37
IDEWOA	120802.09	120810.37	120645.74	6.45

Table 3. Power distribution of 13 units.

Unit	Power	Unit	Power	Unit	Power
P_1	628.32	P_6	60.0	P_{11}	40.0
P_2	224.36	P_7	60.0	P_{12}	40.0
P_3	297.61	P_8	60.0	P_{13}	55.0
P_4	109.85	P_9	60.0		
P_5	109.86	P_{10}	60.0		

Table 4. Power distribution of 40 units.

Unit	Power	Unit	Power	Unit	Power
P_1	114.0	P_{15}	304.5203	P_{29}	10.0078
P_2	114.0	P_{16}	304.5777	P_{30}	97.0
P_3	120.0	P_{17}	489.4332	P_{31}	190.0
P_4	180.0289	P_{18}	489.3492	P_{32}	190.0
P_5	97.0	P_{19}	511.3778	P_{33}	190.0
P_6	140.0	P_{20}	511.3613	P_{34}	200.0
P_7	300.0	P_{21}	525.07	P_{35}	200.0
P_8	300.0	P_{22}	525.30	P_{36}	200.0
P_9	300.0	P_{23}	526.7075	P_{37}	110.0
P_{10}	135.0267	P_{24}	524.6055	P_{38}	110.0
P_{11}	94.0655	P_{25}	528.2533	P_{39}	110.0
P_{12}	94.0118	P_{26}	523.4891	P_{40}	511.5312
P_{13}	214.7950	P_{27}	10.0119		
P_{14}	394.3670	P_{28}	10.1016		

of the simulation experiment show that the average, maximum, and minimum values of the optimal values obtained by the IDEWOA algorithm are the lowest among the listed algorithms, and its convergence performance is excellent, and its standard deviation of 6.45 is also lower than other algorithms, indicating that the IDEWOA algorithm can still maintain better robustness in high dimensions.

Table 4 shows the results of the simulation experiments obtained by the IDEWOA algorithm. The optimal solution obtained under this condition is 120 645.74.

From the above calculation examples, it can be found that the IDEWOA algorithm has better global convergence ability than other algorithms, and it has improved in both the speed of seeking and the accuracy of convergence, and it still maintains good stability under the high-dimensional conditions, and its superiority is obvious.

5. Conclusion

This paper proposes an improved differential evolution whale optimization (IDEWOA) algorithm to be applied to the optimal distribution of economic load in the power system. The algorithm can solve non-convex, nonlinear constraints, and high-dimensional optimization problems. First, in the period of searching and encircling prey in the WOA, the original linear convergence factor is adjusted to make it more suitable for the optimization of the entire algorithm while taking into account global exploration and local development capabilities. Then combined with the crossover and selection strategy of the DE algorithm, the

IDEWOA introduces an adaptive elimination mechanism to speed up the convergence of the algorithm and promote the process of searching for the optimal solution. Finally, the algorithm is applied to the optimization test of the load distribution of 13 and 40 units in the power system. The test results verify the superiority and effectiveness of using IDEWOA to solve the ELD problem. The next step is how to apply the IDEWOA algorithm to the field of multi-objective optimization and how to improve the control parameters in the algorithm.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] He, J.H. (2016) A Research of Thermal Power Unit Dispatch Problem Based on Differential Evolution Algorithm. Master's Thesis, Shenyang University of Technology, Shenyang.

[2] Gaing, Z.L. (2003) Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints. *IEEE Transactions on Power Systems*, **18**, 1187-1195. https://doi.org/10.1109/TPWRS.2003.814889

[3] Ciornei, I. and Kyriakides, E. (2013) Closure on "A GA-API Solution for the Economic Dispatch of Generation in Power System Operation". *IEEE Transactions on Power Systems*, **28**, 571. https://doi.org/10.1109/TPWRS.2012.2236481

[4] Chen, H., Pan, X.Y. and Zhang, J. (2016) A Cluster Evolutionary Algorithm for Power System Economic Load Dispatch Optimization. *Journal of Computer Research and Development*, **53**, 1561-1575.

[5] Storn, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. https://doi.org/10.1023/A:1008202821328

[6] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[7] Li, W., Shi, Q., Sibtain, M., Li, D. and Mbanze, D.E. (2020) A Hybrid Forecasting Model for Short-Term Power Load Based on Sample Entropy, Two-Phase Decomposition and Whale Algorithm Optimized Support Vector Regression. *IEEE Access*, **8**, 166907-166921. https://doi.org/10.1109/ACCESS.2020.3023143

[8] Huang, Y.C. and Zhang, L.B. (2019) Improved Whale Optimization Algorithm and Its Application. *CEA*, **55**, 220-226.

[9] Sun, X.Y., Fan, Z. and Ji, Y.-F. (2019) Robust Multiuser Detection Based on Hybrid Whale Optimization Algorithm. *Science Technology and Engineering*, **19**, 114-119.

[10] Li, L.L., Wang, X., Lang, Y.B., Jia, L.K. and Wang, X. (2019) Capacity Configuration of Micro-Grid Composite Energy Storage System Based on Improved Whale Optimization Algorithm. *Electrical Measurement and Instrumentation*, **56**, 104-110.

[11] Teng, D.Y., Teng, H., Pan, C. and Liu, X. (2018) Multi-objective Reactive Power Optimization Dispatch Based on Improved Whale Optimization Algorithm. *Electrical Measurement and Instrumentation*, **55**, 51-58.

[12] Long, W., Cai, S.H., Jiao, J.J., Tang, M.Z. and Wu, T.B. (2017) Improved Whale Optimization Algorithm for Large Scale Optimization Problems. *Systems Engineer-*

ing-Theory & Practice, **37**, 2983-2994.

[13] Hao, X.H., Song, J.X., Zhou, Q. and Ma, M. (2020) Mixed Strategy Based Improved Whale Optimization Algorithm. *Application Research of Computers*.

[14] Sun, H.X., Zhang, W., Lei, Z.M. and Zhang, H. (2020) Research on Wind Power Consumption Dispatching Based on Improved Whale Optimization Algorithm. *Chinese Technology Letters*, **30**, 298-304.

[15] Sinha, N., Chakrabarti, R. and Chattopadhyay, P.K. (2003) Evolutionary Programming Techniques for Economic Load Dispatch. *International Journal of Emerging Electric Power Systems*, **7**, 83-94. https://doi.org/10.1109/TEVC.2002.806788

[16] He, X.Z. and Huang, J.D. (2016) Economic Load Dispatch of Power Systems Based on Constrained Cuckoo Algorithm. *Computer Engineering*, **42**, 214-219.