A Novel Cuckoo Search Algorithm and Its Application

Ping Liu, Shengjiang Zhang*

Nanchang Vocational University, Nanchang, China
Email: *1038795709@qq.com

Abstract

In this paper, the principle of Cuckoo algorithm is introduced, and the traditional Cuckoo algorithm is improved to establish a mathematical model of multi-objective optimization scheduling. Based on the improved algorithm, the model is optimized to a certain extent. Through analysis, it is proved that the improved algorithm has higher computational accuracy and can effectively improve the global convergence.

Keywords

Cuckoo Search Algorithm, Feature Selection, Infrared Spectrum, Global Convergence

1. Introduction

With the continuous increase of electricity consumption in China, the safe and stable operation of power system has become a hot issue in the power industry. Therefore, the optimal dispatching of power system is the key problem. Up to now, many scholars have studied the optimization scheduling of power system [1] [2] [3] [4] [5], but the current algorithms all have some obvious defects, such as slow calculation speed and relatively fuzzy calculation precision.

In this study, the original Cuckoo algorithm was improved and optimized, and the improved algorithm was used to optimize the established scheduling model, and the superiority of the improved algorithm was verified.

2. Improved Cuckoo Algorithm

2.1. Cuckoo Algorithm

Cuckoo algorithm combines common cuckoo propagation mechanism and levy search method [6]. In the beginning, the algorithm has a good search ability, but

as time goes by, the defects of its search ability are gradually exposed. At the same time, the existing problems also include low search accuracy and slow speed, etc., so it is necessary to improve when solving multi-objective problems. For a D-dimensional optimization problem, D variables are required:

$$X = [x_1, x_2, \cdots, x_d] \tag{1}$$

The position update formula based on Levy is as follows:

$$X_i^{(t+1)} = X_i^t + \alpha \oplus L(\lambda), \ i = (1, 2, \cdots, n) \tag{2}$$

$$L(\lambda) \sim U = T^{-\lambda}, \ 1 < \lambda \leq 3 \tag{3}$$

In the formula, x_i^{t+1} is the position of the nest after update, x_i^t is the current position of the nest, α is the step size, $L(\lambda)$ is the data search path, and u is the standard normal variable.

2.2. Analysis of Mathematical Mechanism of GASCS

Inspired by the law of gravitation, Iranian scholar Emat Rashedi proposed the Gravity Search Algorithm (GSA) in 2009 [7]. The principle of GSA algorithm is: individuals of matter are attracted to each other due to the gravitational action, and the gravitational force moves along the direction of individuals with greater mass. The mass of a substance determines its gravitational pull, and the more massive the substance is, the greater the gravitational pull it will receive. Under the action of external gravity, the position of the material individual changes constantly, and finally, the whole group will gather around the mass of the material individual, which is close to the global optimal solution of the optimization problem. In the universal gravitation algorithm, the fitness value of three attribute description functions defined by mass in theoretical physics is used for reference, namely, active gravitation mass M_a, passive gravitation mass M_P and inertia mass M_i. The position of each material individual corresponds to the solution of the optimization problem, and its universal gravitation and inertial mass M_i jointly determine the fitness value of the corresponding function. In fact, the GSA algorithm controls the evolution of the algorithm by adjusting M_a, M_P and M_i, and promotes the matter individuals in the group to gather near the matter individuals with the greatest gravitational force, so as to search for the optimal solution.

The GSA algorithm model includes mass calculation, gravity calculation, acceleration calculation, velocity updating and position updating of individual matter. The algorithm first initializes the position and velocity of the individual matter within the interspace. As shown in literature [8], let the position and velocity of the m substance individual in the D-dimensional spatial dimension evolving to the R-generation be respectively

$$X_{r,m} = \left(x_{r,m}^{(1)}, x_{r,m}^{(2)}, \cdots, x_{r,m}^{(j)}, \cdots, x_{r,m}^{(d)}\right), \ V_m = \left(v_{r,m}^{(1)}, v_{r,m}^{(2)}, \cdots, v_{r,m}^{(j)}, \cdots, v_{r,m}^{(d)}\right)$$

Among them, $j = 1, 2, 3, \cdots, D$, $x_{r,m}^{(j)}$ and $v_{r,m}^{(j)}$ represent the positional and velocity components of the m individual in the JTH dimension, respectively. The

mass M_i and the gravitation of the individual are determined by the fitness value of the position pair of the individual. Calculate the acceleration of the material individual according to Newton's second law, and update the position $X_{r,m}$ and velocity $V_{r,m}$ of the material individual.

The traditional CS algorithm is based on two searching mechanisms, Levy flight random walk and preferred random walk. Although it has certain convergence performance, it still has some limitations. Therefore, this paper proposes a cuckoo algorithm with gravitational acceleration mechanism. Based on the feature that the gravity search can perceive the global optimal information without learning the change of external environmental factors, the cuckoo nest is assigned with different individual masses, and the optimization process not only follows the Levy flight law, but also follows the law of gravity.

The detailed mechanism of GASCS algorithm is shown in **Figure 1**. *nest* (*m*), *nest* (*s*), and *nest* (*k*) are set as the nest location of the host population evolution respectively, and *nest_g* is the optimal nest location. Moreover, it is assumed that the quality of the host nest meets *nest* (*s*) > *nest* (*m*) > *nest* (*k*). Therefore, the mass of *nest_g* at the optimal nest location satisfies the following mathematical relationship: *nest_g* > *nest* (*s*) > *nest* (*m*) > *nest* (*k*). For the convenience of analysis, $X_{r,m}$ are set to be equivalent to *nest* (*m*), $X_{r,s}$ is set to be equivalent to *nest* (*s*), $X_{r,k}$ is set to be equivalent to *nest* (*k*) and $X_{r,gb}$ is set to be equivalent to *nest_g*. $F_{r,ms}$, $F_{r,mk}$ and $F_{r,mg}$ are set to act on respectively Gravitation of *nest* (*m*) to *nest* (*s*), *nest* (*m*) to *nest* (*k*), and *nest* (*m*) to *nest_g*. The accelerations generated by gravity are a_{ms}, a_{mk} and a_{mg}. $F_{r,m}$ are all the resultant forces acting on *nest* (*m*) and *nest_g* to generate acceleration a_{mg}. Meanwhile, $F_{r,t}$ is the external force acting on $F_{r,m}$ and *nest* (*k*) to generate acceleration a_r. Therefore, the host nest approaches *nest_g*, the most massive nest, under the influence of gravity. According to Newton's Second Law, the rate of acceleration experienced by the host's nest during evolution is determined by both the force exerted on the nest and its own mass. Based on this, the individual newer law (2) and newer law (3)
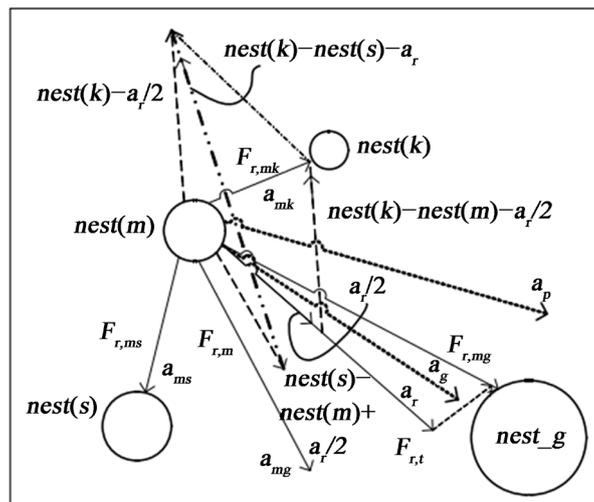

Figure 1. Mechanism of gravitational attraction acceleration.

of the cuckoo algorithm are improved into Equations (4) and (5):

$$X_{r+1,m} = a_r - (\gamma_0 \oplus L(\beta)) * (a_r - X_{r,gb}) \tag{4}$$

$$X_{r+1,m} = X_{r,gb} + p * (X_{r,k} - X_{r,s} - a_r) \tag{5}$$

where, Formula (4) and Formula (5) respectively represent individual positions of Levy flying random walk and preference random walk, $p \in (0,+\infty)$, and a_r is the combined acceleration of $F_{r,t}$. According to Equation (4), it can be seen that, different from the traditional CS algorithm, host nest simply generates candidate solutions according to the Levy flight random walk, the GASCS algorithm takes each host nest as a material individual of different mass, which follows the law of universal gravitation and Newton's second law. As shown in Figure 1, the host nest *nest* (*m*), *nest* (*s*) and *nest* (*k*) move along the direction of the acceleration of a_r and formula (4) step correction term for $-(\gamma_0 \oplus L(\beta)) \cdot (a_r - X_{r,gb})$. $a_g = a_r - (\gamma_0 \oplus L(\beta)) \cdot (a_r - X_{r,gb})$ is obtained from the vector algorithm. Therefore, the algorithm can approximate the global optimal solution *nest_g*.

In summary, due to the position updating of individuals with Levy flying random walk and preference of random walk mode, the combined acceleration a_r is introduced, and the a_r follows algorithm. Therefore the GASCS algorithm can balance the search process of the algorithm and offset the deterioration of algorithm performance caused by unreasonable search step size setting. The proposed algorithm not only takes advantage of the infinite jumping characteristic of Levy flight random walk, which is easy to jump out of the local optimal solution, but also takes advantage of the universal gravitational acceleration to approach the global optimal solution quickly, which overcomes the limitation of CS algorithm.

In view of the imbalance between global search and local search in the limitations of the algorithm, GASCS algorithm proposes a probability mutation method to increase the diversity of population evolution, avoid the algorithm falling into local search and the delay phenomenon in the late execution of the algorithm.

Suppose $X_{r,m} = \left(x_{r,m}^{(1)}, x_{r,m}^{(2)}, \cdots, x_{r,m}^{(j)}, \cdots, x_{r,m}^{(D)} \right)$, and $x_{r,m}^{(j)} \in \left[l^{(j)}, u^{(j)} \right]$. when performing the probability mutation operation, first select the individual population $x_{r,m}^{(j)}$ from $X_{r,m}$ with probability $1/D$, $j \in [1,D]$; Secondly, replace $x_{r,m}^{(j)}$ with $X_{r,m}^{\prime(j)}$ of formula (17), where k is the random integers in $[1,D]$, ξ is a uniformly distributed random number in (0,1), $l^{(j)}$ and $u^{(j)}$ are the upper and lower bounds of $x_{r,m}^{(j)}$; Finally, the candidate solution $X_{r,m}$ is replaced by $X_{r,m}^{\prime}$ by probability mutation and enters into the execution process of the algorithm. Among them, $X_{r,m}^{\prime} = \left(X_{r,m}^{\prime(1)}, X_{r,m}^{\prime(2)}, \cdots, X_{r,m}^{\prime(j)}, \cdots, X_{r,m}^{\prime(D)} \right)$. After the probability mutation, the population evolution can get more diversity, and avoid the problem of low global search efficiency caused by the simplification of the optimization process.

$$X_{r,m}^{\prime(j)} = \left(l^{(j)} + u^{(j)} \right) / 2 + (1-2\xi) * \left(l^{(j)} - u^{(j)} \right) / 2 \tag{6}$$

GASCS algorithm not only uses the characteristics of infinite jump of Levy flight random swimming mode to jump out of local optimal solution easily, but also adopts the diversity of probability variation increasing algorithm to ensure that the algorithm can jump away from local optimal solution. Therefore, GASCS algorithm solves the problem of global search and local search imbalance in CS algorithm.

GSA algorithm mainly includes the following four steps:

1) Calculation of individual mass

According to literature [9], the mass of material individual M is defined as

$$q_{r,m} = \left(f_{r,m} - f_{r,worst} \right) \Big/ \left(f_{r,best} - f_{r,worst} \right) \tag{7}$$

$$M_{r,m} = q_{r,m} \Big/ \sum_{i=1}^{N} q_{r,m} \tag{8}$$

where, $f_{r,m}$ and $M_{r,m}$ respectively represent the functional fitness value and corresponding mass of the substance individual m in the RTH evolution of GSA algorithm. Assuming that the problem solved is a minimization problem, $f_{r,best}$ and $f_{r,worst}$ respectively represent the optimal and worst fitness values of substance individuals in the RTH evolution, and their mathematical expressions are as follows:

$$f_{r,best} = \min f_{r,m}, \ \ m \in \left\{ 1, 2, \cdots, N \right\} \tag{9}$$

$$f_{r,worst} = \max f_{r,m}, \ \ m \in \left\{ 1, 2, \cdots, N \right\} \tag{10}$$

2) Gravity calculation of individual matter

According to the law of universal gravitation, on dimension j, the universal gravitation of material individual m on material individual k is defined as follows:

$$F_{r,mk}^{(j)} = G_r * M_{r,pm} * M_{r,ak} \left(x_{r,m}^{(j)} - x_{r,k}^{(j)} \right) \Big/ \left(R_{r,mk} + \varepsilon \right) \tag{11}$$

where ε is an infinitesimal constant; $M_{r,pm}$ represents the inertial mass acting on the r evolution of the material individual m; $M_{r,ak}$ represents the inertial mass acting on the r evolution of the material individual m, and $M_{r,pm} = M_{r,ak} = M_{r,m}$; $R_{r,mk}$ is the Euclidean space distance between the material individual m and the material individual k; G_r represents the universal gravitation constant of the matter individual in the r evolution, and its expression is shown in Equation (12):

$$G_r = G\left(G_0, r \right) = G_0 \cdot \mathrm{e}^{-\theta \cdot r / w} \tag{12}$$

where, G_0 is the gravitational coefficient of individual matter at the beginning of evolution; θ is the algorithm control parameter, generally take $\theta = 20$; w is the maximum value of the algorithm evolution algebra. On dimension j, the resultant force of gravitational force on material individual m is $F_{r,m}^{(j)}$:

$$F_{r,m}^{(j)} = \sum_{m \neq r}^{N} d_j * F_{r,mk}^{(j)} \tag{13}$$

where, d_j is a random number evenly distributed within the interval (0, 1).

3) Calculation of individual acceleration

According to Newton's second law, the acceleration of material individual m in the r evolution on dimension j is defined as follows:

$$a_{r,m}^{(j)} = F_{r,m}^{(j)} \big/ M_{r,mm} \tag{14}$$

where $M_{r,mm} = M_{r,pm} = M_{r,ak} = M_{r,m}$.

4) Speed and location of material individual renewal

In each evolutionary process of the algorithm, material individuals according to Formula (14) and Formula (15) respectively update velocity $v_{r,m}^{(j)}$ and position $x_{r,m}^{(j)}$, and their specific mathematical expressions are as follows:

$$v_{r+1,m}^{(j)} = d_j * v_{r,m}^{(j)} + a_{r,m}^{(j)} \tag{15}$$

$$x_{r+1,m}^{(j)} = x_{r,m}^{(j)} + v_{r+1,m}^{(j)} \tag{16}$$

2.3. Improved Cuckoo Algorithm (GASCS)

Based on the limitations of CS algorithm, the GASCS algorithm with gravitational acceleration mechanism is proposed in this paper. The algorithm is based on the optimal solution found by the current search, Levy Flight random walk and preferences of individual random walk update location and host nests by combination of gravitational acceleration, the direction of the next evolution and best lair, and mutation probability method is adopted to guide the host nest move along the direction of the global optimal solution, and then to search the global optimal solution.

The execution steps of the GASCS algorithm are shown below.

Step 1. Initialization of the algorithm: the population is N host nests, the spatial dimension of the optimization problem is D, the maximum evolutionary algebra is W, the discovery probability is p_a, the universal gravitational coefficient at the initial time is G_0, the control parameter of the algorithm is θ, the initial moving speed of the host nest is $V_{r,m}$, where the evolutionary algebra $r = 1$.

Step 2. Calculate the randomized host nest location $X_{r,m} (m = 1, 2, \cdots, N)$ corresponds to the fitness function value $f(X_{r,m})$.

Step 3. Calculate G_r by formula (12), at the same time, calculate the current optimal value $f_{r,best}$ and the worst value $f_{r+1,worst}$ by formula (9) and formula (10), and the corresponding optimal solution $X_{r,gb}$.

Step 4. According to formula (7) and formula (8), calculate the mass of host nest $q_{r,m}$, $M_{r,m}$.

Step 5. Calculate the gravitational force $F_{r,m}^{(j)}$ and acceleration $a_{r,m}^{(j)}$ of the host nest of the current evolutionary algebra according to formula (13) and formula (14).

Step 6. The Levy flight random walk of formula (4) is used to generate a new host nest, and the candidate solution $x_{r+1,m}$ is discarded according to the discovery probability P_a

Step 7. Using the preferred random walk of formula (5) to generate a new host nest and replace the abandoned candidate solution in step 6.

Step 8. Generate the fitness value of the function $f\left(X'_{r+1,m}\right)$ corresponding to the candidate solution $X'_{r+1,m}$ by probability mutation method.

Step 9. Calculate the fitness value of the function $f\left(X'_{r+1,m}\right)$ corresponding to the candidate solution $f_{r+1,best}$ generated by the population in step 8, and update the current optimal value $f_{r+1,best}$ and the worst value $f_{r+1,worst}$ and the corresponding optimal solution $X_{r+1,gb}$.

Step 10. If the termination condition of the algorithm is satisfied, the optimal value and solution of the current evolution are output, and the algorithm is stopped; otherwise, go to step 3 and continue with the algorithm.

3. Optimized Scheduling Model

The optimal dispatching of power system is based on the balance between the power supply and the demand load, which not only meets the minimum power generation cost of the system, but also makes the emission of pollution gas meet the standard.

1) Economic scheduling objectives:

$$\min f_1 = \sum_{i=1}^{N}\left(a_i + b_i * P_i + c_i * P_i^2\right) \tag{17}$$

In the formula, N represents the number of thermal power units; a_i, b_i and c_i are the cost coefficients of generating units respectively; P_i is unit output; f_1 represents the objective function of economic scheduling.

2) Environmental treatment objectives:

$$\min f_2 = \sum_{i=1}^{N} E\left(P_i\right) \tag{18}$$

$$E\left(P_i\right) = 10^{-2}\left(\alpha_i + \beta_i P_i + \gamma_i P_i^2\right) + \xi_i \exp\left(P_i\right) \tag{19}$$

where α_i, β_i, γ_i and ξ are emission coefficients of pollutants respectively; f_2 represents the environmental scheduling objective function. The following multi-objective optimal scheduling model is obtained by combining the two objective functions and constraint conditions:

$$\min Y\left(x\right) = \min\left[f_1\left(P_i\right) f_2\left(P_i\right)\right] \tag{20}$$

$$h\left(P_i\right) = 0 \tag{21}$$

$$G\left(P_i\right) \geq 0 \tag{22}$$

In the formula, $H\left(P_i\right)$ and $G\left(P_i\right)$ are respectively the equality and inequality constraints of the scheduling model; $Y\left(x\right)$ is the integrated objective function.

4. The Analysis of Simulation

Select the IEEE6 cell for a 30-node system. Different algorithms were used to analyze and compare the models. The 24 h period was used for calculation. The parameters of unit operation, emission coefficient and line power loss were shown in Literature [10], and the upper and lower limits of output of unit G1 -

G6 and the system demand load were shown in Literature [11]. Keeping the parameter conditions fixed, MOPOS algorithm, NSGA algorithm, multi-objective cuckoo algorithm and its improved algorithm were respectively used to further analyze the above examples. Assuming that the population size was $n = 200$, the maximum number of iterations was $t = 600$, and the threshold value of the execution of communication was 0.6, the iteration process was shown in **Figure 2**. As can be seen from the iteration curves in **Figure 1**, compared with other algorithms, the iterations of the improved cuckoo algorithm are greatly reduced. The key is the introduction of AC operator, dynamic parameters and non-controlled sorting, which leads to the poor search efficiency of the algorithm

There is a change in convergence velocity. It takes 80 iterations for NSGA algorithm to reach the convergence state, and 72, 61 and 48 iterations for MOPOS algorithm, multi-objective cuckoo algorithm and its improved algorithm, respectively. The improved cuckoo algorithm improves the convergence speed and optimization ability.

After the completion of the iteration, the optimal Pareto boundary obtained by NSGA algorithm, MOPOS algorithm, multi-objective cuckoo algorithm and the improved multi-objective cuckoo algorithm is shown in **Figures 3-6**.

As can be seen from **Figure 2** to **Figure 5**, when the demand load of the regional power grid is determined, the cost of power generation is inversely proportional to pollutant emission, and the two balance each other. Through comparison, it can be seen that the first three algorithms did not search for the optimal
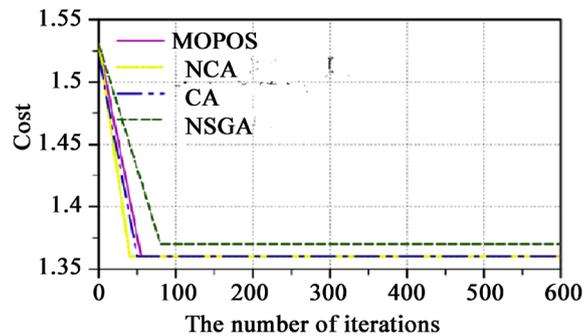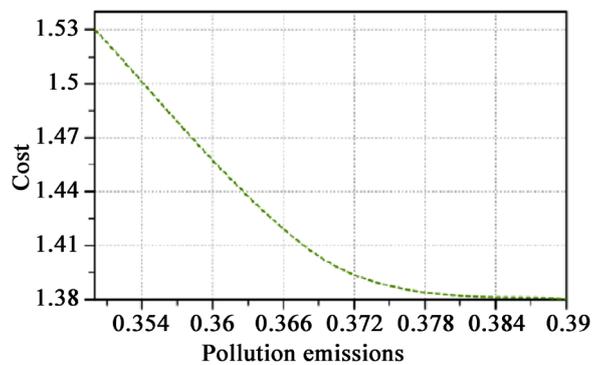

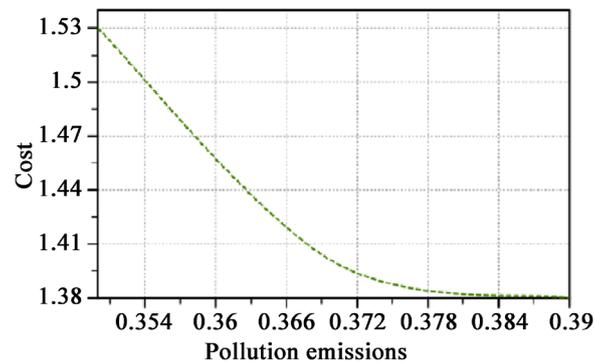**Figure 2.** Cost changes.

Figure 3. The optimal front of NSGA.

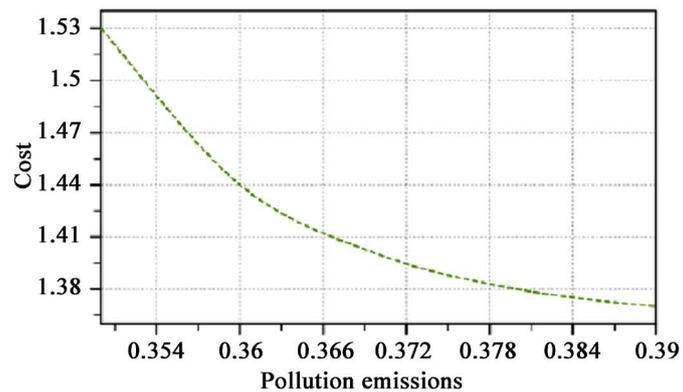**Figure 4.** The optimal front of CA.

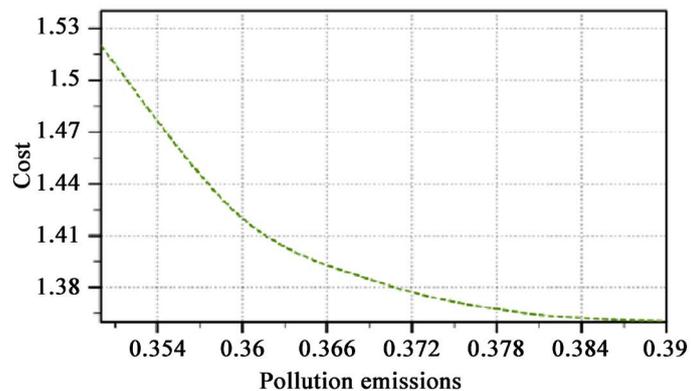**Figure 5.** The optimal front of MOPOS.

Figure 6. The optimal front of NCA.

Pareto peak, and the slope of the optimal Pareto peak obtained earlier in **Figure 2** & **Figure 3** was too large. However, the latter algorithm is too small, and the slope in the middle part of **Figure 4** fluctuates to a certain extent. The latter algorithm is intermittent, while the improved algorithm can obtain the smooth and integrated optimal Pareto front. It can be seen that the improved algorithm has a good effect on solving multi-objective optimization problems.

5. Conclusion

Cuckoo algorithm is a new intelligent optimization algorithm, which has the

advantages of strong generality, fewer control parameters, fast algorithm execution, etc., and has become a hot topic of research by scholars at present. However, this algorithm still has some limitations, such as the imbalance between global search and local search, which leads to the local extremum optimization, slow search speed and low convergence precision. In this paper, we treat the cuckoo nest as a mass entity, and use the law of universal gravitation and Newton's second law to analyze the Levy in the cuckoo algorithm. Flight random walk and preferences random walk optimization between individual gravitation is used to accelerate the search, effectively balance the cuckoo global search ability and local exploitation ability of the algorithm, avoid algorithm execution period into local extreme value point and the hysteresis phenomenon, improve the global search efficiency and convergence precision of the algorithm. The performance simulation of a 30-node 2IEEE6 unit system is carried out by using the optimization algorithm. The results show that the proposed GASCS algorithm has better global optimization performance, better robustness, faster search speed and higher convergence accuracy compared with other improved intelligent optimization algorithms.

Conflicts of Interest

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

References

[1] Yuan, Y., Li, L. and Zhao, H. (2019) Coordinated and Optimized Operation Method of AC/DC Hybrid Micro-Grid Source Network Based on Master-Slave Game Model. *Smart Electricity*, **47**, 30-37.

[2] Qi, J., Sun, S., Li, J., *et al.* (2020) Automatic Power Balance Control Strategy for Wind Storage AC Micro-Grid. *Smart Electric Power*, **48**, 9-14, 34.

[3] Hu, B., Ren, T., Che, X., *et al.* (2020) Research on Control Technology of Micro-Grid Grid-Connected Inverter Based on QPCI Control. *Smart Electric Power*, **48**, 23-27, 41.

[4] Wang, C. and Xiong, B. (2019) Two-Stage Optimization Method for Unbalanced Power Suppression of Micro-grid Based on Energy Storage Scheduling Mode. *Smart Electricity*, **47**, 22-28, 55.

[5] Chen, L. (2019) Study on Decentralized Scheduling Method of Load Balancing in Renewable Energy Power System. *Power Grid and Clean Energy*, **35**, 60-66.

[6] Brown, C.T., Liebovitch, L.S. and Glendon, R. (2007) Lévy flights in Dobe Ju' hoansi Foraging Patterns. *Human Ecology*, **35**, 129-138.
https://doi.org/10.1007/s10745-006-9083-4

[7] Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2012) GSA: A Gravitational Search Algorithm. *Intelligent Information Management*, **4**, 390-395.
https://doi.org/10.4236/iim.2012.46043

[8] Yazdani, S.,Nezamabadi-Pour, H., Kamyab, S. (2014) A Gravitational Search Algorithm for Multimodal Optimization. *Swarm & Evolutionary Computation*, **14**, 1-14.
https://doi.org/10.1016/j.swevo.2013.08.001

[9] Zhao, F.Q., Xue, F.L., Zhang, Y., Ma, W., Zhang, C. and Song, H.B. (2018) A Hybrid Algorithm Based on Self-Adaptive Gravitational Search Algorithm and Differential Evolution. *Expert Systems with Applications*, **113**, 515-530. https://doi.org/10.1016/j.eswa.2018.07.008

[10] Khan, K., Kamal, A., Basit, A., *et al.* (2019) Economic Load Dispatch of a Gridtied DC Microgrid Using the Interior Search Algorithm. *Energies*, **12**, 634. https://doi.org/10.3390/en12040634

[11] Cramer, J.A., Kramer, K.E., Johnson, K.J., *et al.* (2008) Automated Wavelength Selection for Spectroscopic Fuel Models by Symmetrically Contracting Repeated Unmoving Window Partial Least Squares. *Chemometrics and Intelligent Laboratory Systems*, **92**, 13-21. https://doi.org/10.1016/j.chemolab.2007.11.007