ISSN: 2664-9640

Research on grey wolf algorithm based on improved artificial bee colony optimization

Liang Ge [a], Fengyu Feng [b]

School of Computer Science and Technolog Hangzhou Dianzi University, Hangzhou, China

[a]geliang@hdu.edu.cn, [b]fengyufeng2014@gmail.com

Abstract

A hybrid method of artificial bee swarm and grey wolf algorithm (GWO-ABC) is proposed in this paper. The static greedy algorithm is used to replace the dynamic greedy algorithm in ABC employer-bee to strengthen the exploration performance. A new method of searching nectar source is proposed to improve the convergence speed. Then, the detection bees that affect the convergence speed are removed, and the reverse learning strategy is added in the stage of hiring bees to avoid falling into local optimal. Finally, in order to balance the exploration capability of hired bee, adaptive GWO is integrated in the observation bee to enhance the mining performance and optimization efficiency. The strict mathematical derivation proves that the GWO-ABC is stable and attainable. The reliability and effectiveness of the proposed method are verified by the comparative simulations.

Keywords

Grey wolf algorithm; Formatting; Artificial bee colony; Machine learning.

1. Introduction

Grey Wolf Optimizer (GWO) and Artificial Bee Colony (ABC) are two efficient swarm intelligent optimization algorithms [1-3]. Although the local search capability is strong, the global search capability of GWO is weak. The strong global search ability and slow convergence speed are the characteristics of ABC. Therefore, how to combine the advantages of the two algorithms to improve the optimization ability is very important [4-5].

Recent years, the research on artificial bee colony algorithm and grey wolf optimization algorithm has attracted the attention of many scholars. In [6], an improved artificial bee colony algorithm is proposed, by introducing the current optimal solution into the search equation of worker bee and onlooker, the convergence accuracy of the algorithm is improved effectively. In [7], the adaptive neighborhood selection and replacement process is used to refine the ABC to search for the best combination of Web services to meet user needs. In [8], the improved artificial bee colony algorithm, as an integrated learning strategy, improves the recognition rate of radar signals in the system. In [9], in order to improve the performance, a new artificial bee colony algorithm was proposed to improve the service quality. In [10], in order to improve the efficiency, the fuzzy reasoning system is used to adjust the normalized fitness of each wolf and GWO's hunting mechanism control parameters. In [11], an improved wolf optimizer based on tracking and optimization is proposed, which can solve function optimization and classical engineering constraints well. In [12], an improved hierarchical GWO algorithm was proposed to increase the searching range of wolves and maintain the diversity of wolves. In [13], a global maximum power point tracking control strategy based on an improved Grey Wolf optimizer (IGWO) algorithm has been applied to boost full-bridge isolation converter with high tracking accuracy and speed. In [14], an improved wolf optimizer (IGWO) is proposed to effectively solve

the complex constrained optimization problem, which is of great significance in smoothing the peak load of power system.

Although the artificial bee colony algorithm and grey wolf optimization algorithm have been studied, there is no solution to solve the optimization problem by combining the advantages of the two algorithms.

This paper is organized as follows. In section II, the mathematical models of artificial bee colony algorithm and grey wolf optimization algorithm are analyzed. In section III, a hybrid artificial bee colony algorithm and grey wolf algorithm is proposed. In section IV, comparative simulations of the proposed method are carried out. The section V is the summary.

2. Artificial Bee Colony Algorithm and Grey Wolf Algorithm

2.1. Maintaining the Integrity of the Specifications

Grey Wolf Optimizer (GWO) is proposed by Mirjalili in 2014, which is a novel swarm intelligence optimization algorithm [15-18]. The social hierarchy and hunting behavior of wolves are simulated. In nature, grey wolves belong to the top of the predator, which means they are at the top of the food chain [19-20]. Grey wolf populations mainly live in groups, with the average population size between 5 and 12. Their own social hierarchy is strictly Observation d, as shown in Figure 1. It can be seen from Figure 1 that the grey wolf population is mainly divided into four grades, including α wolf, β wolf, δ wolf and ω wolf. The first level is α, which is the top manager and ruler of the grey wolf population. The second is β, whose status is lower than α. If α dies, its position will be replaced by β. The third level is δ, which is subordinate to the management of α and β. Level four is ω, and ω must obey the other wolves.

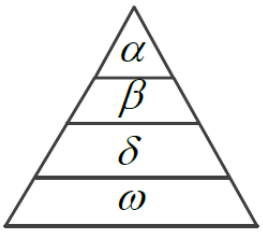


Fig. 1 Social hierarchy of wolves

In GWO, the hunting behavior of grey wolves follows three processes, including tracking and approaching the prey, tracking and surrounding the prey, and attacking the prey. The mathematical model of enclosing behavior can be expressed as follows.

$$\boldsymbol{D} = \left| \boldsymbol{C} \cdot \boldsymbol{X}_p(t) - \boldsymbol{X}(t) \right| \tag{1}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_p(t) - \boldsymbol{A} \cdot \boldsymbol{D} \tag{2}$$

Where, t is the current iteration number, and A, C are coefficient vectors. A=2ar1-a, C=2r2. Xp represents the position vector of the prey. X is the position vector of the grey y wolf, decreases from 2 to 0 during the whole iteration. r1 and r2 represent random vectors uniformly distributed in [0,1]. The mathematical model of attacking prey behavior is as follows.

$$\boldsymbol{D}_\alpha = \left| \boldsymbol{C}_1 \boldsymbol{X}_\alpha - \boldsymbol{X} \right| \tag{3}$$

$$\boldsymbol{D}_\beta = \left| \boldsymbol{C}_2 \boldsymbol{X}_\beta - \boldsymbol{X} \right| \tag{4}$$

$$\boldsymbol{D}_\delta = \left| \boldsymbol{C}_3 \boldsymbol{X}_\delta - \boldsymbol{X} \right| \tag{5}$$

$$X_2 = X_\beta(t) - A_2 D_\beta \tag{6}$$

$$X_3 = X_\delta(t) - A_3 D_\delta \tag{7}$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{8}$$

In the above equations, (3)-(5) are the distance between ω wolf and α wolf, β wolf and δ wolf, respectively. X_α, X_β and X_δ represent position vectors of α, β, δ and ω respectively. (6)-(9) are the position update mode of ω. A and C play A key role in the hunting process of grey wolves and determine the exploration and mining stage of GWO. When $|A|>1$, GWO tends to emphasize exploration ability; When $|A|<1$, GWO tends to emphasize mining capacity; The accessibility of the grey wolf population to prey can be increased and reduced by C.

Firstly, the positions of wolves in the search space is randomly initialized. Then the positions are updated according to equations (3)~(9).Then the fitness values are calculated. Three optimal solutions (static update α, β and δ wolf positions) are updated according to their fitness values. Figure. 2 shows the flow chart of the GWO.

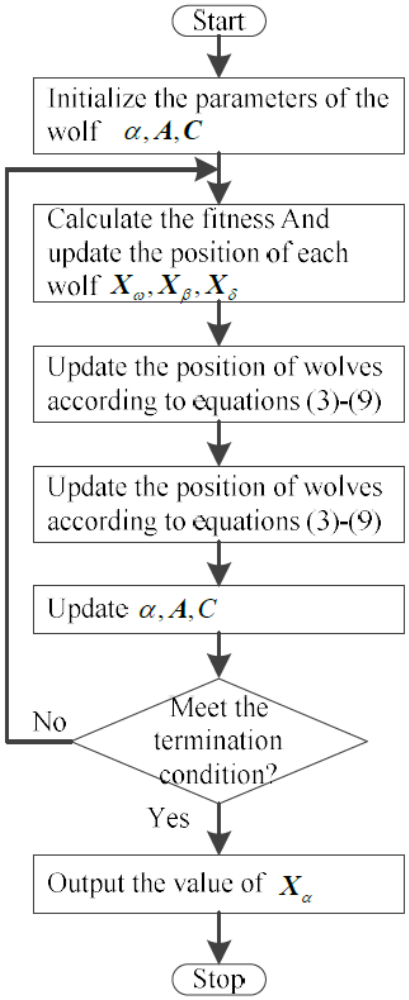


Fig. 2 Flow chart of grey wolf optimization algorithm

2.2. Artificial bee colony algorithm (ABC)

ABC comes from the foraging behavior of bees. In ABC, bees are divided into hiring bees, observing bees and detecting bees, and the search goes through the corresponding three stages [21-22]. Hiring bees and observing bees each accounted for half of the nectar sources. The location of a nectar source represents a candidate solution to the optimization problem, and

the amount of nectar in the nectar source represents the fitness value of the corresponding candidate solution.

1) Stage of hiring bees

The nectar source is sought by the hired bees, and the information is saved. According to Equation (1), a new honey source is searched near the honey source.

$$v_{id} = x_{id} + \varphi_{id}(x_{id} - x_{kd}) \tag{9}$$

Where, x_{id} is the d-dimensional value of the ith honey source, k represents a honey source randomly selected among N honey sources, and $k \neq i$ is true; d represents a random selected dimension, and φ_{id} is a random number uniformly distributed between [-1,1].

The nectar amount (fitness) of the nectar source is calculated according to (2) (if the optimization problem solved is a minimum).

$$fit_i = \begin{cases} 1/(1+f_i), f_i > 0 \\ 1+|f_i|, otherwise \end{cases} \tag{10}$$

Where, fit_i is the fitness value of the ith honey source, and f_i is the objective function value of the ith solution.

The honey source is updated by the greedy algorithm as shown in (3).

$$x_i = \begin{cases} v_i, fit(x_i) < fit(v_i) \\ x_i, otherwise \end{cases} \tag{11}$$

(3) indicates that if the fitness value of the new honey source v_i is better than that of the original honey source x_i , then v_i will be replaced, otherwise x_i will be retained.

2) The observation bees

After the hired bees completed the nectar source search, the bees are observation d to select the nectar source with probability Pi according to the nectar source information.

The calculation of the probability Pi is shown as (4). According to the nectar source information provided by the hired bees, the roulette method is adopted to select the nectar source with the probability Pi.

$$P_i = fit_i / \sum_{i=1}^{N} fit_i \tag{12}$$

Among them, the greater the fitness value, the greater the Pi, so the better nectar source is more likely to be selected. Formula (1) is used for further searching near the selected honey source.

3) Detection of bees

After the limit (threshold) mining, if the nectar amount of a nectar source remains unchanged, indicating that the algorithm falls into the local optimum, the nectar source will be abandoned and the corresponding hired bees will be converted into detection bees. The role of the scout bee is to prevent the algorithm from falling into the local optimal position, and the new nectar source location is generated as follows.

$$x_{id} = a_d + rand * (b_d - a_d) \tag{13}$$

Where rand is the random distribution number between [0, 1], and a_d and b_d represent the lower and upper boundaries of the d-dimensional search space, respectively.

3. Hybrid Grey Wolf Algorithm and Artificial Bee Colony Algorithm

3.1. Fusion of the two search methods

(1) Hired bee search method

(1) is used for hiring bees to search which has strong exploration ability. In order to balance this exploration performance, dynamic greedy algorithm is selected to accelerate the convergence speed, but this dynamic greedy algorithm will reduce the stability of the algorithm. In order to improve the stability, the original ABC dynamic greedy algorithm is abandoned and the static greedy algorithm is adopted. In order to improve the convergence speed, a new method of hiring bees searching nectar source is proposed, shown as follows.

$$v_{id} = x_{id} + \varphi_{id}(x_{id} - x_{kd}) + r(x_{gd} - x_{id}) \tag{14}$$

Where, r is randomly selected in [0, 1], which represents the d-dimensional value of the global optimal honey source.

(2)The Observation bees

The high-quality honey source is selected by the Observation d bees by the roulette method. The new honey source is selected by formula (1) near the honey source, which reflected a certain local searching ability. However, the local search capability is limited due to the use of one - dimensional operation. In order to improve the local search ability, a new search method is proposed, shown as follows.

$$v_{id} = x_{gd} + \varphi_{id}(x_{gd} - x_{id}) \tag{15}$$

The convergence speed is accelerated and the local search capability is strengthened by (15).

3.2. Reverse learning hiring bees

Reverse learning strategies are added to the hiring bees. The new honey source is generated by the reverse learning method, which can be expressed as follows.

$$x_{id} = a_d + (b_d - x_{id}) \tag{16}$$

The reverse learning pseudocode of the hired bee can be expressed as algorithm 1.

Algorithm 1 reverse learning hiring bees

A honey source num from 1 to N is randomly selected

For i =1 to N

If i≠num

New honey sources are produced according to (1) and (6)

Else

Perform reverse learning to generate new nectar sources

End If

End For

Fitness value of honey sources are calculated

(3) is used to update the nectar source and update the trial

3.3. The observation bees of GWO

In order to further improve the convergence speed, optimization efficiency and solution accuracy of ABC, GWO search mode is incorporated into the observation bee stage.

If the fitness value of honey source unchanged after limit iteration, GWO search is used to improve the mining ability of the algorithm. Observation bees optimized by GWO, such as algorithm 2.

Algorithm 2: Observation bees optimized by GWO

ISSN: 2664-9640

For i=1 to N do
Select a honey source i
If trials (i) < limit
According to Equations (1) and (7), new honey sources are generated
Else
According to Equations (3) ~ (9), new honey sources are generated
End If
Calculate the fitness value of honey source
Update the honey source and update trial (i) according to the greedy algorithm in (3).
End For

3.4. Selection method of honey source

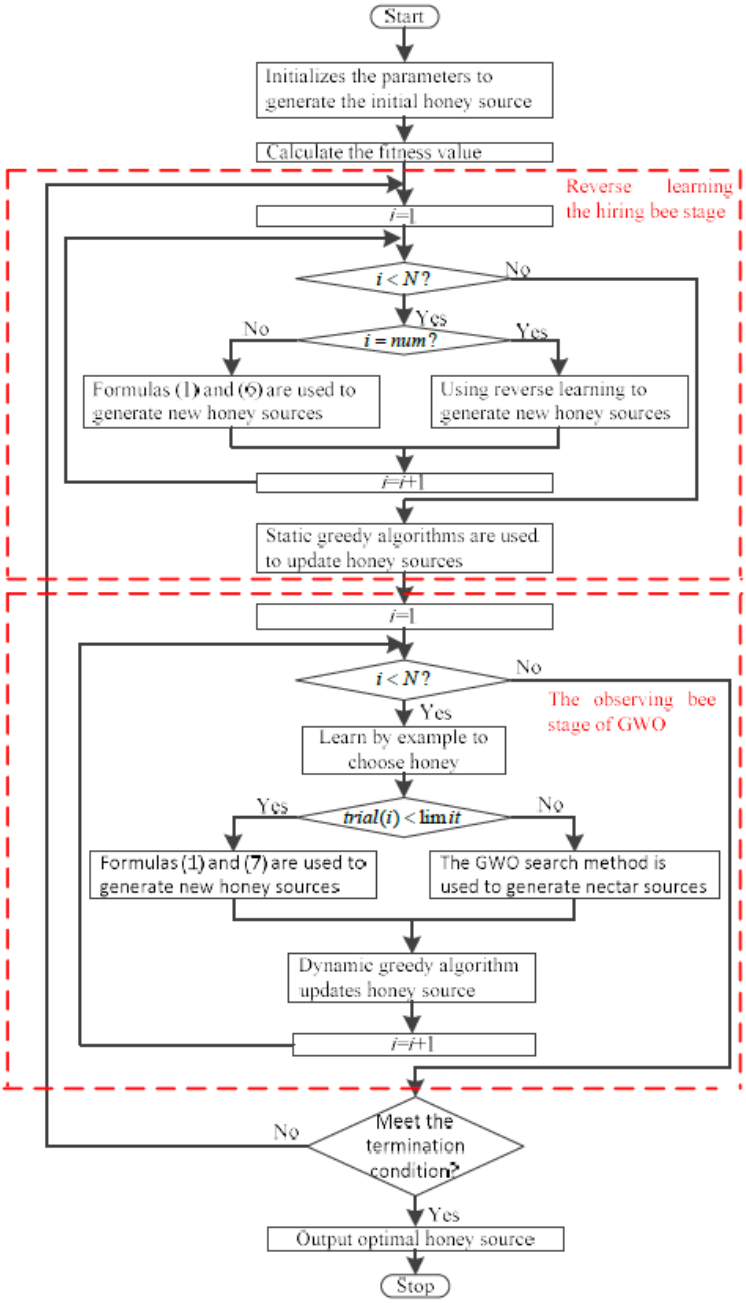


Fig. 3 The flow chart of GWO-ABC

In the observing bee stage, model learning alternative roulette is used to select nectar sources. The process of selecting a honey source is described as follows: first, the fitness of the honey source is ranked from the largest to the smallest. The front source is used as a model for the back source

Then honey source I was randomly given a model according to (9).

$$num = ceil(i * rand) \tag{17}$$

Where, ceil () is an integral function.

The flow chart of the integrated grey wolf algorithm and artificial bee colony algorithm (GWO-ABC) is shown as Figure 3.

3.5. Stability analysis

The stability analysis of GWO-ABC mainly includes first-order stability analysis and second-order stability analysis. To prove the search converges to a fixed point, that is, the optimal honey source location, and its variance converges to zero. First order stability is used to ensure convergence of mean value and second order stability is used to ensure convergence of variance. The stability of the new search method (15) for observing bees is analyzed. Equation (15) can be rewritten as follows.

$$v_{id} = (1+\varphi_{id})x_{gd} - \varphi_{id}x_{id} \tag{18}$$

It can be simplified as follows.

$$y_{t+1} = (1+\varphi)A - \varphi y_t \tag{19}$$

Where, $y_{t+1} = v_{id}$, $\varphi = \varphi_{id}$, $y_t = x_{id}$, $A = x_{gd}$.

Theorem 1 The expected sequence of y_t can be converged and converged to A.

Proof: Take the expected value of this equation twice.

$$E(y_{t+1}) = (1+\mu_\varphi)A - \mu_\varphi E(y_t) \tag{20}$$

Where, μ_φ is the expected value of φ, and φ is a random number uniformly distributed between [-1,1]. Therefore, the following equation can be obtained.

$$E(y_{t+1}) = A \tag{21}$$

Equation (21) indicates that the expected value of the search method is constant, indicating that it always converges to A. The proof is complete.

$$y_{t+1}^2 = [(1+\varphi)A - \varphi y_t]^2 = (1+\varphi^2+2\varphi)A^2 + \varphi^2 y_t^2 - 2A(\varphi+\varphi^2)y_t \tag{22}$$

The expected value of equation (22) can be expressed as follows.

$$\begin{aligned} E(y_{t+1}^2) &= E[(1+\varphi^2+2\varphi)A^2 + \varphi^2 y_t^2 - 2A(\varphi+\varphi^2)y_t] \\ &= [1+E(\varphi^2)+2\mu_\varphi]A^2 + E(\varphi^2)E(y_t^2) - 2A[\mu_\varphi + E(\varphi^2)]E(y_t) \end{aligned} \tag{23}$$

Since φ is A random number uniformly distributed between [-1, 1], the following can be obtained.

$$E(\varphi^2) = 1/3 \tag{24}$$

$$E(y_{t+1}^2) = \frac{1}{3}E(y_t^2) - \frac{2}{3}AE(y_t) + \frac{4}{3}A^2 \tag{25}$$

According to (20), the following can be obtained.

$$E^2(y_{t+1}) = A^2 \tag{26}$$

Theorem 2: The variance of $D(y_t)$ can be converged and converged to 0.

Proof: According to equations (25) and (26), the following can be obtained.

$$D(y_{t+1}) = E(y_{t+1}^2) - E^2(y_{t+1}) = \frac{1}{3}E(y_t^2) - \frac{2}{3}AE(y_t) + \frac{4}{3}A^2 - A^2 = \frac{1}{3}E[(y_t - A)^2] \quad (27)$$

The following condition is true.

$$y_{t+1} - A = \varphi A - \varphi y_t \quad (28)$$

The following can be obtained.

$$E((y_t - A)^2) = \frac{1}{3}E((y_{t-1} - A)^2) \quad (29)$$

The recursive relationship can be expressed as follows.

$$D(y_t) = \frac{1}{3^t}E((y_0 - A)^2) \quad (30)$$

We can show that $D(y_t)$ converges and converges to 0, which means that when $t \to \infty$, $D(y_0) = 0$.

$$\lim_{t\to\infty} D(y_t) = \lim_{t\to\infty}\{\frac{1}{3^t}E[(y_0 - A)^2]\} = E[(y_0 - A)^2 \lim_{t\to\infty}\frac{1}{3^t} = 0 \quad (31)$$

End proof.

In summary, the observation of the first and second order stability of the bee stage is satisfied, indicating that the mean value and variance are convergent.

4. Simulation

The experimental environment adopted Windows 7 operating system, 3.10GHz CPU and 8GB PC memory, and the programming language is MATLAB R2018a.

4.1. Test function

To verify the effectiveness of GWO-ABC, 10 different types of Benchmark functions are used for optimization tests. The function names of 10 Benchmark are listed in Table 1.

Table 1. Benchmark Test Functions

Order	Function
f1	Sphere
f2	Schwefel2.22
f3	Schwefel1.2
f4	Schwefel2.21
f5	Rosenbrock
f6	Step
f7	Quatric
f8	Schwefel2.26
f9	Rastrigin
f10	Ackley

In this paper, five representative comparison algorithms are selected for comparison experiment with GWO-ABC, which are ABC [23], GWO [24], GA-ABC [25] and GA-ACO [26]. To be fair, the 5 algorithms adopt the same setting, that is, the population number is 40, and the

maximum number of iterations is set to 2000. In addition, and the limit value of GWO-ABC is set to 18. For the parameters of other algorithms, please refer to the corresponding literature. Table 2 shows the experimental results of 6 algorithms on 10 Benchmark functions.

TABLE 2. The Comparison Results of Optimized Performance

Name	GWO- ABC	ABC	GWO	GA- ABC	GA-ACO
Sphere	Mean	0	1.69e-27	5.4e-131	5.1e-25
	Std	0	4.0e-27	2.2e-130	3.4e-25
Schwefel2.22	Mean	0	4.2e-15	1.1e-75	1.1e-13
	Std	0	3.5e-15	2.4e-75	13.8e-14
Schwefel1.2	Mean	3.7e-6	1.1e+4	6.7e-40	1.2e+4
	Std	2.0e-5	1.6e+3	2.3e-39	1.6e+3
Schwefel2.21	Mean	0	1.3e+1	2.8e-33	1.0e+1
	Std	0	1.4e+0	4.1e-33	1.5e+0
Rosenbrock	Mean	1.1e+0	1.8e+0	2.7e+1	9.8e-1
	Std	4.5e+0	1.5e+0	7.4e-1	1.1e+0
Step	Mean	0	0	0	0
	Std	0	0	0	0
Quatric	Mean	7.9e-4	8.2e-2	8.9e-4	5.1e-2
	Std	1.0e-3	1.5e-2	5.4e-4	9.0e-3
Schwefel2.26	Mean	8.8e-12	1.8e-11	6.2e+3	9.0e-12
	Std	8.4e-13	2.5e-11	7.0e+2	3.3e-13
Rastrigin	Mean	0	3.2e-7	2.3e-1	0
	Std	0	6.9e-7	1.2e+0	0
Ackley	Mean	1.9e-14	4.0e-13	7.0e-15	1.2e-12
	Std	1.5e-14	1.6e-13	2.2e-15	5.3e-13

As can be seen from Table 2, GWO-ABC ranked first for 12 times, while other algorithms ranked first for only a few times. On the whole, it can be seen that GWO-ABC has the optimal optimization performance among the 5 comparison algorithms. The optimization performance of GWO-ABC is better than that of GWO and ABC. This is because GWO-ABC is the result of the integrated improvement of GWO and ABC, realizing the complementary advantages of GWO and ABC. In summary, compared with the 5 algorithms, GWO-ABC shows excellent optimization performance. The std value of GWO-ABC on most functions is the smallest among the 5 algorithms, indicating that GWO-ABC has extremely strong stability.

(2)Comparison of running time

Fig. 4 shows the average time of each algorithm running independently for 30 times. It can be seen from Figure 4 that the average time of GWO-ABC is 1.018s, which is the smallest among the five algorithms. The reason why GWO-ABC has the least running time is as follows: The reverse learning bees uses the parallel objective function evaluation method, which improves the running speed and saves the running time. In addition, the GWO search method of parallel

computing and the model learning strategy also accelerate the running speed of the algorithm, thus saving the running time.

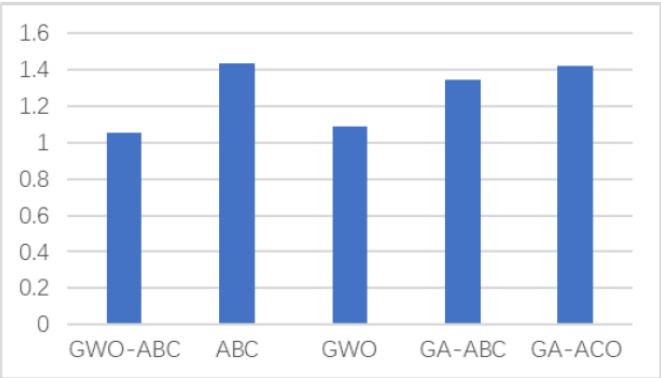


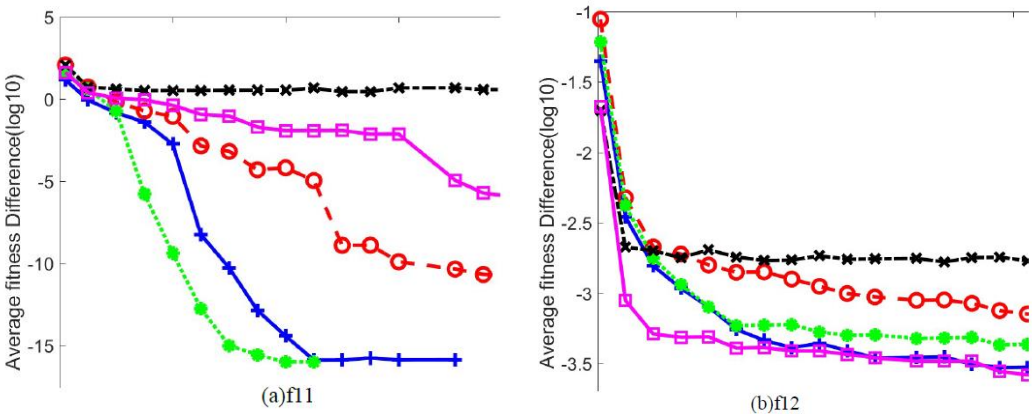
Fig. 4 Comparison of average running time

4.2. Convergence analysis

To test the convergence performance of GWO-ABC, the convergence analysis is carried out. In this paper, functions F11-F16 are used for simulation comparison experiment, as shown in Table 3. The convergence comparison of GWO-ABC and the other four algorithms is shown in Fig. 5.

Table 3. Test Function

Order	Function
f11	Shift Schwefel2.21
f12	Shift Rosenbrock
f13	Shift Rastrigin
f14	Shift Girewank
f15	Shift Ackley
f16	Rotated Sphere



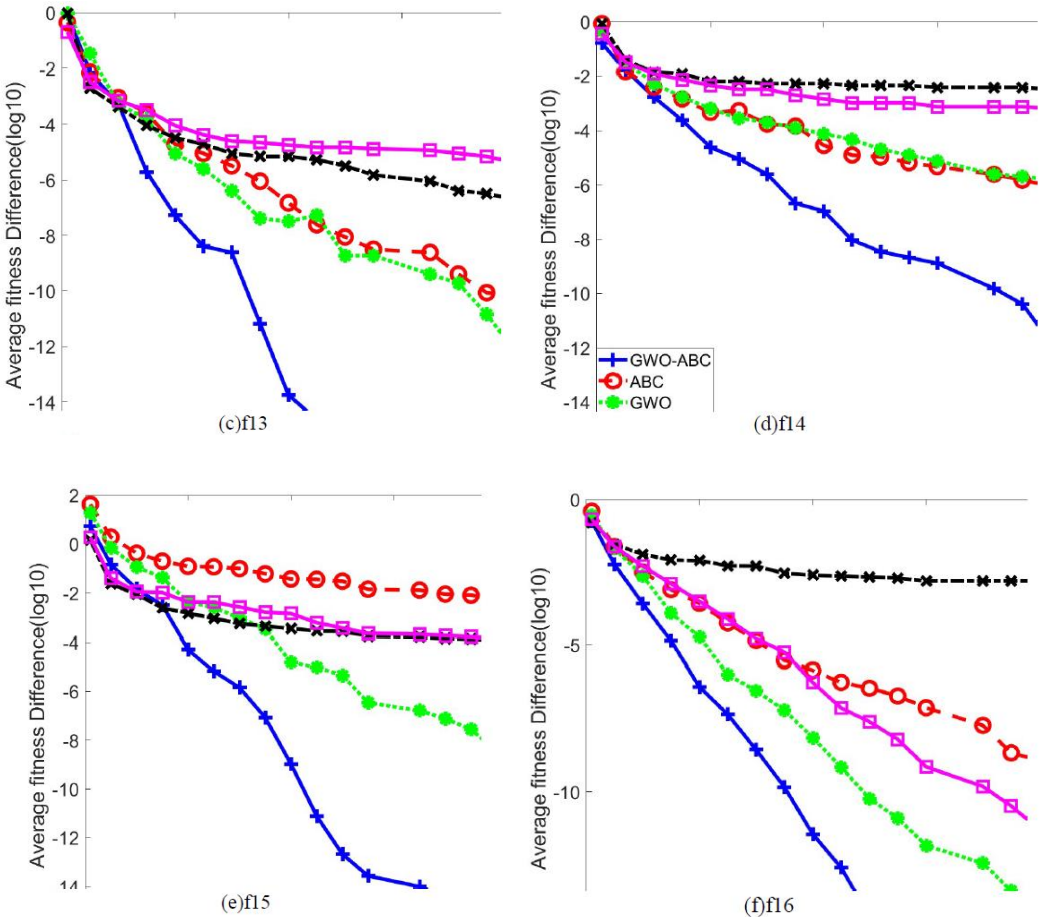


Fig. 5 The convergence curves of the 5 algorithms

It can be seen from the comparison figure that COMPARED with other algorithms, GWO-ABC has a faster convergence speed and the convergence performance is far better than the other four algorithms.

5. Concludes

A hybrid optimization algorithm of GWO and ABC (GWO-ABC) is proposed in this paper. Firstly, ABC is improved by adopting static greedy algorithm when hiring bees, and a new way of searching nectar source is designed. The detection bees that affected the convergence speed are removed, the reverse learning strategy is added in the stage of hiring bees, and the exploration ability is further improved. The GWO method is then incorporated into the observation bee to improve local search capability and optimization efficiency. The optimization results of the function show that GWO-ABC has better optimization effect and runs faster than the four typical algorithms.

References

[1] Xun Li, Kwai Man Luk. The grey wolf optimizer and its applications in electromagnetics, IEEE Transactions on Antennas and Propagation, 2020,vol.68,no.3,pp.2186 – 2197.

[2] Karar Mahmoud, Mahmoud M. Hussein, Mohamed Abdel-Nasser, et al. Optimal voltage control in distribution systems with intermittent PV using multiobjective grey-wolf-lévy optimizer, IEEE Systems Journal, 2020, vol. 14, no.1, pp.760 – 770.

[3] Radu-Emil Precup, Radu-Codrut David, Emil M. Petriu. Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity, IEEE Transactions on Industrial Electronics, 2017, vol.64, no.1, pp.527 – 534.

[4] Shenghan Zhou, Xingxing Xu, Zhenzhong Xu, et al. Fractional-order modeling and fuzzy clustering of improved artificial bee colony algorithms, IEEE Transactions on Industrial Informatics, 2019,vol.15, no.11, pp.5988 – 5998.
[5] Hao Gao, Yujiao Shi, Chi-Man Pun, et al. An improved artificial bee colony algorithm with its application, IEEE Transactions on Industrial Informatics, 2019, vol. 15,no.4, pp.1853 – 1865.
[6] Song Zhang, Sanyang Liu. A discrete improved artificial bee colony algorithm for 0–1 knapsack problem, IEEE Access, 2019, vol.7, pp.104982 – 104991.
[7] Fadl Dahan, Hassan Mathkour, Mohammed Arafah, et al. Two-step artificial bee colony algorithm enhancement for QoS-Aware web service selection problem, IEEE Access, 2018, vol. 7, pp.21787 – 21794.
[8] Jingpeng Gao, Yi Lu; Junwei Qi, Liangxi Shen, et al. A Radar Signal Recognition System Based on non-negative matrix factorization network and improved artificial bee colony algorithm, IEEE Access, 2019, vol.7, pp.117612 – 117626.
[9] Xianzhi Wang, Xiaofei Xu, Quan Z. Sheng, et al. Novel artificial bee colony algorithms for QoS-Aware service selection, IEEE Transactions on Services Computing, 2019, vol.12, no.2, pp.247-261.
[10] Allan Christian Krainski Ferrari, Gideon Villar Leandro, Leandro dos Santos Coelho, et al. Tuning of control parameters of grey wolf optimizer using fuzzy inference, IEEE Latin America Transactions , 2019, vol.17, no.07, pp.1191 – 1198.
[11] M. W. Guo, J. S. Wang, L. F. Zhu, et al. An improved grey wolf optimizer based on tracking and seeking modes to solve function optimization problems, IEEE Access, 2020, vol.8, pp.69861 – 69893.
[12] Qiang Tu, Xuechen Chen, Xingcheng Liu. Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection, IEEE Access, vol.7, pp.78012 – 78028.
[13] Ke Guo, Lichuang Cui, Mingxuan Mao, et al. An improved gray wolf optimizer MPPT algorithm for PV system with BFBIC converter under partial shading, IEEE Access, 2020, vol.8, pp.103476 – 103490.
[14] Zhong-Kai Feng, Shuai Liu, Wen-Jing Niu, et al. Optimal operation of hydropower system by improved grey wolf optimizer based on elite mutation and quasi-oppositional learning, IEEE Access, vol.7, pp.155513 – 155529.
[15] Ibrahim B. M. Taha, Ehab E. Elattar. Optimal reactive power resources sizing for power system operations enhancement based on improved grey wolf optimizer, IET Generation, Transmission & Distribution, 2018, vol.12, no.14, pp.3421 – 3434.
[16] Yuncheng Zhang, Jianzhong Zhou, Yang Zheng, et al. Control optimisation for pumped storage unit in micro-grid with wind power penetration using improved grey wolf optimizer, IET Generation, Transmission & Distribution, 2017, vol.11, no.13,pp.3246 – 3256.
[17] Jianzhong Zhou, Wenlong Zhu, Yang Zheng, et al. Precise equivalent model of small hydro generator cluster and its parameter identification using improved Grey Wolf optimizer, IET Generation, Transmission & Distribution, 2016, vol.10, no.9, pp.2108 – 2117.
[18] C. Komathi, M. G. Umamaheswari. Erratum to "Design of gray wolf optimizer algorithm-based fractional order PI controller for power factor correction in SMPS applications, IEEE Transactions on Power Electronics, 2020, vol.35, no.5, pp.5543 – 5543.
[19] Jamila Snoussi, Seifeddine Ben Elghali, Mohamed Benbouzid, et al. Optimal sizing of energy storage systems using frequency-separation-based energy management for fuel cell hybrid electric vehicles, IEEE Transactions on Vehicular Technology, 2018,vol.67, no.10, pp.9337 – 9346.
[20] Komathi C., Umamaheswari M. G. Design of gray wolf optimizer algorithm-based fractional order PI controller for power factor correction in SMPS applications, IEEE Transactions on Power Electronics, 2020, vol.35, no.2, pp.2100 – 2118.
[21] Liming Zhang, Saisai Wang, Kai Zhang, et al. Cooperative artificial bee colony algorithm with multiple populations for interval multiobjective optimization problems, IEEE Transactions on Fuzzy Systems, 2019, vol.27, no.5, pp.1052 – 1065.
[22] Junzo Watada, Arunava Roy, Bo Wang, et al. An artificial bee colony-based double layered neural network approach for solving quadratic bi-level programming problems, IEEE Access, vol.8, pp.21549 – 21564.
[23] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, vol.69, no.3, pp. 46-61.

[24] Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems [J]. Engineering with Computers, 2013, vol.29, no.2,pp.245-245.
[25] Shyam Sundar, Alok Singh. Metaheuristic approaches for the blockmodel problem, IEEE Systems Journal, 2015,vol.9, no.4,pp.1237– 1247.
[26] Girma S. Tewolde, Weihua Sheng. Robot Path Integration inmanufacturing processes: genetic algorithm versus ant colony optimization, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2008, vol.38, no.2, pp. 278 – 287.