FIR digital filter design based on improved artificial bee colony algorithm

Lian Lian1 · Zhongda Tian2

Accepted: 25 June 2022 / Published online: 1 October 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
The traditional swarm intelligence optimization algorithm is prone to fall into local optimal solutions in finite impulse response (FIR) digital filter design and has slow convergence speed. In order to optimize the design of FIR filter, a FIR digital filter design method based on improved bee colony (ABC) algorithm is proposed. This improved ABC algorithm can adaptively adjust the step size of the selected neighborhood of nectar source location. At the same time, the information of the global optimal solution is used to guide the search of candidate solution, which improves the global search ability of the algorithm. The improved ABC algorithm can balance the conflict between local search ability and global search ability, so it can achieve better optimization effect. The time and space complexity of the algorithm is analyzed in detail. Then, the improved ABC algorithm is used to design three typical FIR digital filters, namely low-pass, band-pass and band-stop filter. The performance of the designed filter is tested by simulation experiments. The experimental results show that compared with other state-of-the-art optimization algorithms, the proposed FIR filter design method has achieved better effect and performance. Meanwhile, the proposed design method has shorter optimization time. The superiority of the proposed method is verified.

Keywords Filter design · FIR filter · Improved artificial bee colony algorithm · Optimization

1 Introduction
Digital filter is an important part in digital signal processing system. Compared with the analog filter, the digital filter can be designed more widely. Because of the programmability, superior cost performance, small size requirement and easier implementation of the software, the digital filter has replaced the analog filter in many applications (Kaplun et al., 2020). According to the length of the impulse response, the digital filter can be divided into infinite impulse response (IIR) filter and finite impulse response (FIR) filter (Lee et al. 2020). IIR digital filter can be designed with analog filter or computer-aided methods, and their implementation process is also complex compared to FIR filter implementation (Agrawal et al. 2019a, 2017). However, the phase frequency characteristic of IIR filter is nonlinear, which limits its application range (Dash et al. 2020). The phase of the FIR filter is strictly linear because of only zeros in transfer function and the structure is nonrecursive; therefore, FIR filters are always stable. Owing to this fact FIR filters are exploited in numerous applications in signal processing (Padmapriya et al. 2019; Umadevi et al. 2019), image compression (Li et al. 2019; Maeda et al. 2018), communication (Hameed et al. 2018), measurement (Park et al. 2019; Ryu et al. 2020), etc. However, a common limitation of FIR filter is that they require higher filter order to achieve the specifications, thus requiring more memory and processing time as compared to IIR filters. A proper design procedure is essential for the effective and reliable operation of FIR filter for a particular application. Thus, designing of digital FIR filters with high performance is a crucial task.

The traditional design methods of FIR digital filter are window function method (Hossin et al. 2018; Naderian et al. 2017). It can approximate the frequency...
characteristics of ideal filters approximately. The window function method is simple, but it is difficult to determine the boundary frequency of the pass-band and stop-band accurately and can only converge to the local optimum. Parks-McClellan algorithm is a representative method of the optimal design of traditional digital filters (Filip 2016). This method can achieve better pass-band and stop-band performance, but the algorithm is complex and the computation time is very long. Artificial neural networks have strong adaptive and self-learning ability. Many scholars combine the neural network and FIR filter, and obtain satisfactory design results. These neural networks for FIR filter design include Hopfield neural network (Jou et al. 2011; Xu et al. 2018), backpropagation neural network (Alwahab et al. 2018; Chauhan and Sathish, 2018), convolutional neural network (Kim and Kim, 2017), etc. However, it is difficult to determine the number of hidden layer nodes. If the hidden layer nodes are too few, the neural network cannot obtain all the characteristics of the signal, which leads to the approximation failure. Too many nodes in the hidden layer can cause excessive training, and the network may remember the redundant features generated by the interference in the signal (Mittal 2020). In addition, some scholars used the L1 method (Aggarwal et al. 2016) and the L1 method combined with some optimization algorithms (Aggarwal et al. 2018, 2015) to design the filter and have achieved good performance.

Recently, some scholars have reviewed the use of evolutionary algorithm to design filters and pointed out that evolutionary algorithm is very suitable for filter design (Aggarwal et al. 2021; Dwivedi et al. 2018). These algorithms include simulated annealing algorithm (Ma et al. 2018; Wu et al. 2015), genetic algorithm (Szapos et al. 2016; Miyata et al. 2018), particle swarm optimization (PSO) algorithm (Zhang et al. 2018; Shao et al. 2017; Kumar 2019), differential evolution (DE) algorithm (Chandra et al. 2016; Dash et al. 2017), ant colony optimization algorithm (Tsutsui and Suyama, 2014), cuckoo search algorithm (CSA) (Sarangi et al. 2018; Kumar et al. 2018; Kumar et al. 2020), etc. These optimization algorithms have been used for digital filter design and have made some progress. However, these algorithms also have various limitations and disadvantages. SSA algorithm has the disadvantages of slow convergence, long execution time, sensitive parameters, which make it inefficient and even infeasible algorithm. GA and ACO algorithms are difficult to apply into practical applications because of their complex structure and slow operation speed. PSO algorithm is easy to fall into local optimum because each particle in the swarm only searches in a limited sample space. Similar to the commonly used evolutionary algorithm, DE algorithm achieves the optimal solution crossover and selection of the difference vector between the individuals. Therefore, for some complex optimization problems, DE algorithm also has a local optimum and premature convergence. There are some problems in CSA, such as low convergence accuracy and low convergence speed. Furthermore, the search probability and search step have a great influence on the performance of the CSA. In a word, these mature intelligent optimization algorithms have more parameters to be adjusted, and improper parameter adjustment is easy to make the algorithm fall into local extremum, thus reducing the performance of the designed FIR filter. On the other hand, some scholars combine multiple optimization algorithms to design filters, such as quantum PSO and ABC (Agrawal et al. 2019b), TVC-PSO and artificial bee colony (ABC) (Agrawal et al. 2018), hybrid PSO (Agrawal et al. 2020), ABC and Nelder–Mead simplex search (Dwivedi et al. 2017), modified multi-objective artificial bee colony algorithm (Dwivedi et al. 2016). The combination of optimization algorithms may increase the complexity of the algorithm. Some other optimization algorithms such as gravitational search algorithm (Saha et al. 2013a), bacteria foraging optimization algorithm (Saha et al. 2013b), cat swarm optimization algorithm (Saha et al. 2013c), seeker optimization algorithm (Saha et al. 2012), colliding bodies optimization algorithm (Mahata et al. 2016) and so forth also get good filter design results. Therefore, it is worth studying to find an intelligent computational algorithm with simple structure, strong robustness, few parameters and easy to adjust for the design of FIR filter.

ABC algorithm is an intelligent optimization algorithm based on bee behavior (Karaboga and Basturk, 2007). For a continuous optimization problem like CEC2005, CEC2014 and CEC2015 benchmark functions, ABC algorithm is preferred over other optimization algorithms, such as GA, adaptive GA, PSO and DE for its remarkable performance. Compared with other optimization algorithms, ABC algorithm has the characteristics of strong global convergence, few parameters and wide application range, which is especially suitable for the design of FIR filter (Sharma et al. 2016). In the study, ABC algorithm has been found to outperform the state-of-the-art evolutionary algorithm in meeting the specified FIR design (Kockanat et al. 2018). However, the strong randomness of the standard ABC algorithm in local search may lead to premature convergence of the algorithm, thus reducing the ability of the algorithm to converge to the global optimal solution. In this paper, the neighborhood search method of the standard ABC algorithm is improved, which can enhance the local chemotaxis search ability of the algorithm, so improve the optimization ability of the ABC algorithm. The improved ABC algorithm is more suitable for the design of FIR digital filters. The test results of 12 Benchmark functions show that the proposed improved ABC algorithm has better
optimization results and optimization performance than standard ABC algorithm. Based on the improved ABC algorithm, the design of FIR filter is transformed into the optimization of filter parameters, and then the improved ABC algorithm is used to search the parameter space of FIR filter efficiently. It can guarantee the optimization of the parameters and make the FIR filter have better filtering performance. The effectiveness and superiority of the proposed FIR design method are verified by the simulation design of the filter.

In this paper, we focus on integrated, analytical and comparative study of improved ABC, ABC, GA, PSO, ACO, DE and CSA for the design of digital FIR low-pass, band-pass and band-pass filter. For achieving high accuracy, the optimal filter should have a low-pass-band ripple and optimum stop-band attenuation. Based on the above discussion, the main contributions of this paper are summarized as follows:

1. An improved ABC algorithm is proposed. This improved ABC algorithm can adaptively adjust the step length function of the selected neighborhood nectar source location, use the information of the global optimal solution to guide the search of the candidate solution and improve the development efficiency.

2. The superior performance of the proposed improved ABC algorithm is studied in detail. The analysis of time complexity and space complexity of several optimization algorithms are discussed.

3. Compared with the state-of-the-art algorithms, the capabilities of proposed algorithm dealing with practical problems in FIR filter design are verified. Three typical FIR filters are selected for simulation experiments. The results show that the designed FIR filter by the improved ABC algorithm has better amplitude frequency response characteristics and less optimization time.

The structure of this paper is as follows. Section 2 introduces the proposed improved ABC algorithm. Section 3 gives the design process of FIR digital filter based on improved ABC algorithm. Section 4 introduces the simulation results and verifies the effectiveness of the proposed FIR digital filter design method. The conclusions and future works are presented in Sect. 5.

2 Improved ABC algorithm

2.1 Standard ABC algorithm

The nectar source of ABC algorithm is seen as a point in solution space. The quality of nectar source is \( SN \). Suppose the dimension of the problem to be solved is \( D \), the location of nectar source is represented by \( X_i = [x_{i1}, x_{i2}, \ldots, x_{id}], \) \( i \in \{1, 2, \ldots, SN\}, \) \( d \in \{1, 2, \ldots, D\} \) and \( x_{id} \in (L_d, U_d) \), where \( i \) represents the position of the i-th honey source, \( d \) represents the d-th solution in D-dimensional solution space, \( t \in \{1, 2, \ldots, M\} \) is the number of iterations, \( M \) represents the maximum number of iterations. \( L_d \) represents the lower bound of the search space, and \( U_d \) represents the upper bound of the search space (Dwived et al. 2018). Equation (1) gives the location of nectar source randomly generated in the search space.

\[
x_{id} = L_d + r(U_d - L_d)
\]

where \( r \) is random distribution in the range of \((0, 1)\).

At the beginning of the search phase, according to Eq. (2), a new nectar source is generated by the employed bees around the search source (Karaboga and Akay, 2009).

\[
v_{id} = x_{id} + q_{ij}(x_{id} - x_{jd})
\]

where \( v_{id} \) is a new randomly generated nectar source, \( j \) represents the position of the j-th honey source, \( j \neq i \). \( q_{ij} \) is a random number, and \( q_{ij} \in [-1, 1] \). When the fitness of new nectar source \( V_i \) is better than current nectar source \( X_i \), greedy algorithm is adopted to replace \( X_i \) with \( V_i \). Otherwise \( X_i \) is retained. Onlooker bees then share the information based on their nectar sources. Its follow probability is as follows.

\[
p_i = \frac{\text{fit}_i}{\sum_{i=1}^{SN} \text{fit}_i}
\]

Namely, \( \text{fit}_i \) is the i-th fitness value. Onlooker bee generates a random number belonging to \([0, 1]\) and compares it with \( p_i \). If the random number is less than \( p_i \), a new nectar source will be generated according to Eq. (2).

In the optimization process of the algorithm, in order to avoid the algorithm falling into the local optimum, it is necessary to make the algorithm have the ability to jump out of the local optimum. Namely, when the fitness value of the algorithm does not change, the algorithm should expand the search range and carry out further search beyond the local optimal value. In ABC algorithm, a threshold of the number of iterations (limit) is set. In the process of searching, if the nectar source \( X_i \) reaches the threshold \( \text{limit} \) after trial iterative times and fails to find a better nectar source, the nectar source \( X_i \) will be abandoned and the corresponding employed bees will be changed to scout bees. The scout bees will randomly generate a new nectar source instead of \( X_i \) in the search space. The new nectar source is generated in following Eq. (4), and \( t \) is the current number of iterations.

\[
x_{id} = L_d + \frac{\text{fit}_i}{\sum_{i=1}^{SN} \text{fit}_i} (U_d - L_d)
\]
\[ x_i^{t+1} = \begin{cases} L_d + \text{rand}(0, 1)(U_d - L_d), & t \geq \text{limit} \\ x_i^t, & t < \text{limit} \end{cases} \tag{4} \]

In above Eq. (4), the value of \( \text{limit} \) needs to be set in combination with the actual problem. If \( \text{limit} \) is set too large, it will lead to large trial iterations of ABC algorithm to jump out of the local optimum. If the value of \( \text{limit} \) is too small, resulting in very small trial iterations of ABC algorithm, a new nectar source will be generated, and the exploration ability of local search will be lost.

The implementation steps of ABC algorithm are as follows.

**Step 1** The generation of training data sample set. Initialization of parameters, including the number of nectar sources—\( S N \), the maximum number of iterations—\( M \) and the maximum number of nectar sources mining—\( \text{limit} \). The optimization parameters are given. Let \( t = 1 \).

**Step 2** A employed bee is assigned to collect nectar. According to Eq. (2), the search process is started to generate new nectar source \( V_i \).

**Step 3** The fitness value is calculated from sample data. Update the nectar source according to the greedy algorithm.

**Step 4** According to Eq. (3), the follow probability of onlooker is updated. Onlooker bees search and save nectar sources based on greedy algorithm.

**Step 5** The algorithm determines whether the nectar source should be discarded. If true, onlooker bees become scout bees. Otherwise, go to Step 7.

**Step 6** According to Eq. (4), scout bees will generate new nectar sources.

**Step 7** Let \( t = t + 1 \). If the termination conditions have been met, the optimal parameters can be output. Otherwise it goes to Step 2 and continues.

### 2.2 Improvement of ABC algorithm

In the standard ABC algorithm, the location of the nectar source is locally updated and searched in a random way. The randomness of the standard ABC algorithm in local search is strong, which may lead to premature convergence of the algorithm, thus reducing the ability of the algorithm to converge to the global optimal solution (Ahirwal et al. 2014). Therefore, in the neighborhood search strategy of ABC algorithm, the adaptive step size parameter is added, and the global optimal solution guidance term is added. The improved ABC algorithm can enhance the local search ability of the algorithm and make the algorithm realize the balance between the global search and the local search, so it can achieve better optimization effect. The improved ABC algorithm will have better optimization performance for the optimization problems with more local optimal values.

In Eq. (2), \( \varphi_{id} \) is a random number, and \( x_{id} \) is a random choice among neighbor individuals. Hence, the new global random search capability obtained from Eq. (2) is very strong. However, the solution may be a better one or a worse one, so the local search ability of neighborhood search in Eq. (2) is poor. In order to improve the local search ability of ABC algorithm, the random step \( \varphi_{id} \) in Eq. (2) needs to be improved so that it can adjust adaptively with the change of fitness. The new step adjustment strategy is as follows.

\[
v_{id} = x_{id} + R_{id}(x_{id} - x_{jd}) + c_{id}(x_{bestd} - x_{id}) \tag{5}\]

\[
R_{id} = \begin{cases} r_{id}(1 - f_i/f_{best}), & f_i \neq f_{best} \\ \varphi_{id}, & f_i = f_{best} \end{cases} \tag{6}\]

\[
c_{id} = c_{min} + (c_{max} - c_{min}) \frac{2}{1 + \exp(-\alpha(\frac{f_{id} - f_{best}}{\beta}))} - 1 \tag{7}\]

where the random value of \( r_{id} \) is +1 or -1, \( \varphi_{id} \) is a random number that varies from \([-1, 1]\), and \( f_i \) represents the fitness function in the optimization problem. \( t \) represents the current number of iterations. \( M \) represents the maximum number of iterations. \( c_{max}, c_{min}, \alpha \) and \( \beta \) are constant values. \( x_{bestd} \) is the current optimal solution of \( D \)-dimension. \( f_{best} \) is the optimal value of fitness function. Compared to the random step size \( \varphi_{id} \), the value range of \( R_{id} \) is broader. The absolute value of \( R_{id} \) could be greater than 1. Therefore, in the early iteration process, a large step size is helpful to expand the search space of the algorithm. When \( f_i \) is close to \( f_{best} \), \( R_{id} \) approaches 0. In this case, a smaller step size is helpful for the algorithm to find the optimal solution in the local search. \( R_{id} \) plays a guiding role in the trend of finding nectar sources. In the early stage of iteration, the parameter \( c_{id} \) should be smaller to reduce the global optimization and improve the global search ability. In the later stage of the iteration, \( c_{id} \) should keep a large number, so that the algorithm can quickly converge to the global optimum.

In the new neighborhood search strategy, the global optimal solution guidance term is added to guide the artificial bee to move to the current optimal solution purposefully when searching the food source location, which overcomes the disadvantage of too strong randomness in the search process of the standard ABC algorithm. In the proposed improved ABC algorithm, the values of parameters \( \alpha \) and \( \beta \) have an important influence on the guidance parameter of the global optimal solution. The values of \( \alpha \) and \( \beta \) should be moderate to reduce the steady-state error of the algorithm and improve the convergence speed. Another important parameter is \( \text{limit} \), which value directly
affects the global search ability of swarm. In the optimization process, by comparing the honey content of the new solution and the original solution, the greedy selection mechanism is used to select the solution with larger honey content for local search, so as to guide the evolution of individuals in the population toward the optimal solution. The onlooker bees select the nectar source according to the roulette method. It is observed that the probability of onlooker bees searching for a certain nectar source is directly proportional to the honey content of the nectar source, so that the nectar source with high honey content can be exploited better, and the algorithm converges to the optimal solution with the increase of iteration times. When a solution is abandoned, the scout bees are generated to find a new nectar source. This operation makes the algorithm accept the degradation of the solution to a certain extent. On the other hand, it keeps the search range large enough in a period of time to avoid premature convergence.

In order to verify the performance improvement of the improved ABC algorithm, this paper selects 12 benchmark functions to verify the effective of the improved ABC algorithm. The details of these Benchmark functions, such as dimension, variable range, are given in Table 1. In Table 1, $m$ is the dimension of the variable of the function.

The specific parameters of standard ABC and improved ABC are as follows. The number of nectar source (SN) is 50, the maximum number of iterations ($M$) is 5000, the parameter limit is set as 1000, $c_{max}$ is 1, $c_{min}$ is 0, $z$ is 50, $\beta$ is 5. For fairness, both algorithms run 20 times. The average value obtained by running 30 times is taken as the optimization result. The performance and stability of the algorithm are evaluated by the mean and standard deviation of the experimental results. In general, we believe that the smaller the mean of the algorithm is excellent. When the mean value is the same, the smaller the standard deviation, the better the algorithm more stable. Table 2 shows the optimization results of the standard ABC algorithm and the improved ABC algorithm. It can be seen from Table 2 that the mean value of the improved ABC algorithm is better than those of the standard ABC algorithm. At the same time, the standard deviation of the improved ABC algorithm is smaller, which indicates the strong robustness of the proposed algorithm. In sum, the designed improved ABC algorithm has obtained better optimization performance.

### 2.3 Complexity analysis

In this section, the complexity analysis of the proposed improved ABC algorithm is given. We discuss the complexity of the algorithm from two aspects of space and time. In this study, $P$ is the number of population, $D$ is the dimension of the problem to be solved, and $M$ is the maximum number of iterations.

Firstly, the space complexity of the optimization algorithm is analyzed. According to the flow chart of these optimization algorithms, the number of variables needed is analyzed. Finally, we can get how much storage space these algorithms need.

1. Improved ABC. The number of population is $P$, the number of individuals is $D$, and the required space of 3 kind bees is $1.5PD$. The space of the position of nectar source is $PD$. The space needed for the feasible solution is $D + 2P$. The total space required for improved ABC algorithm is $2.5PD + 2D + 4P + 23$.

2. ABC. The number of population is $P$, the number of individuals is $D$, and the required space of 3 kind bees is $1.5PD$. The space of the position of nectar source is $PD$. The space needed for the feasible solution is $D + 2P$. The total space needed for the current solution is $D + 2P$. The total space needed for the upper and lower boundary, the selected probability and other parameters is 23. Therefore, the total space required for ABC algorithm is $2.5PD + 2D + 4P + 19$.

3. GA. The number of population is $P$, the number of individuals is $D$, the space of all individuals is $PD$. The space of fitness value is $PD$. The space required for the selection operation is $0.5P + 4D$. The space required for the mutation operation is $0.5P + 4D$. The space required for variables such as crossover probability, selection probability and maximum number of iterations is 17. Therefore, the total space required for GA algorithm is $2PD + P + 8D + 17$.

4. PSO. The number of population is $P$, the number of individuals is $D$, the space of position and velocity individual is $2PD$. The space required for the local optimal value is $PD$. The space required to update the position is $2P + 0.5D$. The space required to update the velocity is $P + 0.5D$. The required space for parameters such as inertia weight, acceleration factor and maximum number of iterations, and etc., is 34. Therefore, the total space required for PSO algorithm is $3PD + 3P + D + 34$.

5. ACO. The number of population is $P$, the number of ant is $D$, and the space of all ants is $PD$. The space needed to calculate the transfer probability of ants is $PD$. The space needed to calculate the tabu table is $PD + P$. The space needed for updating residual information is $2P + D$. Other parameters include the maximum number of iterations, heuristics, expectation heuristics, information intensity, etc. The required...
| Function ID | Function | Dimension | Range       | Optimal value |
|-------------|----------|-----------|-------------|---------------|
| F1          | $f_1 = \sum_{d=1}^{m} x_d^2$ | 100       | $[-100, 100]$ | 0             |
| F2          | $f_2 = \sum_{d=1}^{m} |x_d| + \prod_{d=1}^{m} |x_d|$ | 100       | $[-10, 10]$   | 0             |
| F3          | $f_3 = \sum_{d=1}^{m} \left( \sum_{j=1}^{d} x_j \right)^2$ | 100       | $[-100, 100]$ | 0             |
| F4          | $f_4 = \max \{ |x_d|, 1 \leq d \leq m \}$ | 100       | $[-100, 100]$ | 0             |
| F5          | $f_5(x) = \sum_{d=1}^{m} 100(x_{d+1}^2 - x_d)^2 + (1 - x_d)^2$ | 100       | $[-30, 30]$   | 0             |
| F6          | $f_6(x) = \sum_{d=1}^{m} (|x_d + 0.5 |)^2$ | 100       | $[-100, 100]$ | 0             |
| F7          | $f_7(x) = \sum_{d=1}^{m} (|x_d + 0.5 |)^2$ | 100       | $[-1.28, 1.28]$ | 0             |
| F8          | $f_8(x) = \sum_{d=1}^{m} -x_d \sin \left( \sqrt{|x_d|} \right)$ | 100       | $[-500, 500]$ | $-18.9829 \times m$ |
| F9          | $f_9(x) = \sum_{d=1}^{m} |x_d^2 - 10 \cos(2\pi x_d) + 10|$ | 100       | $[-5.12, 5.12]$ | 0             |
| F10         | $f_{10}(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{m} \sum_{d=1}^{m} x_d^2 - \exp \left( \frac{\sum_{d=1}^{m} \cos(2\pi x_d)}{\sqrt{m}} \right)} \right)$ | 100       | $[-32, 32]$   | 0             |
| F11         | $f_{11}(x) = \frac{1}{\prod_{d=1}^{m} \cos \left( \frac{x_d}{\sqrt{m}} \right) + 1}$ | 100       | $[-600, 600]$ | 0             |
| F12         | $f_{12}(x) = 0.1 \left\{ \sin^2(3\pi x_d) + \sum_{d=1}^{m} (x_d - 1)^2[1 + \sin^2(3\pi x_d + 1)] \right\} + (x_m - 1)^2[1 + \sin^2(2\pi x_m)] \right\}$ | 100       | $[-50, 50]$   | 0             |
6. DE. The number of population is $P$, the space complexity of the algorithms

Table 2 The comparison results of standard ABC algorithm and improved ABC algorithm

| Function | Algorithm | Improved ABC | ABC |
|----------|-----------|--------------|-----|
| F1       | Mean      | 4.2513E-189  | 4.4407E-158 |
|          | Std       | 0.0000E + 00 | 2.4323E-157 |
| F2       | Mean      | 5.2857E-250  | 2.4456E-86  |
|          | Std       | 0.0000E + 00 | 1.1395E-85  |
| F3       | Mean      | 4.6149E-101  | 5.2630E-136 |
|          | Std       | 2.5068E-100  | 2.5009E-135 |
| F4       | Mean      | 2.6347E-240  | 1.6505E-86  |
|          | Std       | 0.0000E + 00 | 9.0403E-86  |
| F5       | Mean      | 1.0263E-07   | 1.0693E-05  |
|          | Std       | 3.6832E-07   | 2.2612E-05  |
| F6       | Mean      | 1.2510E-26   | 5.5364E-13  |
|          | Std       | 2.2562E-26   | 1.5388E-12  |
| F7       | Mean      | 6.0602E-05   | 1.3205E-04  |
|          | Std       | 5.7900E-05   | 1.0883E-04  |
| F8       | Mean      | -8.3267E + 03| -8.3026E + 03|
|          | Std       | 8.3508E-07   | 1.2293E-05  |
| F9       | Mean      | 0.0000E + 00 | 6.3893E-07  |
|          | Std       | 0.0000E + 00 | 2.2612E-05  |
| F10      | Mean      | 1.0066E-15   | 8.8818E-16  |
|          | Std       | 6.4836E-16   | 0.0000E + 00|
| F11      | Mean      | 1.5246E-08   | 6.4319E-05  |
|          | Std       | 8.3508E-07   | 1.2293E-05  |
| F12      | Mean      | 5.3341E-24   | 1.5583E-13  |
|          | Std       | 1.1517E-23   | 3.0981E-13  |

storage space is 19. Therefore, the total space required for ACO algorithm is $3PD + P + 2D + 19$.

6. DE. The number of population is $P$, the number of individuals is $D$, and the space of all individuals is $PD$. The space required for compilation and cross operations is $PD$. The space needed for greedy selection operations is $PD + P$. The space needed to select the next generation of individual operations is $D$. The storage space required for parameters such as mutation operator, crossover operator and evolution algebra is $D$. Therefore, the total space required for DE algorithm is $3PD + 2P + D + 21$.

7. CSA. The number of population is $P$, the number of individuals is $D$, and the space of all individuals is $PD$. The space required for the nest position is $D$. The storage space required for the number of available nests, the number of discarded and rebuilt nests is $1.5PD$. The space needed for Levi’s position update is $P + 3D$. The storage space needed for other parameters, such as discovery probability and number of cycles, etc., is 23. Therefore, the total space required for CSA algorithm is $2.5PD + P + 4D + 23$.

From the above analysis results, the spatial complexity of these algorithms is shown in Table 3.

The results in Table 3 show that the space complexity of these optimization algorithms is a function of the number of populations and the dimensions of the problems to be solved. As a whole, the space requirements of these optimization algorithms are not much different. With the development of computer hardware, the space requirement of optimization algorithm has little influence on the realization of the algorithm. Therefore, it can be considered that these optimization algorithms are equal in space complexity, and the time complexity should be used to evaluate the algorithm.

The time complexity of these optimization algorithms is analyzed. In the calculation process, the addition and the subtraction are equal to 1 time unit, and the multiplication and the division are equal to 4 time units.

1. Improved ABC. (a) Initialization phase: The number of population is $P$, each employed bee has $D$ characteristics, and the time required for initialization is $PD$. The information of employed bees and onlooker bees is stored, and the time required is $PD$. The total time is $2PD$; (b) the employed bees phase: The time for the employed bees to find a new nectar source near the existing nectar source is $3PD$. $PD$ is needed to select the nectar source again. It needs to repeat $M$ times in total, so it needs $4MPD$; (c) according to Eq. (5), (6) and (7), it takes $4.5P^2$ for the onlooker bees to select a new nectar source and $3.5PD$ to generate a new nectar source. It takes $1.5D$ to select a nectar source. It needs to repeat $M$ times in total, so the time required is $M(4PD + 4.5P^2 + 1.5D)$; (d) The time needed for onlooker bees to randomly select nectar source is $D$. Up to $0.5P$ times. $M$ times in total, so the time required is $0.5MPD$. Therefore, the time complexity of improved ABC algorithm is $2PD + M(8.5PD + 4.5P^2 + 1.5D)$.

2. ABC. (a) Initialization phase: The number of population is $P$, each employed bee has $D$ characteristics, and
the time required for initialization is PD. The information of employed bees and onlooker bees is stored, and the time required is PD. The total time is 2PD; (b) the employed bees phase: The time for the employed bees to find a new nectar source near the existing nectar source is 3PD. PD is needed to select the nectar source again. It needs to repeat M times in total, so it needs 4MPD; (c) it takes 0.5P (P + 3D) for the onlooker bees to select a nectar source and 3.5PD to generate a new nectar source. It takes 1.5D to select a nectar source. It needs to repeat M times in total, so the time required is M(4PD + 0.5P^2 + 1.5D); (d) the time needed for onlooker bees to randomly select nectar source is D. Up to 0.5P times. M times in total, so the time required is 0.5MPD. Therefore, the time complexity of ABC algorithm is 2PD + M(8.5PD + 0.5P^2 + 1.5D).

3. GA. (a) Initialization phase: The population number is P, each individual has D characteristics, so the initialization time is PD; (b) cross phase: The time required to select the individuals to be cross calculated is P. The time needed to select the cross location of an individual is D. The time needed to calculate the crossover operator is 6. It needs to execute M times in total. So the total time required is M(6 + D + P); (c) mutation phase: The time required for each chromosome mutation is D. The time required for mutation with a certain probability is PD. Total M times in mutation phase. So the time required is M(PD + D + P); Therefore, the time complexity of GA is PD + M(PD + 2D + 2P + 6).

4. PSO. (a) Initialization phase: The population number is P, each particle has D characteristics, and the velocity and position of initialization particles are two characteristics, so the time required is 2PD; (b) Update phase: The update time of each particle speed is 12D. The update time of position is D. The update phase is executed M times, so the time required for this phase is 12DM; (c) the search stage of local and global optimal value: For local optimal value, it needs to traverse D times, and for global optimal value, it needs to traverse P times. M times are carried out in this phase. The time required is M(P + D); (d) update phase of speed and position: The time required for calculation is 12D. Execution times are M times. The time required is 12MD. Therefore, the time complexity of PSO is 2PD + M(25D + P).

5. ACO. Initialization phase: The population number is P, each individual has D characteristics, so the initialization time is PD; (b) construction of solution space stage: The time required for a single individual to construct solution space is 4D + 5. In this stage, M times are repeated. Therefore, the time required for this stage is MP(4D + 5); (c) the pheromone updating stage, the time to calculate the path length of each ant is 8D. The time to find the current optimal solution is 2P. The time to update pheromone is 10PD. M times are executed. The total time in this stage is M(10PD + 2P + 8D). Therefore, the time complexity of ACO is 2PD + M(14PD + 12D + 5P).

6. DE. (a) Initialization phase: The population number is P, each individual has D characteristics, so the initialization time is PD; (b) mutation phase: It takes 5D to randomly select 3 individuals from the population. Total M times in mutation phase. So the time required is 15MD; (c) cross phase: The time required to select the individuals to be cross calculated is P. The time needed to select the cross location of an individual is D. It needs to execute M times in total. So the total time required is MPD; (d) boundary condition treatment requires time is 4D + 2P, and it needs to execute M times in total. So the total time required is M(4D + 2P). Therefore, the time complexity of DE is PD + M(16PD + 4D + 2P).

7. CSA. Initialization phase: The population number is P, each individual has D characteristics, so the initialization time is PD; (b) find the best nest. The time required is MPD. (c) The cuckoo in the population is updated. The time required is MPD. (d) Find the best nest. Compare with the previous generation nest to find the best nest. The time required is 2D + 8. It needs to run m times, and the required time is M(2D + 8). (e) The time to dynamically discover the probability and change the nest position is MPD. (f) Find the current best nest. The time required is MD. Therefore, the time complexity of CSA is PD + M(4PD + 4D + 8).

In conclusion, the running time of these optimization algorithms is also shown in Table 4.

According to the time complexity comparison results of these algorithms in Table 4, the time complexity of each algorithm is a function of P, D and M. Considering that

| Algorithm   | The time complexity                                      |
|-------------|---------------------------------------------------------|
| Improved ABC| 2PD + M(8.5PD + 4.5P^2 + 1.5D)                         |
| ABC         | 2PD + M(8.5PD + 0.5P^2 + 1.5D)                         |
| GA          | PD + M(PD + 2D + 2P + 6)                               |
| PSO         | 2PD + M(25D + P)                                       |
| ACO         | 2PD + M(14PD + 12D + 5P)                               |
| DE          | PD + M(16PD + 4D + 2P)                                 |
| CSA         | PD + M(4PD + 4D + 8)                                   |

Table 4 The time complexity of the algorithms
$M$ will be much larger than $P$ and $D$, the $M$ term in the time complexity expression of each algorithm is ignored. Although the value range of $P$ and $D$ is very different, it is generally considered that the time complexity of algorithm ACO is the highest, followed by improved ABC, ABC, CSA, DE, and the lowest are GA and PSO. When $P$ is smaller (less than 20), the time complexity of PSO algorithm is higher than that of GA algorithm, while when $P$ is larger, the time complexity of PSO algorithm is smaller than that of GA algorithm. The above time complexity analysis results are based on theory. In the optimization of practical problems, the calculation time is also affected by the convergence speed, which can make up for the high time complexity of the algorithm. For GA, DE and CSA, with the increase of the dimension of the problem to be solved, the control parameters of the algorithm are more and stricter, so the algorithm is not flexible in application. A good parameter setting will converge very fast. If the parameter setting is not correct, it may not converge all the time. Because PSO algorithm relies on local optimal and all optimal solutions, but lacks stochastic process, the convergence of PSO is not very good. In the case of a large number of local optimal solutions, it is very easy to fall into local optimal. Because of the time complexity of ACO algorithm, it needs more running time. The algorithm has a certain ability to deal with different problems, and the initial convergence speed is very fast, but when it falls into the local optimum, it cannot jump out of the local optimum quickly and approach the global optimum. The improved ABC algorithm and the standard ABC algorithm, although the time complexity is not low, have the convergence speed only slower than GA, can quickly converge to the global optimal solution. Therefore, the comparison of optimization results of practical problems can truly reflect the time complexity of optimization algorithm. In this paper, the average computing time of these optimization algorithms is given in the simulation. The simulation results show that the improved ABC algorithm improves the performance without increasing the time complexity of the algorithm.

3 The design of FIR digital filter based on improved ABC algorithm

The unit sampling response of the $N$-order FIR digital filter is $h(0)$, $h(1)$, \ldots, $h(N-1)$. The transfer function can be expressed as (Song et al. 2020; Bouhamla et al. 2020)

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$ \hspace{1cm} (8)

Let $z = e^{j\omega}$, then the frequency response of the filter is (Ahirwal et al. 2013)

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n} = |H(e^{j\omega})|e^{j\phi(\omega)}$$ \hspace{1cm} (9)

where $|H(e^{j\omega})|$ and $\phi(\omega)$ are the magnitude response and the phase response, respectively.

If the ideal frequency response for the FIR digital filter is $|H_d(e^{j\omega})|$, then on the discrete points $\{\omega_i | i = 1, 2, \ldots, N\}$, the sum of the square error between the amplitude of the designed filter $|H(e^{j\omega})|$ and that of the ideal filter $|H_d(e^{j\omega})|$ is

$$E = \sum |H(e^{j\omega}) - H_d(e^{j\omega})|^2$$ \hspace{1cm} (10)

Equation (9) is substituted into Eq. (10), and there is

$$\min E = \sum_{i=1}^{M} \left( |\sum_{n=0}^{N-1} h(n)e^{-j\omega_in}| - |H_d(e^{j\omega_i})| \right)^2$$ \hspace{1cm} (11)

s.t. $h(n) \in [-1, 1]$

The frequency response of the ideal filter is generally in the form of rectangular waveform (such as low-pass filter and high-pass filter). The ideal sampling response obtained by inverse Fourier transform is in the form of sampling signal, and the maximum amplitude of the sampling signal is 1, which requires that the modulus of the designed filter coefficient $h(n)$ should be less than 1. Therefore, the parameter to be optimized, that is, the coefficient of FIR filter, is defined as $[-1, 1]$. It is clear that $E$ is a nonlinear function of the filter coefficient $h(n)$. Therefore, $E$ is a function with $N$ unknown values. According to the minimum mean square error criterion in frequency domain, the design of FIR filter is to select the filter coefficient $h(0)$, $h(1)$, \ldots, $h(N-1)$ to minimize the objective function $E$. Obviously, this is a combinatorial optimization problem. Therefore, we can use the improved ABC algorithm to solve the above combinatorial optimization problem.

Optimization design of FIR filter is to search the optimal filter coefficients, and this process can be viewed as using improved ABC algorithm to find the most abundant nectar source. In this paper, the objective function is the minimizing operation for Eq. (11). The objective function is smaller, the frequency response of designed filter ($|H(e^{j\omega})|$) is more close to the ideal frequency response of the filter ($|H_d(e^{j\omega})|$).

Obviously, the smaller the value of fitness function is, the smaller the mean square error of the filtering coefficient corresponding to the nectar source is. It means that the nectar source corresponds to a better filtering coefficient. In the design of FIR filter based on improved ABC algorithm, Eq. (11) is chosen as fitness function. The smaller the fitness function is, the closer the frequency response $H(e^{j\omega})$ of the designed filter is to the frequency response $|H_d(e^{j\omega})|$ of the ideal filter. At the end of the improved ABC
algorithm, the most abundant nectar source found by bees corresponds to the optimal filter coefficient of FIR filter, namely are \( h(0), h(1), \cdots, h(N - 1) \). According to the above introduction, the implementation steps of FIR filter based on improved ABC algorithm are described as follows.

Step 1 The frequency response of the ideal filter is given.
Step 2 Initialization of improved ABC algorithm. Determine the number of nectar source—SN, the maximum number of iterations—\( M \), the maximum number of nectar source mining—\( limit \), \( c_{\text{max}}, c_{\text{min}}, \alpha, \beta \), etc. Randomly generate \( SN \) initial nectar sources.

Step 3 The optimization process based on improved ABC algorithm.

Step 3.1 The values of the parameters to be optimized are given. The initial populations \( h(0), h(1), \cdots, h(N - 1) \) are generated, \( h(n) \in [-1, 1], n = 1, 2, \cdots, N - 1 \). Suppose that number of iterations \( t = 1 \).

Step 3.2 An employed bee is assigned for nectar source. The searching process is begun according to Eqs. (5), (6) and (7). A new nectar source \( V_i \) will be generated.

Step 3.3 The fitness value is calculated according to Eq. (11). The nectar source will be retained according to greedy algorithm.

Step 3.4 The probability of nectar source be followed is calculated by Eq. (3). The onlooker bees are searching and retaining the nectar source according to greedy algorithm.

Step 3.5 The algorithm determines whether the nectar source should be discarded. If true, the onlooker bees are changed into scout bees. Otherwise, go to step 3.7.

Step 3.6 The scout bees will generate new nectar source according to Eqs. (5), (6) and (7).

Step 3.7 Let \( t = t + 1 \). If the maximum numbers of iterations are satisfied, optimal parameters \( h(0), h(1), \cdots, h(N - 1) \) are output, go to Step 4. Otherwise, go to Step 3.2 and continue to execution.

Step 4 After the optimal filter coefficients of FIR filter are obtained, the performance of the filter is verified by experiments.

4 Simulation

In order to verify the effectiveness and feasibility of the FIR digital filter design method based on improved ABC algorithm and other algorithms, some simulation experiments are performed out. The simulation software is MATLAB 2010b. The configuration information of the simulation computer is CPU: Intel i7-4770 3.4 GHz, memory: 8 GBytes, operating system: Windows 7 professional.

This paper designs three typical FIR filters include low-pass, band-pass and band-stop digital filter. In this paper, the sampling points equal the FIR filter length. The length of the filter is the order of the filter. Assuming that the length of the filter is \( N \), the purpose of designing the filter is to obtain \( N \) filter systems, namely, the number of sampling points (the number of tap-weight). From the perspective of implementation, FIR filter includes transversal, cascade, frequency sampling and fast convolution. In this study, frequency sampling FIR filter is selected as the research object.

Unlike IIR filter, FIR filter can guarantee arbitrary amplitude frequency characteristics and strict linear phase frequency characteristics. At the same time, its unit sampling response is finite, so the FIR filter is a stable system. The frequency response between the sampling points of FIR filter is formed by the extension and superposition of the weighted interpolation function of each sampling point, which will have a certain error. These approximation errors may lead to the existence of poles in the designed FIR filter, but these poles will never exceed the unit range, so they will not pose a threat to the stability of FIR system. However, the error will affect the performance of FIR filter, which can be solved by increasing the number of sampling points \( N \), resulting in an increase in the amount of calculation. The sampling points of the band-stop and band-pass filters are slightly larger than the theoretical value.

The technical indicators of low-pass FIR digital filter are as follows.

\[
H_d(e^{j\omega}) = \begin{cases} 
1, & 0 \leq \omega \leq \omega_{cl} \\
0, & \omega_{ch} \leq \omega \leq \pi 
\end{cases} \tag{12}
\]

where \( \omega_{cl} = 0.4\pi, \omega_{ch} = 0.6\pi \). The sampling interval \( \Delta \omega \) is \( \frac{0.6 - 0.4}{3} \pi \text{ rad/s} \). The sampling points \( N \geq \frac{2\pi}{\Delta \omega} \) are 30. Therefore, the sampling points of low-pass FIR filter are 30.

The technical indicators of band-pass FIR digital filter are as follows.

\[
H_d(e^{j\omega}) = \begin{cases} 
1, & \omega_{cl1} \leq \omega \leq \omega_{ch1} \\
0, & 0 \leq \omega < \omega_{cl2}, \omega_{ch2} < \omega \leq \pi 
\end{cases} \tag{13}
\]

where \( \omega_{cl1} = 0.25\pi, \omega_{ch1} = 0.75\pi, \omega_{cl2} = 0.15\pi, \omega_{ch2} = 0.85\pi \). The sampling interval \( \Delta \omega \) is \( \frac{0.85 - 0.15}{5} \pi \text{ rad/s} \). The sampling points \( N \geq \frac{2\pi}{\Delta \omega} \) are 60. Therefore, the sampling points of low-pass FIR filter are chosen as 64.

The technical indicators of band-stop FIR digital filter are as follows.

\[
H_d(e^{j\omega}) = \begin{cases} 
0, & 0 \leq \omega < \omega_{cl1}, \omega_{ch2} < \omega \leq \pi \\
1, & 0 < \omega < \omega_{cl2}, \omega_{ch1} < \omega \leq \pi 
\end{cases} \tag{14}
\]

where \( \omega_{cl1} = 0.25\pi, \omega_{ch1} = 0.75\pi, \omega_{cl2} = 0.15\pi, \omega_{ch2} = 0.85\pi \). The sampling interval \( \Delta \omega \) is \( \frac{0.85 - 0.15}{5} \pi \text{ rad/s} \). The
sampling points $N \geq \frac{2B}{\Delta f}$ are 60. Therefore, the sampling points of low-pass FIR filter are chosen as 64.

In order to compare the convergence performance and the optimization effect of the proposed method, the same experiments are compared with GA (Szopos et al. 2016), PSO (Zhang et al. 2018), DE (Dash et al. 2017), ACO (Tsutsumi and Suyama, 2014), CSA (Sarangi et al. 2018) and standard ABC (Kockanat et al. 2018), respectively. The filter coefficients to be optimized are limited to $[-1, 1]$. The number of iterations of these algorithms is 500. The population number of these algorithms is 50. The parameters of all algorithms involved are given in Table 5.

In the experiment, these optimization algorithms run randomly 50 times. The optimal solution of each algorithm is recorded, and the optimal solution with the smallest fitness value among 50 results is selected as the filter coefficient to obtain the amplitude frequency response of the filter. Figure 1 shows the amplitude frequency response curves of the low-pass FIR digital filter designed by these optimization algorithms. Figure 2 shows the amplitude frequency response curves of the band-pass FIR digital filter designed by these optimization algorithms. Figure 3 shows the amplitude frequency response curves of the band-stop FIR digital filter designed by these optimization algorithms. It can be observed from Figs. 1, 2 and 3 that the amplitude frequency response of the designed FIR filter is closer to the ideal FIR filter. The filter can descend from the pass-band to the stop-band more smoothly; the pass-band ripple is closest to the zero phase digital filters.

Figures 4, 5 and 6 give the phase response of these optimization algorithms for low-pass, band-pass and band-stop FIR filters. Compared with other optimization algorithms, for low-pass, band-pass or band-stop filters, the stop-band attenuation of the designed FIR filter is smaller and closer to zero phase digital filter. The coefficients of the filter optimized by the improved ABC algorithm proposed in this paper are more reasonable, so better performance is achieved.

Figure 7 shows the fitness curve of the low-pass FIR digital filter designed by these optimization algorithms. Figure 8 shows the fitness curve of the band-pass FIR digital filter designed by these optimization algorithms. Figure 9 shows the fitness curve of the band-stop FIR digital filter designed by these optimization algorithms.

It can be seen from Figs. 7, 8 and 9 that the fitness curve of the improved ABC algorithm is faster than the other optimization algorithms, and the effect is very obvious. On analyzing the convergence behavior with existing other
evolutionary techniques it is observed that proposed improved ABC algorithm is better in stability, exploration ability. The main reason for the performance improvement is that in addition to the advantages of ABC algorithm itself, the improvement of local search strategy further improves the global optimization ability of the algorithm. Table 6 shows the average experimental results of the low-pass filters designed by these optimization algorithms. Table 7 shows the average experimental results of the
band-pass filters designed by these optimization algorithms. Table 8 shows the average experimental results of the low-stop filters designed by these optimization algorithms. From these tables, it can be concluded that the improved ABC algorithm is superior to other optimization algorithms in terms of robustness and solution. Namely, the other optimization algorithms are prone to premature and fall into the local optimal solution, which cannot guarantee the convergence to the optimal solution, and the robustness of these algorithms is poor. Because of the characteristics of the algorithm itself, the effect of FIR filter designed by GA is the most undesirable. If the population size, the number of iterations and other parameters of these algorithms are increased, it will undoubtedly increase the complexity of the algorithm itself. Compared with these algorithms, ABC algorithm can obtain better performance.
But ABC algorithm also exists the problem of limited local search ability. In the proposed ABC algorithm, the local search ability of ABC algorithm is enhanced by improving neighborhood search strategy, which effectively balances the contradiction between global search and local search ability, thus improving the performance of FIR filter.

Table 9 shows the performance parameters of the FIR low-pass filters designed by these optimization algorithms.

| Fitness value (dB) | GA       | PSO      | ABC      | ACO      | DE       | CSA      | Improved ABC |
|-------------------|----------|----------|----------|----------|----------|----------|--------------|
| Maximum value     | 7.3200   | 6.5700   | 6.2500   | 6.8422   | 6.3030   | 5.6202   | 6.8600       |
| Minimum value     | -5.1100  | -5.2200  | -5.7200  | -5.3211  | -5.3202  | -5.9206  | -6.1100      |
| Average value     | -3.0165  | -3.2044  | -3.4538  | -3.2241  | -3.0711  | -3.5547  | -3.7512      |
| Variance          | 12.6762  | 12.6488  | 11.7613  | 13.4850  | 12.1500  | 10.8647  | 11.0616      |
| Standard deviation| 3.5602   | 3.5565   | 3.4096   | 3.6722   | 3.0711   | 3.2962   | 3.0077       |

But ABC algorithm also exists the problem of limited local search ability. In the proposed ABC algorithm, the local search ability of ABC algorithm is enhanced by improving neighborhood search strategy, which effectively balances the contradiction between global search and local search ability, thus improving the performance of FIR filter.

Table 9 shows the performance parameters of the FIR low-pass filters designed by these optimization algorithms.
Table 8 The experimental results of the band-stop filter

| Algorithm   | Fitness value (dB) | GA   | PSO | ABC   | ACO   | DE   | CSA   | Improved ABC |
|-------------|--------------------|------|-----|-------|-------|------|-------|--------------|
| Maximum     | 7.2515             | 7.3511 | 6.9547 | 5.9800 | 6.0205 | 6.2212 | 7.2141       |
| Minimum     | −5.9101            | −6.1304 | −6.4608 | −6.2200 | −5.9903 | −6.5104 | −7.1036       |
| Average     | −2.3531            | −2.8592 | −3.2453 | −3.6446 | −3.3214 | −3.5722 | −3.6887       |
| Variance    | 20.4136            | 20.0909 | 18.6316 | 18.4576 | 18.3880 | 17.2242 | 16.7319       |
| Standard deviation | 4.5181            | 4.1681 | 4.0422 | 4.1685 | 4.1932 | 3.9018 | 3.8715       |

Table 9 The performance parameters of the FIR low-pass filters

| Algorithm   | Pass-band ripples (dB) | Stop-band ripples (dB) | Pass-band attenuation (dB) | Stop-band attenuation (dB) | Transition band error (p) |
|-------------|------------------------|------------------------|----------------------------|----------------------------|---------------------------|
| Improved ABC | 0.0443                 | 89.9605                | −0.0647                    | −43.2070                   | 0.0192                    |
| ABC         | 0.0814                 | 81.4750                | −0.0815                    | −31.9876                   | 0.0201                    |
| GA          | 0.1237                 | 56.0357                | −0.1145                    | −30.3467                   | 0.0218                    |
| PSO         | 0.0877                 | 75.7649                | −0.0878                    | −38.4269                   | 0.0204                    |
| ACO         | 0.0963                 | 60.5801                | −0.0965                    | −37.5362                   | 0.0212                    |
| DE          | 0.1142                 | 68.1037                | −0.1143                    | −35.9639                   | 0.0208                    |
| CSA         | 0.1027                 | 75.4765                | −0.1328                    | −34.6338                   | 0.0199                    |

Table 10 The performance parameters of the FIR band-pass filters

| Algorithm   | Pass-band ripples (dB) | Stop-band ripples (dB) | Pass-band attenuation (dB) | Stop-band attenuation (dB) | Transition band error (p) |
|-------------|------------------------|------------------------|----------------------------|----------------------------|---------------------------|
| Improved ABC | 0.0752                 | 112.5430               | −0.1582                    | −40.2232                   | 0.0204                    |
| ABC         | 0.0841                 | 106.5295               | −0.2340                    | −37.1766                   | 0.0217                    |
| GA          | 0.1080                 | 91.1102                | −0.1907                    | −29.7765                   | 0.0223                    |
| PSO         | 0.0866                 | 91.0407                | −0.1931                    | −40.0643                   | 0.0234                    |
| ACO         | 0.0945                 | 93.1061                | −0.2012                    | −39.5322                   | 0.0220                    |
| DE          | 0.0851                 | 92.8054                | −0.2206                    | −38.3745                   | 0.0218                    |
| CSA         | 0.1167                 | 103.9606               | −0.2327                    | −37.7095                   | 0.0219                    |

Table 10 shows the performance parameters of the FIR band-pass filters designed by these optimization algorithms. Table 11 shows the performance parameters of the FIR band-stop filters designed by these optimization algorithms. It can be seen from these tables that the filters designed by the improved ABC algorithm have smaller pass-band ripples, pass-band attenuation and transition band error, and larger stop-band ripples and stop-band attenuation. Therefore, the filters designed by the improved ABC algorithm have better performance parameters.

From the data in the three tables, we can see that the stop-band attenuation of FIR filter designed by the improved ABC algorithm is more than 10 dB larger than that of the other six algorithms, and the transition band and stop-band have obvious boundaries, the stop-band changes uniformly, and the pass-band attenuation is small. At the same time, the filter designed by the improved ABC algorithm has smaller transition band error, which shows that its transition band is narrower. In addition, the filter designed by the improved ABC algorithm has a small fluctuation of stop-band attenuation variance, which shows the good stability of the algorithm. For the three kinds of filters designed in the simulation, the improved ABC algorithm proposed in this paper has higher convergence speed and accuracy than other algorithms, so it has better performance. Among other optimization algorithms, ABC is close to PSO and DE, followed by ACO and CSA, and GA is the worst. Combining the fitness curves of Figs. 7, 8 and 9, it can be analyzed that GA is too easy to fall into local convergence and cannot reach the optimal solution.
compared with other algorithms. Compared with GA, CSA is more similar to PSO algorithm without optimal individual. In the later stage of iteration, CSA lacks vitality and optimization ability. Therefore, the performance of CSA is only better than GA and worse than other optimization algorithms. Therefore, in the design of low-pass, band-pass and band-stop filters, the improved ABC algorithm shows superior performance and high stability.

The $t$ test can determine the means of two groups, which are statistically different from each other or not. This paper performs $t$ test by individually comparing ABC, GA, PSO, ACO, DE and CSA with improved ABC. The high positive $t$ value indicates the superiority of the improved ABC over other optimization optimizations. Table 12 clearly indicates the superiority of the improved ABC over ABC, GA, PSO, ACO, DE and CSA for the design of FIR low-pass, band-stop, band-pass filters. All the $t$ values of pass-band ripple and stop-band ripple for other optimization algorithms are positive $t$ values, which signify the superiority of the improved ABC over other six optimization algorithms.

Table 13 shows the results of Kruskal–Wallis test for FIR filters designed by these optimization algorithms. The algorithm runs 20 times, takes the maximum absolute value error of filter error as the evaluation object, and sets the significance level to 0.05. As can be seen from Table 13, the p-values of improved ABC, ABC and ACO are greater than the significance level of 0.05, so the assumption that there is a significant difference in the performance of the designed FIR filter can be rejected. Meanwhile, the p-values of GA, CSA, PSO, DE are smaller than the significance level of 0.05, and it can be inferred that there are differences in the performance of the filters obtained by these algorithms at least twice. Therefore, it can be concluded that the FIR design performance of improved ABC, ABC and ACO is better than GA, DE, PSO and CSA.

In order to compare the real-time performance and space complexity of the algorithms, Table 14 shows the average optimization time and space occupied by memory required

### Table 11 The performance parameters of the FIR band-stop filters

| Algorithm | Pass-band ripples (dB) | Stop-band ripples (dB) | Pass-band attenuation (dB) | Stop-band attenuation (dB) | Transition band error ($\pi$) |
|-----------|------------------------|------------------------|----------------------------|----------------------------|----------------------------|
| Improved ABC | 0.0165 | 104.2122 | -0.0763 | -44.7402 | 0.0188 |
| ABC | 0.0174 | 74.2398 | -0.0977 | -37.7861 | 0.0195 |
| GA | 0.0220 | 50.2578 | -0.1119 | -30.2224 | 0.0203 |
| PSO | 0.0193 | 72.7989 | -0.0925 | -42.3580 | 0.0198 |
| ACO | 0.0187 | 61.0800 | -0.1099 | -40.2961 | 0.0202 |
| DE | 0.0179 | 68.2910 | -0.1269 | -38.6017 | 0.0210 |
| CSA | 0.0174 | 69.6475 | -0.1299 | -38.3721 | 0.0207 |

### Table 12 $t$ Test of improved ABC over other algorithms for low-pass, band-pass and band-stop FIR filters

| Algorithm | Low-pass filter | Band-pass filter | Band-stop filter |
|-----------|-----------------|------------------|------------------|
| ABC | 0.8924 | 4.2637 | 0.3028 | 5.4022 | 1.4205 | 3.0231 |
| GA | 0.7925 | 4.0027 | 0.3128 | 4.9956 | 1.3684 | 2.3028 |
| PSO | 0.7761 | 3.9620 | 0.2037 | 3.0174 | 2.9605 | 1.4309 |
| ACO | 0.9750 | 4.2307 | 0.1995 | 3.4485 | 2.0017 | 1.6821 |
| DE | 1.0024 | 1.9208 | 0.1753 | 3.0521 | 2.0482 | 2.0064 |
| CSA | 0.5062 | 2.0257 | 0.1027 | 2.9980 | 1.0264 | 2.6843 |

| Algorithm | $p$-value (Kruskal–Wallis test) |
|-----------|-------------------------------|
| Improved ABC | 3.124 |
| ABC | 2.957 |
| GA | 0.028 |
| PSO | 0.033 |
| ACO | 3.047 |
| DE | 0.030 |
| CSA | 0.025 |
for the FIR filters designed by these optimization algorithms. It can be seen from Table 14 that the improved ABC algorithm proposed in this paper has shorter optimization time. Although the time complexity of the improved ABC algorithm is increased, the actual calculation time is shortened because it can search the global optimal value faster. Furthermore, the results in Table 14 also show that the memory occupied by these optimization algorithms in the design of FIR filter is relatively close, which is also consistent with the conclusions obtained in Table 3.

5 Conclusions

In this paper, an improved ABC algorithm is exploited for the design of optimal digital FIR filter. A comparative and analytical study of FIR filter design using seven popular optimization techniques (improved ABC, ABC, GA, PSO, ACO, DE and CSA) is discussed. The performance of low-pass, band-pass and band-stop filters optimized by above seven optimization algorithms is studied. The aim of the filter design is to find the optimal filter coefficients having minimum relative error with respect to the ideal filter response. The simulation result clearly signifies that improved ABC performs better by means of magnitude response with high stop-band attenuation, optimum pass-band and stop-band ripples and smallest optimization time. The proposed improved ABC uses the information of the global optimal solution to guide the search of candidate solution, improves the development efficiency and obtains good optimization performance. From the simulation results, it is observed that improved ABC may be treated as an efficient tool to be used for optimal filter design. It is concluded that improved ABC is more efficient, accurate, faster and a better global optimizer than other six optimization algorithms. The future work of this paper is to implement the FIR filter design method proposed in this paper on FPGA to verify the actual performance of the filter.

Acknowledgements This work is partially supported by the Doctoral Scientific Research Foundation of Liaoning Province (Grant No. 20180540050).

Funding Fund was provided by Doctoral Scientific Research Foundation of Liaoning Province (Grant No. 20180540050).

Data availability Some or all data, models or code that supports the findings of this study is available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Aggarwal A, Rawat TK, Kumar M, Upadhyay DK (2015) Optimal design of FIR high pass filter based on L1 error approximation using real coded genetic algorithm. Eng Sci Technol Int J 18(4):594–602
Aggarwal A, Rawat TK, Kumar M, Upadhyay DK (2016) Efficient design of digital FIR differentiator using L-1-method. Radio-engineering 25(2):383–389
Aggarwal A, Rawat TK, Kumar M, Upadhyay DK (2018) Design of optimal band-stop FIR filter using L-1-norm based RCGA. Ain Shams Eng J 9(2):277–289
Agrawal N, Kumar A, Bajaj V (2017) A new design method for stable IIR filters with nearly linear-phase response based on fractional derivative and swarm intelligence. IEEE Trans Emerg Topics Comput Intell 1(6):464–477
Agrawal N, Kumar A, Bajaj V (2018) Design of digital IIR filter with low quantization error using hybrid optimization technique. Soft Comput 22(9):2953–2971
Agrawal N, Kumar A, Bajaj V, Singh GK (2019a) Design of bandpass and bandstop infinite impulse response filters using fractional derivative. IEEE Trans Industr Electron 66(2):1285–1295
Agrawal N, Kumar A, Bajaj V (2019b) A new method for designing of stable digital IIR filter using hybrid method. Circuits Syst Signal Process 38(5):2187–2226
Agrawal N, Kumar A, Bajaj V (2020) Design of infinite impulse response filter using fractional derivative constraints and hybrid particle swarm optimization. Circuits Syst Signal Process 39(12):6162–6190
Agrawal N, Kumar A, Bajaj V, Singh GK (2021) Design of digital IIR filter: A research survey. Appl Acoust 172:107669
Ahirwal MK, Kumar A, Singh GK (2013) EEG/ERP adaptive noise canceller design with controlled search space (CSS) approach in cuckoo and other optimization algorithms. IEEE-ACM Trans Comput Biol Bioinf 10(6):1491–1504
Ahirwal MK, Kumar A, Singh GK (2014) Adaptive filtering of EEG/ERP through bounded range artificial bee colony (BR-ABC) algorithm. Digital Signal Process 25:164–172
Alwahab DA, Zaghar DR, Laki S (2018) FIR filter design based on neural network. In: 11th International Symposium on Communication Systems, Networks and Digital Signal Processing. 8471878.

FIR digital filter design based on improved artificial bee colony algorithm

| Algorithm     | Time (s) | Memory (MB) |
|---------------|----------|-------------|
| Improved ABC  | 5.306    | 53          |
| ABC           | 6.329    | 51          |
| GA            | 11.367   | 62          |
| PSO           | 6.660    | 54          |
| ACO           | 6.508    | 56          |
| DE            | 7.361    | 50          |
| CSA           | 6.227    | 52          |
transformation: Maximum error invariance. Int J Numer Model Electron Networks Devices Fields 33(1):e2692
Chandra A, Chattopadhyay S (2016) A new strategy of image denoising using multiplier-less FIR filter designed with the aid of differential evolution algorithm. Multimed Tools Appl 75(2):1079–1098
Chauhan A, Sathish KP (2018) Implementation of FIR filter Mac unit by using neural networks in FPGA. In: International Conference on Advances in Computing, Communications and Informatics. 2496–2501.
Dash J, Dam B, Swain R (2017) Design of multipurpose digital FIR double-band filter using hybrid firefly fractionally differential evolution algorithm. Appl Soft Comput 59:529–545
Dash S, Dam B, Swain R (2020) Improved firefly algorithm based optimal design of special signal blocking IIR filters. Measurement 149:106986
Dwivedi AK, Ghosh S, Londhe ND (2016) Low power FIR filter design using modified multi-objective artificial bee colony algorithm. Eng Appl Artif Intell 55:58–69
Dwivedi AK, Ghosh S, Londhe ND (2017) Low-power FIR filter design using hybrid artificial bee colony algorithm with experimental validation over FPGA. Circuits Systems Signal Process 36(1):156–180
Dwivedi AK, Ghosh S, Londhe ND (2018) Review and analysis of evolutionary optimization-based techniques for FIR filter design. Circuits Syst Signal Process 37(10):4409–4430
Filip S (2016) A robust and scalable implementation of the Parks-McClellan algorithm for designing FIR filters. ACM Trans Math Softw 43(1):7
Hameed S, Pamarti S (2018) Design and analysis of a programmable receiver front-end with time-interleaved baseband analog-FIR filtering. IEEE J Solid-State Circuits 53(11):3197–3207
Hossin MA, Shi M, Thanh V, Son NT (2018) An adjustable window-based FIR filter and its application in audio signal de-noising. In: 3rd International Conference on Robotics and Automation Engineering. 248–252.
Jou YD, Chen FK (2011) WLS design of FIR Nyquist filter based on neural networks. Digital Signal Process 21(1):17–24
Kaplan D, Aryashev S, Veligosha A, Doynikova E, Lyakhov P, Butusov D (2020) Improving calculation accuracy of digital filters based on finite field algebra. Appl Sci-Basel 10(1):45
Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization. J Global Optim 39(3):459–471
Kim T, Kim S (2017) Remote pedestrians detection at night-time in FIR image using contrast filtering and locally projected region-based CNN. In: Proceedings of SPIE Conference on Infrared Technology and Applications XLIII. 101772G.
Kockanat S, Koza T, Karaboga N, Logoglu A (2018) Adaptive FIR filtering using ABC algorithm: a noise reduction application on mitral valve doppler signal. Elektronika Ir Elektrotechnika 24(5):62–68
Kumar M (2018) Optimal design of fractional delay FIR filter using cuckoo search algorithm. Int J Circuit Theory Appl 46(12):2364–2379
Kumar M (2019) Fractional order FIR differentiator design using particle swarm optimization algorithm. Int J Nuner Model Electron Netw Devices Fields 32(2):e2514
Kumar A, Agrawal N, Sharma I (2020) Design of finite impulse response filter with controlled ripple using cuckoo search algorithm. In: 3rd International Conference on Computer Vision and Image Processing (CVIP). 471–482.
Lee D, Lee D, Kim YH, Jeon HK, Kim BG, Kim LS (2020) A 10.8 Gb/s quarter-rate 1 FIR 1 IIR direct DFE with non-time-overlapping data generation for 4:1 CMOS clockless multplexer. IEEE Trans Circuits Syst II - Express Briefs 67(1):67–71
Li L, Muneeswaran V, Ramkumar S, Emayavaramban G, Ramirez GR (2019) Metheuristic FIR filter with game theory based compression technique- a reliable medical image compression technique for online applications. Pattern Recogn Lett 125:7–12
Ma RX, Wang YH, Hu W, Zhu XY, Zhang K (2018) Optimum design of multistage half-band FIR filter for audio conversion using a simulated annealing algorithm. In: 13th IEEE Conference on Industrial Electronics and Applications. 74–78.
Maeda Y, Fukushima N, Matsuo H (2018) Taxonomy of vectorization patterns of programming for FIR image filters using kernel subsampling and new one. Appl Sci-Basel 8(8):1235
Mahata S, Saha SK, Kar R, Mandal D (2016) Optimal design of wideband infinite impulse response fractional order digital integrators using colliding bodies optimisation algorithm. IET Signal Proc 10(9):1135–1156
Mittal S (2020) A survey of FPGA-based accelerators for convolutional neural networks. Neural Comput Appl 32(4):1109–1139
Miyata T, Asou H, Aikawa N (2018) Design method for FIR filter with variable multiple elements of stopband using genetic algorithm. In: 23rd IEEE International Conference on Digital Signal Processing. 8631793.
Naderian S, Salemmia A (2017) Method for classification of PQ events based on discrete Gabor transform with FIR window and T2FK-based SVM and its experimental verification. IET Gener Transm Distrib 11(1):133–141
Padmapriya S, Jagadeeswarri M, Prabha L (2019) Design of an optimized twin mode reconfigurable adaptive FIR filter architecture for speech signal processing. Informacije Midem-J Microelectron Electron Compon Mater 49(4):241–254
Park KY, Lee HS (2019) Design of de-noising FEM-FIR filters for the evaluation of temporal and spatial derivatives of measured displacement in elastic solids. Mech Syst Signal Process 120:524–539
Ryu DK, Lee CJ, Park SK, Lim MT (2020) Unbiased FIR filtering with incomplete measurement information. Int J Control Autom Syst 18(2):330–338
Saha SK, Kar R, Mandal D, Ghoshal SP (2012) Seeker optimisation algorithm: application to the design of linear phase finite impulse response filter. IET Signal Proc 6(8):763–771
Saha SK, Kar R, Mandal D (2013a) Design and simulation of FIR band pass and band stop filters using gravitational search algorithm. Memetic Comput 5(4):311–321.
Saha SK, Kar R, Mandal D (2013b) Bacteria foraging optimisation algorithm for optimal FIR filter design. Int J Bio-Inspir Comput 5(1):52–66
Saha SK, Ghoshal SP, Kar R (2013c) Cut swarm optimization algorithm for optimal linear phase FIR filter design. ISA Trans 52(6):781–794
Saranig SR, Panda R, Das PK, Abraham A (2018) Design of optimal high pass and band stop FIR filters using adaptive Cuckoo search algorithm. Eng Appl Artif Intell 70:67–80
Shao P, Wu ZJ, Zhou XY, Tran CT (2017) FIR digital filter design using improved particle swarm optimization based on refraction principle. Soft Comput 21(10):2631–2642
Sharma I, Kuldeep B, Kumar A, Singh VK (2016) Performance of swarm based optimization techniques for designing digital FIR filter: A comparative study. Eng Sci Technol Int J 19(3):1564–1572
Song XY, Zhao M, Xing SY (2020) A multi-strategy fusion artificial bee colony algorithm with small population. Expert Syst Appl 142:142
Szopos E, Neag M, Saracut I, Popescu V, Topa M (2016) Synthesis tool based on genetic algorithm for FIR filters with user-defined
magnitude characteristics. Circuits Syst Signal Process 35(1):253–279
Tsutsumi S, Suyama K (2014) Design of FIR filters with discrete coefficients using ant colony optimization. Electron Commun Japan 97(4):30–37
Umadevi S, Vigneswaran T (2019) Reliability improved, high performance FIR filter design using new computation sharing multiplier: suitable for signal processing applications. Clust Comput 22:13669–13681
Wu C, Xu XZ, Zhang XR, Zhao L (2015) Sparse FIR filter design based on simulated annealing algorithm. Adv Electr Comput Eng 15(1):17–22
Xu W, Zhang RH, Li AY, Shi BY, Yan SX (2018) Least squares design of 2-D FIR notch filters based on the Hopfield neural networks. In: International Conference on Electronics and Electrical Engineering Technology. 102–106.
Zhang CF, Yue Z, Wang FJ, Yan D (2018) A novel framework for FIR digital filter design based on P system with PSO. International Conference on Information Systems and Computer Aided Education. 297–307.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.