Chinese Journal of Electronics
vol. 33, no. 5, pp. 1165–1180, September 2024
https://doi.org/10.23919/cje.2022.00.407

RESEARCH ARTICLE

An Efficient and Fast Area Optimization Approach for Mixed Polarity Reed-Muller Logic Circuits

Yuhao ZHOU[1], Zhenxue HE[2], Jianhui JIANG[1], Xiaojun ZHAO[2], Fan ZHANG[2], Limin XIAO[3], and Xiang WANG[4]

1. *School of Software Engineering, Tongji University, Shanghai 201804, China*
2. *Key Laboratory of Agricultural Big Data of Hebei Province, Hebei Agricultural University, Baoding 071001, China*
3. *School of Computer Science and Engineering, Beihang University, Beijing 100191, China*
4. *School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*

Corresponding author: Zhenxue HE, Email: hezhenxue@buaa.edu.cn
Manuscript Received November 29, 2022; Accepted March 22, 2023

Abstract — Area has become one of the main bottlenecks restricting the development of integrated circuits. The area optimization approaches of existing XNOR/OR-based mixed polarity Reed-Muller (MPRM) circuits have poor optimization effect and efficiency. Given that the area optimization of MPRM logic circuits is a combinatorial optimization problem, we propose a whole annealing adaptive bacterial foraging algorithm (WAA-BFA), which includes individual evolution based on Markov chain and Metropolis acceptance criteria, and individual mutation based on adaptive probability. To address the issue of low conversion efficiency in existing polarity conversion approaches, we introduce a fast polarity conversion algorithm (FPCA). Moreover, we present an MPRM circuits area optimization approach that uses the FPCA and WAA-BFA to search for the best polarity corresponding to the minimum circuits area. Experimental results demonstrate that the proposed MPRM circuits area optimization approach is effective and can be used as a promising EDA tool.

Keywords — Area optimization, Combinatorial optimization problem, Fast polarity conversion algorithm, Mixed polarity Reed-Muller, Bacterial foraging algorithm.

Citation — Yuhao ZHOU, Zhenxue HE, Jianhui JIANG, *et al.*, "An Efficient and Fast Area Optimization Approach for Mixed Polarity Reed-Muller Logic Circuits," *Chinese Journal of Electronics*, vol. 33, no. 5, pp. 1165–1180, 2024. doi: 10.23919/cje.2022.00.407.

I. Introduction

At present, integrated circuits have entered the development stage of ultra-large-scale integrated circuits and system-on-chips. The integration of chips continues to increase, the speed is also accelerating, and the area is getting larger and larger. However, the ever-increasing area has caused various portable devices to encounter power supply difficulties, chip overheating and increased power consumption [1], [2]. These problems will lead to more and more expensive cooling costs [3]. Modern EDA technology conducts logic synthesis and optimization, testing and design, and database management and support by customizing various standards. The technology takes hardware description language (HDL) as the core, adopts concept-driven and rule-driven, and starts from high-level system-level design, including scheme design and verification, logic circuit design, and even the underlying ASIC layout design, all completed by automated means, so as to realize the integration of design, testing and process. Therefore, polarity optimization techniques can be introduced in higher levels of circuit design (such as logic or behavior levels) to optimize the area of integrated circuits [4]–[8]. Previous research on area optimization focused on Boolean logic circuits. For some arithmetic circuits and communication circuits, XNOR/OR-based Reed-Muller (RM) logic circuits have more advantages in terms of power consumption, area and testabili-

Associate Editor: Prof. Hongbin SUN, Xi'an Jiaotong University.

ty [9], [10]. Polarity is the key factor of Reed-Muller logic, which directly determines the simplification of function, and then affects the power consumption and area performance of the circuit. Reed-Muller logic circuits optimization is to search for a certain polarity in a specific polarity space to optimize some performance indicators of the corresponding circuit, usually called the optimal polarity.

The mixed polarity Reed-Muller (MPRM) expression is an important criterion to measure the complexity of the Reed-Muller circuits, and the MPRM logic circuit with n input variables has 3^n different polarities, that is, the optimization space is 3^n. So, it is necessary and vital to find a novel optimization theory and approach to minimize MPRM logic expressions. The current research on Reed-Muller logic circuits optimization is mainly aimed at fixed polarity Reed-Muller (FPRM) logic circuits, and its polarity search space is a proper subset of MPRM logic circuit. Related research has shown that the power and area performance of MPRM logic circuits are better than those of FPRM logic circuits [11], [12]. But at the same time, the time and space complexity of MPRM logic circuit is higher than that of FPRM logic circuit, which makes the optimization of MPRM logic circuit much more complex than that of FPRM logic circuit. Therefore, new breakthroughs are urgently needed in the optimization theory and solution approach of MPRM logic circuits.

The area optimization for XNOR-based MPRM circuits is more difficult than that for FPRM circuits. Firstly, most of the researches focused on XOR/AND circuits [13]–[15]. However, after benchmark tests, it has been verified that XNOR/OR-based circuits have a more concise expression [16]–[18], which will further reduce the area. Although some logic minimization algorithms for MPRM have been proposed above, they cannot handle multiple input variables, and they are less efficient when polarity conversion. Secondly, previous researches focused on FPRM circuits and XOR-based EXOR-sum-of-products (ESOP) circuits [19]–[21]. The approaches proposed in the above works can be used for circuits with multiple input variables, but most of them use traditional algorithms and do not consider time constraints. In addition, translating to XNOR logic via De Morgan's law would be further time consuming. Finally, polarity conversion is a prerequisite for measuring circuit complexity. The above works are mostly converted from Boolean logic to specified polarity [17], [18], [22]. As far as we know, there is no approaches of conversion between polarities currently. Therefore, there is an urgent need for a conversion approach between polarities to improve efficiency. The change of true form and complement form of various polarities is analyzed, and we propose a fast polarity conversion algorithm (FPCA) between polarities. It is necessary to introduce a search algorithm that can handle multiple inputs and higher efficiency to optimize the MPRM circuits area based on XNOR/OR.

The area optimization and polarity conversion algorithm optimization for MPRM logic circuit is a computationally hard problem. In addition, we also need to solve the double nested combinatorial optimization problem. Compared with the existing work, our main contributions are as follows:

1) We propose a whole annealing adaptive bacterial foraging algorithm (WAA-BFA). In the chemotaxis behavior of the algorithm, the non-directional random search of bacteria will affect the convergence performance of the algorithm. In other words, the chemotaxis behavior is equivalent to the traveling salesman problem (TSP), and we need to consider the overall search effect. So MPRM circuits optimization considering individual chemotaxis is a double nested combinatorial optimization problem. So, we nested the whole annealing optimization (WAO) on the basis of BFA optimization for in-depth exploration and optimization. In the migration behavior of the algorithm, we use an adaptive migration optimization (AMO) strategy to improve the global convergence accuracy and speed up the convergence speed at the same time.

2) We propose a fast polarity conversion algorithm (FPCA). The conversion from 0 polarity to any polarity can be completed by the algorithm. This algorithm is more efficient than the Boolean conversion. As far as we know, this is the first time that an XNOR/OR-based MPRM circuits fast polarity conversion algorithm has been proposed.

3) We propose an MPRM circuits area optimization approach (MAOA) to search MPRM circuits with minimum area based on WAA-BFA and FPCA. As far as we know, this is the first time to apply the BFA optimization to RM circuits area optimization.

4) The validity of each algorithm was verified by experiments using a set of benchmarks from the Microelectronics Center of North Carolina (MCNC). The experiment results show that compared with the three existing area optimization approaches for XNOR/OR-based MPRM circuits, the maximum area optimization rate obtained by MAOA reaches 84.07%; compared with the current conversion approaches, the maximum time saving rate obtained by FPCA reaches 99%.

The paper is structured as follows. In Section II, we introduce preliminary preparations. Section III introduces a fast polarity conversion algorithm. Section IV introduces the WAA-BFA. Section V introduces the area optimization algorithm for MPRM. Section VI reports experimental results. Conclusions are presented in Section VII.

II. Preliminary Preparations

1. MPRM expression

Any n-variable Boolean function can be expressed canonically by the product of sums (POS) form based on AND/OR, as shown below:

$$f(x_{n-1}, x_{n-2}, \ldots, x_0) = \prod_{i=0}^{2^n-1} (a_i + M_i) \tag{1}$$

where the binary form of the subscript i can be expressed as $(i_{n-1}\ i_{n-2}\ \cdots\ i_0)$, $0 \le k < n$; M_i represents the ith maxterms, the + represents the logical sum, and $a_i \in \{0, 1\}$, which corresponds to the presence or absence of the ith maxterms, respectively.

The logical function f can also be expressed as MPRM based on XNOR/OR operations as follows:

$$f(x_{n-1}, x_{n-2}, \cdots, x_0) = \odot \prod_{i=0}^{2^n-1} (d_i + s_i) \tag{2}$$

where $\odot\prod$ represents the XNOR operations; the + represents the logical sum, and $d_i \in \{0, 1\}$ indicates whether the s_i items appear in the expression, respectively. s_i is based on OR operations and is represented as $s_i = \widehat{x}_{n-1} + \widehat{x}_{n-2} + \cdots + \widehat{x}_0$. Multi-input s_i can be decomposed into m_i two-input OR operations, then $m_i = \sum_{k=0}^{n-1} \bar{i}_k - 1$. Formula (2) is decomposed into two-input OR operations and two-input XNOR operations, and their term number calculation formula is $\lambda = \sum_{i=0}^{2^n-1} (\bar{d}_i \cdot (\sum_{k=0}^{n-1} \bar{i}_k - 1))$ and $\eta = \sum_{i=0}^{2^n-1} (\bar{d}_i - 1)$, respectively. Here $\bar{d}_i$ and $\bar{i}_k$ represent the complement form. So, the area of the MPRM logic circuits can be presented as

$$\begin{aligned} \text{AreaCost} &= \lambda + \eta \\ &= \sum_{i=0}^{2^n-1} \left(\bar{d}_i \cdot \left(\sum_{k=0}^{n-1} \bar{i}_k - 1 \right) \right) + \sum_{i=0}^{2^n-1} (\bar{d}_i - 1) \\ &= \sum_{i=0}^{2^n-1} \bar{d}_i \cdot \sum_{k=0}^{n-1} \bar{i}_k - 1 \end{aligned} \tag{3}$$

As shown in (4), in RM expression, each variable has three modes (i.e., true (x_k), complement ($\bar{x}_k$), mixed-mode ($\widehat{x}_k$)), and the ternary form of polarity P is $(p_{n-1}\ p_{n-2}\ \cdots\ p_0)$. If mixed-mode exists in the RM expression, the circuit is called an MPRM circuit. MPRM is represented by the ternary 0, 1, or 2, that is, there are 3^n polarities for a function of n variables. Thus, the number of maxterms λ (OR gates) and η (XNOR gates) existed can be expressed as the area. Due to limited space, a detailed description of logic synthesis can be found in [11], [12], and [17].

$$p_k = \begin{cases} 0, & \text{if } x_k \text{ appears in true form} \\ 1, & \text{if } x_k \text{ appears in complement form} \\ 2, & \text{if } x_k \text{ appears in mixed form} \end{cases} \tag{4}$$

2. Traditional polarity conversion approach

Nowadays, the approaches of XNOR/OR-based MPRM polarity conversion are only from Boolean to any polarity. At present, the recognized convenient and effective conversion algorithm was proposed by in 2020 [18]. The conversion algorithm from Boolean to MPRM polarity is described as Algorithm 1.

Algorithm 1 The conversion algorithm from Boolean to MPRM polarity

Input: Polarity $p = (p_{n-1}\ p_{n-2}\ \cdots p_0)$, the OR term coefficient $i = (i_{n-1}\ i_{n-2}\ \cdots i_0)$, initialize count $= n$.

Output: The OR term i.

```
for j = count–1 to 0 do
    if p_j == 1 then
        Each i_j ⊕ p_j// (i.e., i_j and p_j exclusive-OR operation);
    end if
end for
for j = 0 to count do
    if p_j == 0|| p_j == 1 and the coefficient of the term is (i_{n-1}... i_{j+1} 1 i_{j-1} ... i_0) then
        Generate new OR term coefficient (i_{n-1}... i_{j+1} 0 i_{j-1} ... i_0);
    else if p_j == 2
        continue
    end if
    for all newly generated OR terms do
        if the newly generated OR term coefficient is same as the original OR term coefficient then
            Delete the newly generated OR term coefficient and the original OR term coefficient;
        else
            Add the newly generated OR term coefficient to the original OR term coefficient;
        end if
    end for
end for
```

III. Fast Polarity Conversion Algorithm

To evaluate each polarity of logic function and find the best polarity, the corresponding logic expansion of each polarity must be obtained by the polarity conversion algorithm [23]. However, the transformation of multi-valued logic functions is not applicable to MPRM logic. Therefore, efficient polarity conversion algorithm is the prerequisite for area optimization of all MPRM logic circuits. There are two types of polarity conversion: polarity conversion between Boolean logic and RM logic, and polarity conversion between logic expansions under different polarities of RM. Currently, most of the polarity conversion uses Algorithm 1 mentioned in Section II. Since there are many max-terms in Boolean functions, Algorithm 1 takes a lot of time. Therefore, we propose the FPCA based on the principle of graph folding approach. The graph folding approach was first applied to Boolean conversion. The following uses a three-variable function as an example to introduce the transformation between two logical expansions implemented by the graph fold-

ing approach.

$$f_1(x_1,x_2,x_3) = (x_1+x_2+x_3)(x_1+\bar{x}_2+x_3) \cdot (\bar{x}_1+x_2+\bar{x}_3)(\bar{x}_1+\bar{x}_2+x_3) \quad (5)$$

As shown in Figure 1, according to (5), the product term is first converted into the form of the maxterms, and the "0" and "1" in the logic function are folded and transformed, and the two parts are subjected to the XNOR operation (i.e., four steps of A→B→C→D). Finally, all inverse variables are eliminated and the RM expression is obtained. The graph folding approach is convenient to operate, but the graph folding approach based on serial or parallel operation needs to expand the original term to the minimum term/maximum term. When the input variable N of the circuit increases, the amount of computation increases exponentially. We have absorbed the advantages of the convenient operation of the graph folding approach and adopted the parallel processing approach of data, and for the first time proposed an algorithm that can realize the conversion between MPRM polarities (that is, the conversion of 0 polarity to any polarity). The FPCA is showed as Algorithm 2.

Algorithm 2 The FPCA from 0 polarity to MPRM polarity

Input: Polarity $0 = (0\,0\ldots0)$, polarity $p = (p_{n-1}\,p_{n-2}\ldots p_0)$, the OR term coefficient $i = (i_{n-1}\;i_{n-2}\;\ldots i_0)$, initialize count $= n$.

Output: All OR term coefficients.

```
for j = count–1 to 0 do
  if p_j == 1 and the coefficient of the term is (i_{n-1}...i_{j+1} 0 i_{j-1} ... i_0) then
    Generate new OR term coefficient (i_{n-1}...i_{j+1} 1 i_{j-1} ... i_0);
  else if p_j == 2 and the coefficient of the term is (i_{n-1} ...i_{j+1} 1 i_{j-1} ... i_0) then
    Generate new OR term coefficient (i_{n-1}...i_{j+1} 0 i_{j-1} ... i_0);
  else if p_j ==0
    continue
  end if
  for all newly generated OR terms do
    if the newly generated OR term coefficient is same as the original OR term coefficient then
      Delete the newly generated OR term coefficient and the original OR term coefficient;
    else
      Add the newly generated OR term coefficient to the original OR term coefficient;
    end if
  end for
end for
```

Example 1 Suppose that 0 polarity $f^0(x_2,x_1,x_0) = \odot\prod(0,3,4,5,6,7)$, find the mixed polarity XNOR/OR expression with polarity 19 (the ternary form is 201). First, the 0 polarity maxterms are displayed in the first column. We generate new terms for x_0 and x_2. According to Algorithm 2, we can derive the MPRM expression

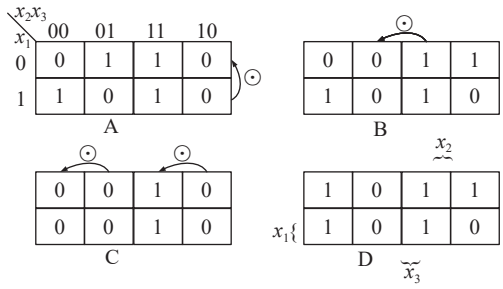

Figure 1 Graph folding approach for Boolean conversion.

Maxterms	Newterms	Newterms	Newterms
$x_2x_1x_0$	x_0	x_1	x_2
0 0 0	0 0 1		0 0 0
0 1 1	1 0 1		0 1 0
1 0 0	1 1 1		
1 0 1			
1 1 0			
1 1 1			

Figure 2 Mixed polarity conversion using FPCA.

with required polarity. As shown in Figure 2, we have

$$f^{19}(x_2,x_1,x_0) = \odot\prod(1,2,3,4,6) = (x_2+x_1)\odot(x_2+\bar{x}_0)\odot x_2 \odot(\bar{x}_2+x_1+\bar{x}_0)\odot(\bar{x}_2+x_0) \quad (6)$$

And the logical decomposition diagram of (6) is shown in Figure 3.

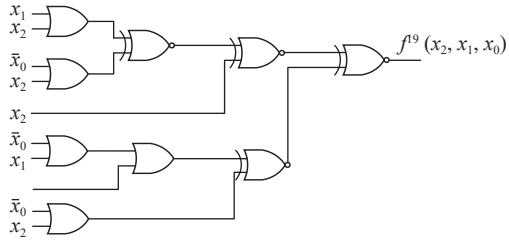

Figure 3 Logical decomposition diagram.

IV. Whole Annealing Adaptive Bacterial Foraging Algorithm

Searching for the best polarity corresponding to an MPRM circuit with a minimum area from the huge polarity optimization space is a computationally hard problem. In recent years, the bacterial foraging algorithm (BFA) has been widely used in the field of optimization due to its strong global search capabilities [24]–[26]. The BFA mainly searches for the optimal solution of the problem through the iterative calculation of chemotaxis operation, replications operation and migrations operation. Chemotaxis operation is the core idea of the BFA, and literature [27] introduced the Lyapunov stability theorem in control theory to prove and analyze the stability in the chemotaxis operation. In chemotaxis operation,

the individual swims through the quorum sensing mechanism to transmit signal values. Let $P(j,k,l)=\{\theta^i(j,k,l), i=1,2,\ldots,S\}$ represent the current position of each individual i in the population, and the influence value of signal transmission between bacteria in the population as

$$\begin{aligned} &J_{\mathrm{cc}}(\theta,P(j,k,l)) \\ &=\sum_{i=1}^{s}J_{\mathrm{cc}}\left(\theta,\theta^i(j,k,l)\right) \\ &=\sum_{i=1}^{s}\left[-d_{\mathrm{attractant}}\exp\left(-w_{\mathrm{attractant}}\sum_{m=1}^{D}\left(\theta_m-\theta_m^i\right)^2\right)\right] \\ &\quad+\sum_{i=1}^{s}\left[-h_{\mathrm{repellant}}\exp\left(-w_{\mathrm{repellant}}\sum_{m=1}^{D}\left(\theta_m-\theta_m'\right)^2\right)\right] \end{aligned} \tag{7}$$

where $d_{\mathrm{attractant}}$, $w_{\mathrm{attractant}}$, $h_{\mathrm{repellant}}$ and $w_{\mathrm{repellant}}$ are respectively the depth of attraction, the width of attraction, the height of repulsion and the width of repulsion force.

The algorithm flow is shown in Figure 4. BFA mainly uses chemotaxis operation to update the position of individual bacteria for optimization, accelerates the convergence speed through replication operation, and uses migration operation to prevent the algorithm from falling into a local optimum.

The above works are to solve the problems of continuity optimization. The area optimization of MPRM logic circuits is a high-dimension combinatorial optimization problem. Therefore, we propose WAA-BFA for area optimization of MPRM logic circuits. This section starts with the optimization mechanism of BFA, analyzes the role of each step in the algorithm and the influence of parameter design on the algorithm in detail, and combines the WAO deep exploration and AMO strategy with the chemotaxis behavior and migrations behavior respectively. The details of WAA-BFA are as follows.

1. WAO deep exploration

The chemotaxis operation in BFA only moves according to the fitness of the current direction, which makes the role of random search seem minimal [28]–[31]. As shown in Figure 5, when the bacteria are swimming in one of the directions, if their fitness is not good, they will immediately stop swimming and the bacteria will search for the direction again. In chemotaxis operation, it is difficult to decide whether the bacteria will accept the current solution after choosing the direction. This problem is similar to the TSP (i.e., whether to choose the point closest to the current city). The continuous and blind searches of bacteria will interfere with the convergence of the algorithm, and will fall into a local extremum, reducing the performance of converging to the global optimal solution. Therefore, in the study of BFA, we need to optimize the acceptance mechanism of chemotaxis behavior.

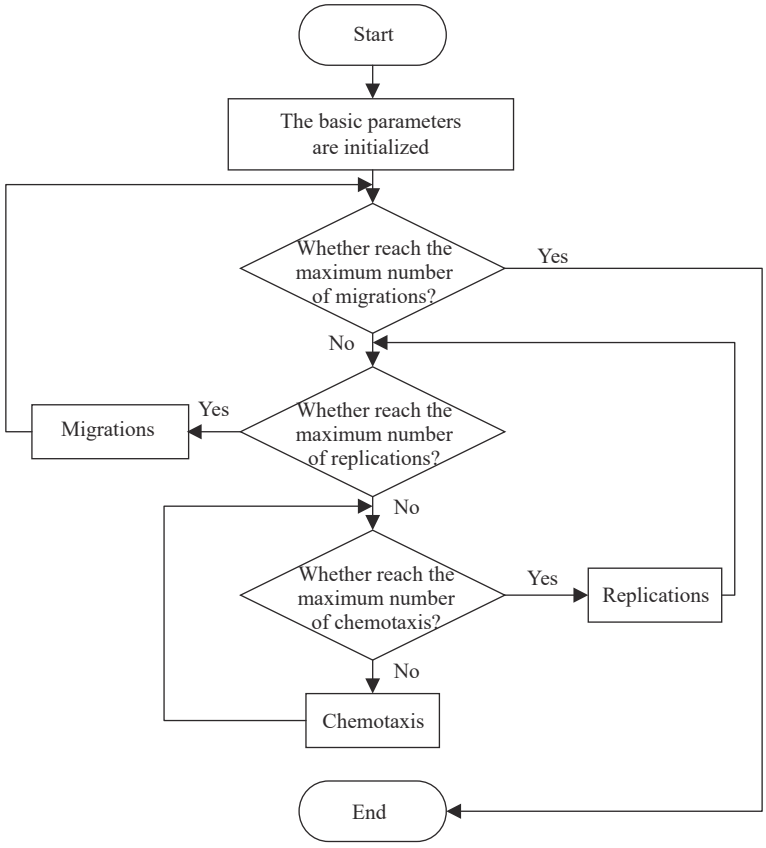

Figure 4 BFA flow chart.

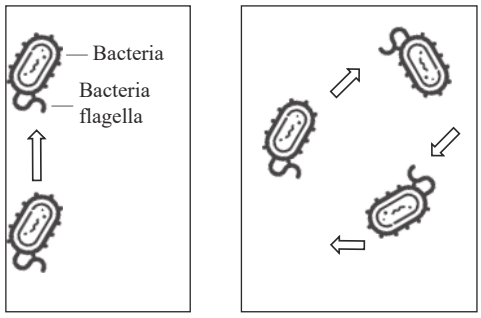

Figure 5 Swimming and turning of bacteria.

The stochastic optimization idea based on the iterative solution strategy has achieved good results in the simulated annealing algorithm (SA) [32], [33]. The SA establishes a mathematical model based on the cooling process of the solid material, which can be extended to general combinatorial optimization problems. The iterations of the SA are divided into the cooling down iterations and the same temperature iterations (Markov chain length L) [34]–[36]. The Markov chain represents the optimization degree of the optimal state at each temperature. That is the number of new solutions produced by bacteria in the same direction. In the SA, the temperature is reduced according to the set function, the new solution is accepted with a certain probability, the optimal local solution can be randomly jumped out, and the optimal global solution is randomly searched in the solution space. We hope that the chemotaxis behavior of bacteria is not limited to the current local extremum, and the individual bacteria can fully consider all the excellent schemes in the solution space. Therefore, we introduce the equilibrium probabilistic acceptance strategy of SA into BFA to solve such combinatorial optimization problems (i.e., chemotaxis behavior of bacteria), and propose the WAO deep exploration.

The WAO compares the algorithm's evolution to an annealing process and adds the Metropolis acceptance criteria and the Markov chain to the chemotaxis operation. The new fitness function value of individual i is shown in (8) after the completion of a chemotaxis operation. WAO is a new type of bacteria foraging algorithm that allows the parent to participate in the competition. It has a more efficient local search performance. The state sequence of the whole annealing algorithm is determined by (9) and (10). Metropolis acceptance criterion is also called the equilibrium probability equation.

$$J^i(j+1,k,l) = J^i(j+1,k,l)+J_{\mathrm{CC}}\left(\theta^i(j+1,k,l), P(j,k,l)\right) \tag{8}$$

$$p = \begin{cases} 1, & f_j < f_i \\ \exp(-(f_j - f_i/t)), & f_j \geq f_i \end{cases} \tag{9}$$

$$t_{k+1} = \lambda \times t_k,\ k = 0,1,\ldots \tag{10}$$

where t_k is the current solid temperature (i.e., current number of iterations), t_0 is set to 100, and $\lambda \in (0,\ 1)$ is the temperature cooling rate. $P(j,\ k,\ l)$ represents the position information of individual i in the optimization space after the jth chemotaxis operation, the kth replications operation, and the lth migrations operation. When set to state j, the energy of the solid is f_j. Let i represent the equilibrium state of the previous iteration. In state i, the solid energy is f_i. Compare the internal energy of two state objects, if the energy in state j is less than that in state i, that is, $f_j < f_i$, then state j is an acceptable ideal state. If the energy at state j is greater than the energy at state i, (i.e., $f_j > f_i$), then state j is irrational state. At this point, the probability of accepting the non-ideal solution is calculated according to the equilibrium probability (9). After a series of cooling optimization process, the particles in the solid state gradually tend to be stable. That is, the probability of equilibrium acceptance decreases gradually.

Boltzmann distribution function is shown in (11). It is worth noting that the probability value F conforms to the distribution of the function. In the whole annealing algorithm in this section, each state of the particle inside the solid is equivalent to a solution in the problem model, and the movement of this state in the large search space Ω forms a Markov chain of finite length, and its probability distribution p is shown in (12).

$$f(v) = \sqrt{\left(\frac{m}{2\pi kT}\right)^3} 4\pi v^2 \exp(-mv^2/2kT) \tag{11}$$

$$p = \exp(f(i)/t_k) \Big/ \sum_{j\in\Omega} \exp(f(j)/t_k) \tag{12}$$

As shown in Figure 6, WAO search is a process (1→2→3→4) of temperature jump to the poorer solution, that is to say, by giving the bacteria individual a sudden change in the search process that is time-varying and eventually tends to zero. WAO is an optimization strategy that effectively prevents individual bacteria from falling into a local minimum and eventually tending to a global optimal serial structure. Algorithm 3 shows the WAO.

Algorithm 3 The WAO

Input: A polarity $f_j = (p_{n-1}\ p_{n-2}\ \ldots p_0)$, the temperature cooling rate λ, paratmeter t_k, $t_{\min}$.
Output: A new polarity $f_i = (p_{n-1}\ p_{n-2}\ \ldots p_0)$.

```
while t_k > t_min do
  for count = 0 to L (Markov chain length) do
    Search f_new around polarity f_j according to (8);
    Compute f_new according to (3);
    At least one (p_n ∈ f_new)! = (p_n ∈ f_j);
    if f_new > f_j then
    f_i = f_new;
    else
    Compute P according to (9);
    if rand > P then
      f_i = f_new;
    end if
```

```
13:        end if
14:      end for
15:      t_{k+1}=λ×t_k;
16:   end while
17: return f_i = (p_{n-1} p_{n-2} ... p_0).
```

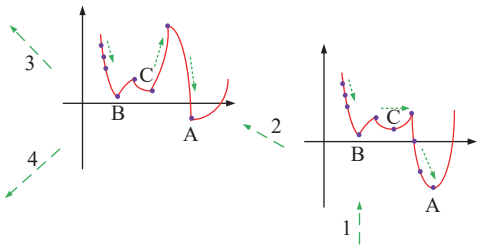

Figure 6 WAO search behavior.

2. AMO strategy

The migrations operation is to allow bacteria to mutate and jump out of local extremes. Traditional BFA uses fixed migration probability, and the bacteria with the best and worst fitness values can be eliminated with equal probability, which limits the optimization ability and efficiency of the algorithm [37]–[39]. Therefore, bacteria close to the optimal solution location may be replaced by bacteria far away from the optimal solution location [40], [41], thus affecting the performance of the algorithm. To overcome the above problems, we make the migration probability of individuals change linearly with fitness values and realize the tradeoff between search and randomness in different ways. Therefore, it is a complex problem for individuals to adapt to change migrations rate during evolution. According to the characteristics of migrations probability, formal description is an effective way to conform to dynamic change [42]–[44].

According to the relationship between the iterations g and fitness f, we change the adaptive migration probability curve. The adaptive formula is described by (13). Equation (13) satisfies the two requirements. First, under this probability distribution, the best bacteria $f_{\max}$ have the lowest probability, and the worst bacteria $f_{\min}$ will definitely be eliminated. Second, at the beginning of the search, it is recommended that the individuals explore the entire search space. Therefore, a larger value is given. Instead, it is recommended that individuals dig at the end of the search, which can be achieved by assigning small values.

$$p_m = \begin{cases} k_1 + \dfrac{k_2 \times g}{G} \times \dfrac{k_3 \times (f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_1 + k_2 \times k_3, & f < f_{\text{avg}} \end{cases} \tag{13}$$

where G is the maximum iterations; k_2 is the weight coefficients of iteration; k_3 is the weight coefficients of fitness value. In this paper, a small correction constant k_1 is added to p_m, which can avoid the p_m of the optimal solution of each generation being zero and solve the problem that the algorithm is prone to fall into local optimization. On the other hand, p_m is appropriately increased, so that new individuals can be generated faster in the early iteration and search speed is accelerated.

Figure 7 shows the probability distributions of different angles of the proposed AMO. Since the fitness value of each individual changes with the evolution of the algorithm, we define a three-dimensional probability function graph for the three weights in order to explain the role of the three weights in AMO more clearly. As shown in Figure 7, AMO can be adjusted at both ends of the mean fitness value according to probability. In the optimization problem of minimum value, the direction of adjustment of probability p_m is opposite to that of maximum value optimization.

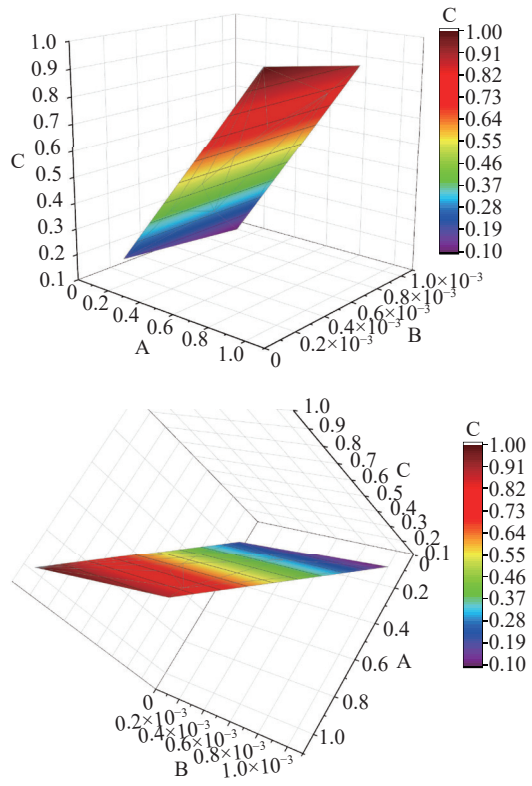

Figure 7 Linear probability distribution.

From different angles of probability distribution, it can be seen that this is a strategy with linear adjustment at both ends, which has a higher possibility of jumping out of local optimal value than that of linear adjustment at one end. In addition, we will get three optimal weight coefficients. Then, the probability of the high-order bacteria gradually decreases, and the probability of the low-order bacteria gradually increases. The improved AMO strategy adjusts dynamically according to the number of evolutionary iterations and search space, meets the changing rule of step size in actual search, and achieves the balance between individual global search and local search. Especially for high-dimensional combinatorial optimization problems, the linear probability distribution of AMO strategy is more effective in global convergence because the search space explodes ex-

ponentially with the dimension. Algorithm 4 shows the AMO.

Algorithm 4 The AMO

Input: Polarity f_i, Population P={f_1 …,f_n }, maximum iterations G.

Output: Migration rate p_m.

```
for i={1,2,...,n} do
    Compute f_i according to (3);
end for
Compute f_max, f_min and f_avg;
if f_i >= f_avg then
    Compute p_m according to (13);
else
    p_m = k_1+k_2×k_3;
end if
return p_m.
```

3. Algorithm description

WAA-BFA is described in Algorithm 5, which includes individual evolution based WAO deep exploration, and individual mutation based on AMO strategy.

Algorithm 5 The WAA-BFA

Input: Evolutionary parameters.

Output: The best solution.

```
Randomly generate initial population;
for i=1:n do
        Calculate the fitness values;
end for
for mi=0:mi_max (Maximum number of migrations) do
    if t_k> t_min then
        for re=0:re_max (Maximum number of replications) do
            for count=0 to L (Markov chain length) do
                Perform WAO deep exploration;
            end for
            Bacterial replication operations;
            re ←re+1;
        end for
        t_{k+1}=λ×t_k;
        Compute p_m based on AMO strategy;
    else
        break
    end if
end for
Output the best solution.
```

V. Area Optimization for MPRM Logic Circuits

Since the area optimization of MPRM logic circuits is an NP-hard problem, we propose an MAOA based on the WAA-BFA and FPCA to search for the optimal polarity corresponding to MPRM logic circuits with minimum area. Chromosome encoding, WAA-BFA for MPRM and MAOA are discussed in detail as follows.

1. Chromosome encoding

As shown in Section II, the polarity of the MPRM expression can be displayed by substituting 0, 1, or 2, so the polarity of the MPRM expression will be encoded in ternary form.

Example 2 The MPRM expression of 7 variables "2222011" is encoded as shown in Figure 8.

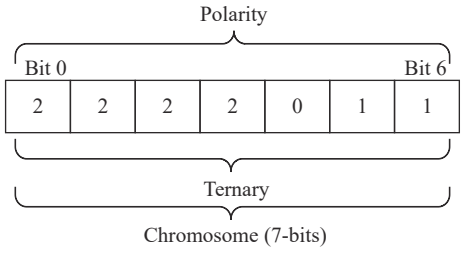

Figure 8 Chromosome encoding.

Therefore, the fitness function of MAOA can be defined as follows:

$$\text{fitness}(c_i) = 1.0/\text{objective}() \tag{14}$$

where fitness(c_i) denotes fitness function of the ith chromosome, objective() denotes objective function corresponding to circuit performance.

2. WAA-BFA for MPRM

WAA-BFA mainly uses WAO deep exploration to update the position of individual bacteria for optimization and uses AMO strategy to prevent the algorithm from falling into a local optimum. Figure 9 shows examples of two operations.

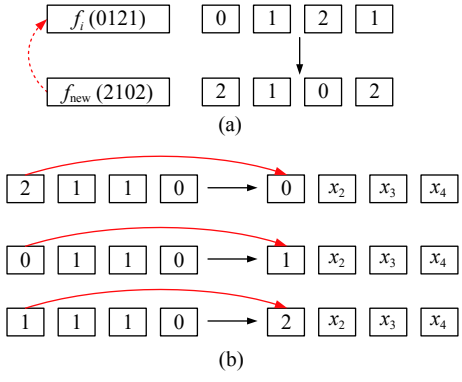

Figure 9 Overview of operations. (a) WAO deep exploration for chemotaxis operation; (b) AMO strategy for migrations operation.

Example 3 WAO deep exploration for MPRM. Suppose a polarity with four variables is (0, 1, 2, 1). As shown in Figure 9(a), the values of x_1, x_2, x_3, and x_4 are updated according to (8). Suppose the updated value is f_{new} = (2, 1, 0, 2), and compute f_{new} and P according to (3) and (9), respectively. If rand > P, $f_i = f_{\text{new}}$.

Example 4 AMO strategy for MPRM. Suppose a

polarity with four variables is (x_1, x_2, x_3, x_4), and the polarity bit that needs to be migrated is x_1. As shown in Figure 9(b), the migration probability is calculated according to the proposed AMO strategy. If the migration condition is met, then 2→0, or 0→1, or 1→2.

3. MAOA process

The flow chart of the MAOA is shown in Figure 10. The concrete implementation steps for the MAOA are as follows:

Step 1: N bacteria individuals were initialized, and fitness function was constructed.

Step 2: Use FPCA for polarity conversion and calculate the fitness value of individual bacteria according to (3).

Step 3: Whether the minimum temperature condition is met. If the condition is not met, go to step 4; otherwise, the optimal polarity value is outputted and end.

Step 4: Whether the maximum number of replications condition is met. If the condition is not met, go to step 5; otherwise, skip to step 8.

Step 5: Whether the maximum number of chemotaxis condition is met. If the condition is not met, go to step 6; otherwise, skip to step 7.

Step 6: The bacteria randomly choose the direction and perform the Markov chain forward operation in each

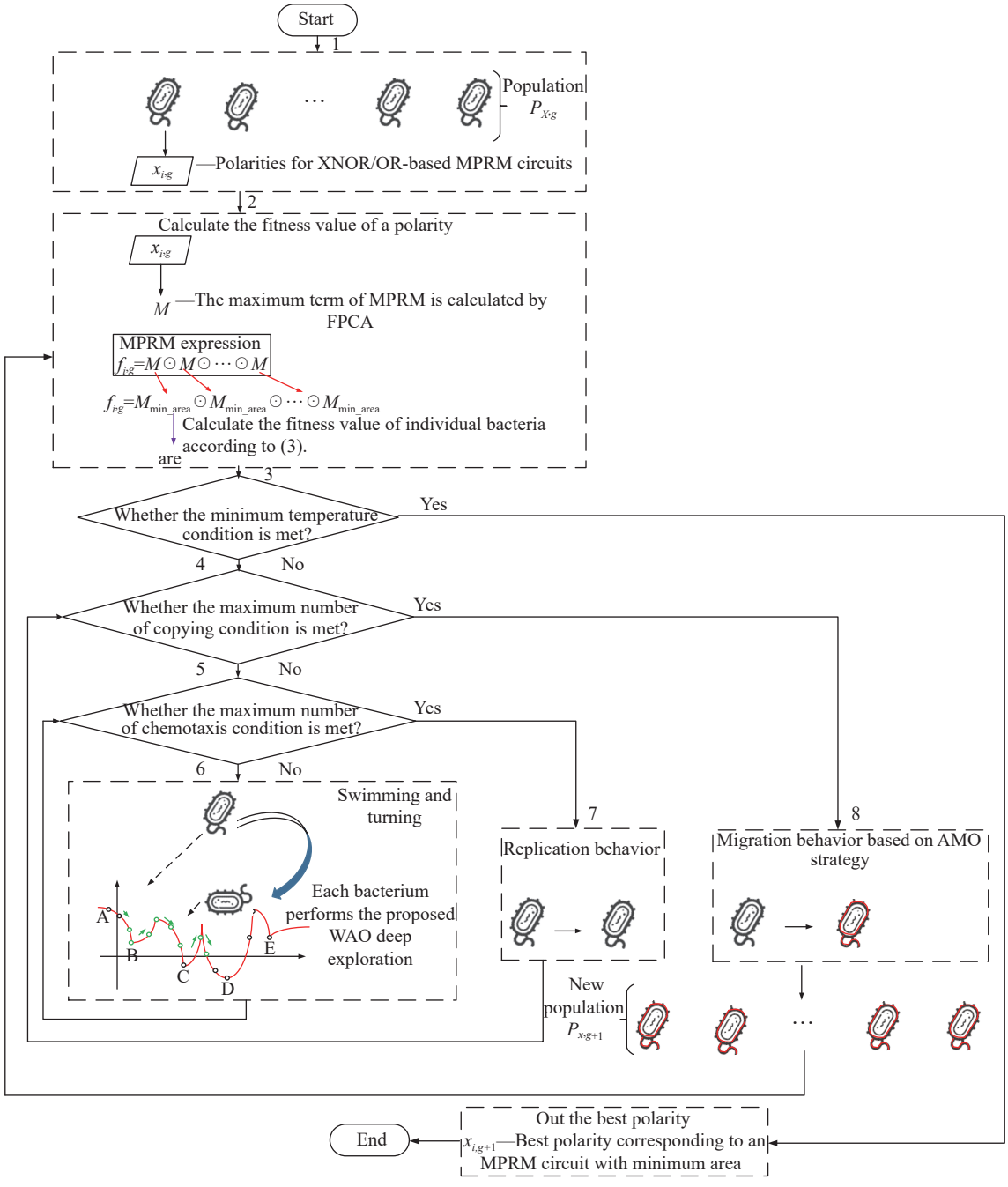

Figure 10 MAOA flow chart.

direction according to WAO. The current information of individual i is expressed as $\boldsymbol{\theta}^i = [\theta_1^i, \theta_2^i, \ldots, \theta_D^i]$ in the D-dimensional vector, $i = 1, 2, \ldots, S$, $P^i(j, k, l)$ represents the location information of individual i in the optimization space after the jth convergence operation, the kth replication operation and the l migrations operation. Then skip to step 5.

Step 7: Copy $n/2$ individual bacteria for the next initial population. The number of chemotaxis operations goes to zero, and the temperature is cooled. Then skip to step 4.

Step 8: The temperature is cooled, and the migration probability of bacteria is calculated according to WAO and AMO strategies. The number of replications operations goes to zero, then skip to step 2.

VI. Experiments

All the algorithms we proposed will be verified in this section. The experimental algorithms all use C language and run on an Intel Core i7-8750H CPU 8 GB RAM system. In addition, the experiments were conducted using a set of MCNC benchmarks. All test circuits have unique logical expression. The criterion for the end of the experiment is to fix the maximum running time and the tenth best solution has not been improved. Each circuit is randomly selected, and the characteristics of the circuit can be found in [45]. The scale of the test circuits represents the scale of the polarity search space, which is based on the input variable n. The experiments are listed as follows:

1) Experiment 1: Parameters verification
2) Experiment 2: MAOA verification
3) Experiment 3: FPCA verification
4) Experiment 4: Convergence verification

1. Experiment 1: Parameters verification

For the proposed MAOA, five major parameters (k_1, k_2, k_3, L and P) were calibrated by experiments design. Firstly, three key parameters k_1, k_2 and k_3 were designed by orthogonal experiment. We chose the three-level test, and the test levels are shown in Table 1. Secondly, due to the extensibility characteristics of P and L, we used the control variable approach to conduct repeated experiments to determine the variables. The purpose of control variable experiment is to select a set of excellent parameters and converge to the optimal solution in an acceptable time. Finally, all the suggested parameters are listed.

Table 1 Orthogonal test levels

Factor level	k_1	k_2	k_3
A1	0.001	0.1	0.1
A2	0.003	0.3	0.3
A3	0.005	0.5	0.5

According to the number of parameters and the number of factor levels of each parameter, orthogonal array L_9 (3^3) was selected and the experiment was repeated for 5 times to get the average value. In addition, the values of A1, A2 and A3 are given in Table 1. Figure 11 shows the convergence curves of five runs, where *B-C-D-E-F* represent the results of five runs, respectively.

It can be seen from Table 2 that the influence of parameters on the performance of the algorithm changes greatly. The AVERAGE represents the average of the area obtained from 5 runs, which reflects the optimization accuracy of the test scheme and is the main purpose of the algorithm parameter optimization. The smaller the AVERAGE, the higher the fitness value of the level test. In order to express the influence of parameters on MAOA performance more clearly, we carefully draw the convergence curve, so as to observe the stability of parameters on the optimization of test function more intuitively.

It can be seen that level test Figure 11(a) has the smallest average and that test No. 2 has the strongest stability in the convergence diagram. We expect that the algorithm can be closed to a minimum extreme point in each run. The stability of trial No. 2 has played an important role in improving the optimization performance of the algorithm. In addition, we use the maximum acceptable run time as a termination condition. Finally, the parameter settings of MAOA are drawn in Table 3.

2. Experiment 2: MAOA verification

Three recently proposed algorithms are compared to demonstrate the validity of MAOA in finding minimum area MPRM logic circuits. It includes genetic algorithm (GA)-based AOA [18], discrete particle swarm (DPSO)-based AOA [17], and metaheuristic algorithm (MA)-based AOA [24], were chosen. In consideration of randomness and fairness, each approach was repeated 10 times under the MCNC reference circuits and the approach parameters being compared were set according to the original publication.

As shown in Table 4, we perform tests on circuits of various scales, including large-scale circuits with multiple input variables. The benchmark in all experiments represents the name of the circuit being tested. The BEST and AVERAGE represent the minimum area of the search and the average of the minimum area of ten searches. For example, the third column and the seventh row represents the minimum area searched by the BFA, and the third column and the ninth row represents the minimum area searched by the MAOA. Then Save1 in ARR_best represents the area reduction rate (ARR) of MAOA on the best value compared with the GA. The circuit area is calculated as described in Section II and can be referred to (3). The ARR can be defined as follows:

$$\mathrm{ARR} = \frac{\mathrm{Area}_\beta - \mathrm{Area}_\alpha}{\mathrm{Area}_\beta} \times 100\% \tag{15}$$

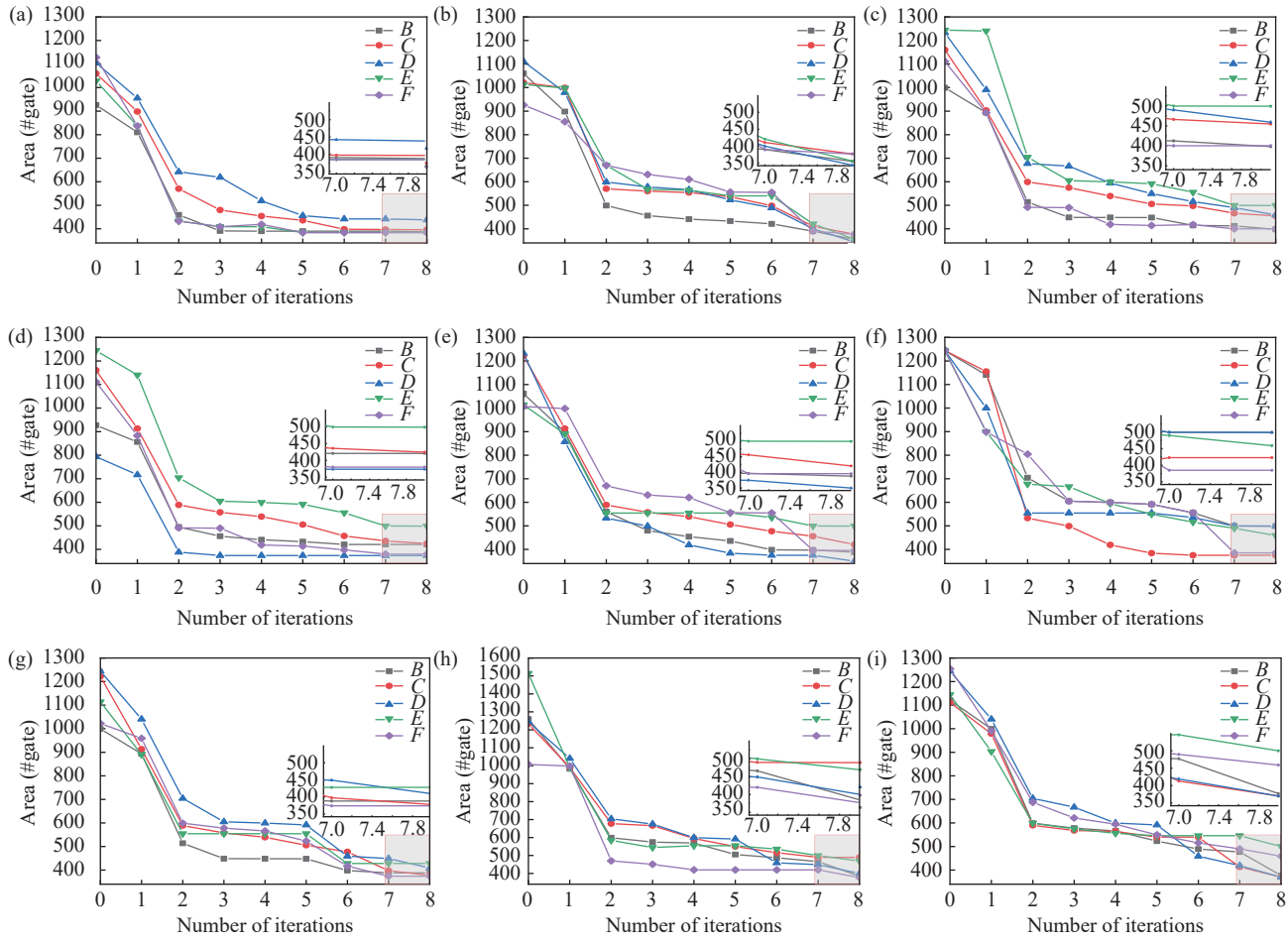

Figure 11 Convergence curves of different parameters (see Table 2 and Table 1 for the details).

Table 2 Orthogonal experiment

Level test	Column			
	k_1	k_2	k_3	AVERAGE
Figure 11(a)	A1	A1	A1	397.8
Figure 11(b)	A1	A2	A2	361.8
Figure 11(c)	A1	A3	A3	419.7
Figure 11(d)	A2	A1	A3	442.1
Figure 11(e)	A2	A2	A1	410.7
Figure 11(f)	A2	A3	A2	443.4
Figure 11(g)	A3	A1	A2	395.3
Figure 11(h)	A3	A2	A3	423.9
Figure 11(i)	A3	A3	A1	415.1

where Area_β and Area_α represent the area searched by each method, respectively.

From the above experiments, we can conclude that MAOA is superior to the present algorithm. Compared with GA, the maximum ARR of MAOA was 80.20% in ARR_best; compared with DPSO, the maximum ARR of MAOA was 84.07% in ARR_best; compared with BFA, MAOA has a maximum area saving rate of 65.00% in ARR_best. The above results can be explained by the following reasons:

Table 3 Parameters settings

Parameter	Value
Variable k_1	0.001
Variable k_2	0.3
Variable k_3	0.3
Population size P	30
Initial temperature t_0	100
Cooling rate λ	0.95
Markov chain length L	5
Chemotaxis number	5
Acceptable time (s)	60

1) The mutation probability of GA and the choice of crossing point affect the convergence of the algorithm, which is particularly sensitive to the control parameters. The flying speed of DPSO particles affects the convergence of the algorithm, and the particles often fly out of the boundary value.

2) The proposed MAOA introduces the WAO deep exploration into the chemotaxis behavior of bacteria, which can effectively avoid falling into the local optimum and eventually reach the global optimum by giving the search process a time-variant probability jump

Table 4 Area comparison results

Benchmark	Value	rd53	con1	5xp1	rd84	sao2	br1	br2	table3	amd	alu4	table5	bcd	e64
Scale	–	3^5	3^7	3^7	3^8	3^{10}	3^{12}	3^{12}	3^{14}	3^{14}	3^{14}	3^{17}	3^{26}	3^{65}
GA	BEST	19	21	37	55	179	149	88	1826	346	485	64	86	886
	AVERAGE	21.2	26.4	41.9	57.9	212.5	179.9	104	2788.2	393.9	519.6	116.3	129.3	1929.3
DPSO	BEST	19	28	37	57	175	140	105	1807	403	567	85	99	1099
	AVERAGE	23	35.5	41.4	58.7	183.2	178.5	148.1	3381.9	505.1	691.1	197.6	274.1	2033
MA	BEST	21	21	37	55	175	159	84	1917	363	505	92	106	500
	AVERAGE	21.8	31.9	43.8	56.5	202.9	188.3	105.3	2990.9	426.1	676.5	138.3	263.6	1200.5
MAOA	BEST	19	21	37	55	175	126	78	1584	311	479	50	58	175
	AVERAGE	20.4	23.5	37	56.5	183	150.3	80.2	1790.5	343.4	484.8	63.8	100.8	421.2
ARR_best	Save1	0.00	0.00	0.00	0.00	2.23	15.44	11.36	13.25	10.12	1.24	21.88	32.56	80.20
	Save2	0.00	25.00	0.00	3.51	0.00	10.00	25.71	12.34	22.83	15.52	41.18	41.41	84.07
	Save3	9.52	0.00	0.00	0.00	0.00	20.75	7.14	17.37	14.33	5.15	45.65	45.28	65.00
ARR_average	Save1	3.77	10.98	11.69	2.42	13.88	16.45	22.88	35.78	12.82	6.70	45.14	22.04	78.17
	Save2	11.30	33.80	10.63	3.75	0.11	15.80	45.85	47.06	32.01	29.85	67.71	63.23	79.28
	Save3	6.40	26.33	15.53	0.00	9.81	20.18	23.84	40.14	19.41	28.34	53.87	61.76	64.91

that eventually approaches zero. MAOA is a serial structure optimization algorithm that avoids the blind search of individual bacteria. In addition, MAOA will develop the effective information carried by the optimal solution generated in each iteration, which ensures the convergence direction of the algorithm.

3) The proposed AMO strategy adopts adaptive migrations probability based on various weight coefficients to avoid the destruction of good solutions in the process of evolution. In addition, MAOA conducts multiple operations from extensive search to meticulous search in the optimization process. Using AMO strategy, the algorithm can not only ensure the comprehensiveness and accuracy of the search, but also quickly jump out of the local optimum. Experiments show that excellent adaptive strategies can improve the convergence speed and accuracy greatly, especially for multi-peak functions.

3. Experiment 3: FPCA verification

In this section, we compare Algorithm 2 with the currently recognized valid conversion algorithm (Algorithm 1) proposed in 2020 to prove the superiority of FPCA (Algorithm 2). To ensure the fairness of the experiment, we used circuits with different input sizes and different population sizes. The conversion time for each algorithm on different circuits is shown as experimental results in Table 5. In addition, the time saving rate (TSR) can be defined as follows:

$$\mathrm{TSR} = \frac{\mathrm{A1\_time} - \mathrm{A2\_time}}{\mathrm{A1\_time}} \times 100\% \tag{16}$$

where A1_time and A2_time represent the conversion time used by each algorithm, respectively.

Table 5 The conversion algorithm comparison results

Benchmark	Value	alu3	b2	misex2
Scale	–	3^5	3^{16}	3^{25}
$N = 10$	A1_time (s)	0.026	25	1439.1
	A2_time (s)	0.023	8.201	50.908
$N = 50$	A1_time (s)	0.116	317.788	>6000
	A2_time (s)	0.097	18.508	55.1
$N = 100$	A1_time (s)	0.184	766.3	>12000
	A2_time (s)	0.166	22.828	59.171
TSR	$N = 10$	12%	67%	96%
	$N = 50$	16%	94%	>99%
	$N = 100$	10%	97%	>99%

In Table 5, benchmark is the circuits name, and N is the population scale. A1_time and A2_time represent the conversion time used by each algorithm, respectively. For example, 12% is the TSR of Algorithm 1 compared to Algorithm 2 in alu3 circuits where N is 10. It can be seen that Algorithm 2 can show greater advantages, and the maximum TSR can reach more than 99%. The experimental results can be explained by the following reasons:

1) To detect the merits and demerits of a certain polarity, the expansion formula under the polarity to be evaluated should be obtained first through polarity conversion. As we all know, Boolean function has a large number of maximum terms, and each variable needs to be processed in turn during conversion. Only one variable can be operated in each cycle, so the efficiency of polarity conversion is low.

2) The proposed FPCA eliminates the effect of negative deflection by inverting the list bar and realizes the conversion between polarities. The FPCA performs polarity switching in a natural order from zero to any polarity, with fewer bits between adjacent polarities requiring fewer base operands and faster polarity switching.

4. Experiment 4: Convergence verification

In order to show the search capabilities of MAOA, GA, DPSO and MA more intuitively, the smallest area searched during the iterative process of circuits of different scales is accumulated and averaged to draw area convergence curve. The abscissa is the number of iterations, and the ordinate is the average area of the test circuit running 10 times. As shown in Figures 12–17, MAOA can search the smallest area with the least number of iterations. The main reason for this is that the proposed WAO deep exploration and AMO strategy accelerate the convergence speed of the algorithm.

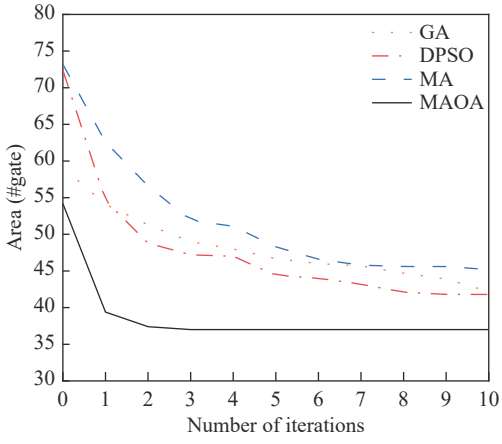

Figure 12 5xp1 convergence graph.

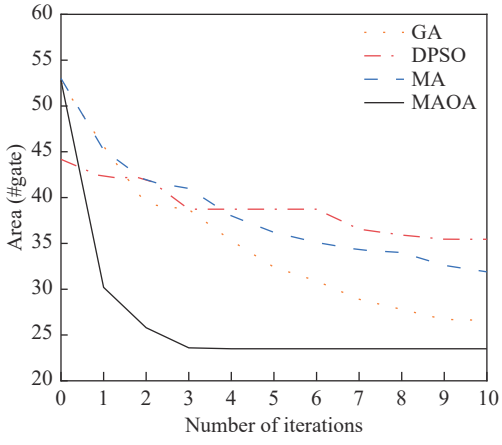

Figure 13 con1 convergence graph.

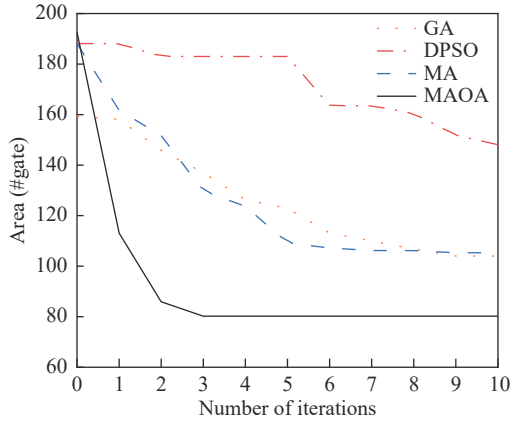

Figure 14 br2 convergence graph.

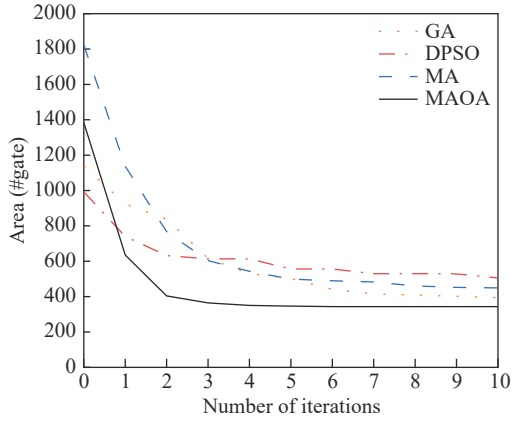

Figure 15 amd convergence graph.

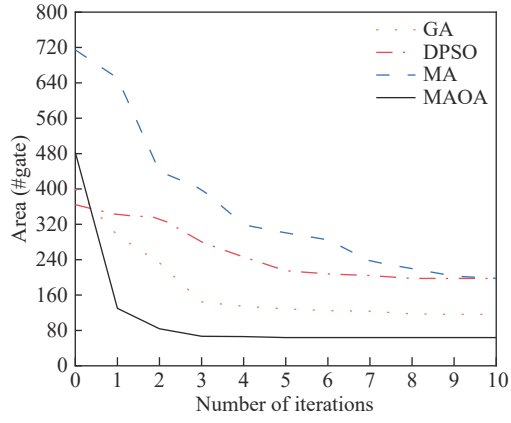

Figure 16 table5 convergence graph.

VII. Conclusion

We propose a WAA-BFA that applies WAO deep exploration and AMO strategies to chemotaxis operation and migration operation to deal with nested combinatorial optimization problems. Then, we propose an FPCA, which can quickly and effectively complete the conversion from 0 polarity to any polarity.

In addition, we propose an MAOA for MPRM logic circuits based on WAA-BFA and FPCA, which searches the best polarity corresponding to an MPRM circuit.

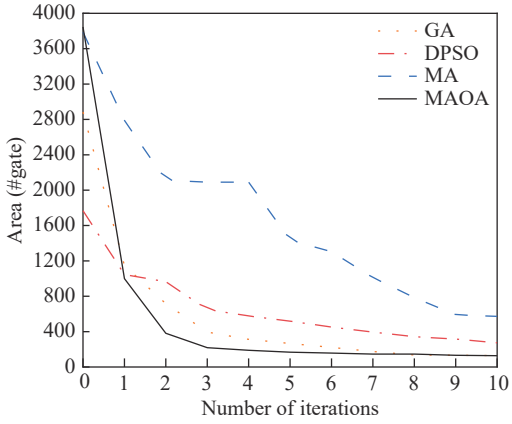

Figure 17 bcd convergence graph.

The experiment mainly includes four parts, namely, comparison of experimental algorithms on the parameters, comparison of experimental algorithms on the area, FPCA time verification, and comparison of experimental algorithms on the convergence. The experimental results confirm that MAOA is effective and can be used as a promising EDA tool.

There are several possible directions to extend this work in the future. Firstly, WAA-BFA can be studied to deal with the benchmark functions, so that our algorithm can have better performance. Secondly, delay optimization has entered our field of vision, and it is also an interesting work to study the delay optimization of MPRM logic circuits. Finally, we can further consider applying our proposed algorithm to sequential benchmark circuits.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62102130 and 61974105), the Central Government Guides Local Science and Technology Development Fund Project (Grant No. 226Z0201G), the Natural Science Foundation of Hebei Province (Grant No. F2020204003), the Hebei Youth Talents Support Project (Grant No. BJ2019008), the Science and Technology Research Projects of Higher Education Institutions in Hebei Province (Grant No. QN2022095), the Basic Scientific Research Funds Research Project of Hebei Provincial Colleges and Universities (Grant No. KY2022073), and the Key R&D Program of Hebei Province (Grant No. 21327407D).

References

[1] H. W. Chen, X. Y. Xue, C. S. Liu, *et al.*, "Logic gates based on neuristors made from two-dimensional materials," *Nature Electronics*, vol. 4, no. 6, pp. 399–404, 2021.

[2] P. Yin, Z. Shu, Y. J. Xia, *et al.*, "A low-area and low-power comma detection and word alignment circuits for JESD204B/C controller," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 7, pp. 2925–2935, 2021.

[3] S. H. Gunther, F. Binns, D. M. Carmean, *et al.*, "Managing the impact of increasing microprocessor power consumption," *Intel Technology Journal*, vol. 5, no. 1, pp. 1–9, 2001.

[4] M. Chen, R. R. Zheng, Y. T. Fu, *et al.*, "An analog control strategy with multiplier-less power calculation circuit for flyback microinverter," *IEEE Transactions on Power Electronics*, vol. 36, no. 8, pp. 8617–8621, 2021.

[5] H. Liang, Y. S. Xia, L. B. Qian, *et al.*, "Low power 3-input AND/XOR gate design," *Journal of Computer-Aided Design & Computer Graphics*, vol. 27, no. 5, pp. 940–945, 2015. (in Chinese)

[6] Z. X. He, L. M. Xiao, F. Gu, *et al.*, "An efficient and fast polarity optimization approach for mixed polarity Reed-Muller logic circuits," *Frontiers of Computer Science*, vol. 11, no. 4, pp. 728–742, 2017.

[7] P. J. Wang, Z. H. Wang, R. Xu, *et al.*, "Conversion algorithm for MPRM expansion," *Journal of Semiconductors*, vol. 35, no. 3, article no. 035007, 2014.

[8] D. L. Bu and J. H. Jiang, "An efficient optimization algorithm for multi-output MPRM circuits with very large number of input variables," in *Proceedings of the 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, Chongqing, China, pp. 228–232, 2014.

[9] A. Das and S. N. Pradhan, "Design time temperature reduction in mixed polarity dual Reed-Muller network: A NSGA-II based approach," *Advances in Electrical and Computer Engineering*, vol. 20, no. 1, pp. 99–104, 2020.

[10] C. Carvalho and G. V. G. L. Neumann, "The next-to-minimal weights of binary projective Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6300–6303, 2016.

[11] Z. X. He, Y. H. Pan, K. J. Wang, *et al.*, "Area optimization for MPRM logic circuits based on improved multiple disturbances fireworks algorithm," *Applied Mathematics and Computation*, vol. 399, article no. 126008, 2021.

[12] A. Das and S. N. Pradhan, "Area-power-temperature aware AND-XOR network synthesis based on shared mixed polarity Reed-Muller expansion," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 12, pp. 35–46, 2018.

[13] X. Wang, R. Zhang, W. K. Wang, *et al.*, "Polarity searching for MPRM logic circuit based on improved adaptive genetic algorithm," in *Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, Beijing, China, pp. 1354–1358, 2015.

[14] Y. H. Zhou, Z. X. He, T. Wang, *et al.*, "Area and power optimization approach for mixed polarity Reed-Muller logic circuits based on multi-strategy bacterial foraging algorithm," *Applied Soft Computing*, vol. 130, article no. 109720, 2022.

[15] Z. X. He, L. M. Xiao, Z. S. Huo, *et al.*, "Fast minimization of fixed polarity Reed-Muller expressions," *IEEE Access*, vol. 7, pp. 24843–24851, 2019.

[16] M. H. A. Khan, "An ASIC architecture for generating optimum mixed polarity Reed-Muller expression," *Engineering Letters*, vol. 13, no. 3, pp. 236–243, 2006.

[17] H. Z. Yu, P. J. Wang, D. S. Wang, *et al.*, "Discrete ternary particle swarm optimization for area optimization of MPRM circuits," *Journal of Semiconductors*, vol. 34, no. 2, article no. 025011, 2013.

[18] A. Das, Y. C. Hareesh, and S. N. Pradhan, "NSGA-II based thermal-aware mixed polarity dual Reed-Muller network synthesis using parallel tabular technique," *Journal of Circuits, Systems and Computers*, vol. 29, no. 15, article no. 020008,

2020.

[19] R. Drechsler, M. Theobald, and B. Becker, "Fast OFDD-based minimization of fixed polarity Reed-Muller expressions," *IEEE Transactions on Computers*, vol. 45, no. 11, pp. 1294–1299, 1996.

[20] R. Drechsler, B. Becker, and N. Drechsler, "Genetic algorithm for minimisation of fixed polarity Reed-Muller expressions," *IEE Proceedings-Computers and Digital Techniques*, vol. 147, no. 5, pp. 349–353, 2000.

[21] U. Kalay, D. V. Hall, and M. A. Perkowski, "A minimal universal test set for self-test of EXOR-sum-of-products circuits," *IEEE Transactions on Computers*, vol. 49, no. 3, pp. 267–276, 2000.

[22] B. A. Al Jassani, N. Urquhart, and A. E. A. Almaini, "Manipulation and optimisation techniques for Boolean logic," *IET Computers & Digital Techniques*, vol. 4, no. 3, pp. 227–239, 2010.

[23] E. Zaitseva, V. Levashenko, I. Lukyanchuk, *et al.*, "Application of generalized Reed-Muller expression for development of non-binary circuits," *Electronics*, vol. 9, no. 1, article no. 12, 2020.

[24] Y. H. Zhou, Z. X. He, C. Chen, *et al.*, "An efficient power optimization approach for fixed polarity Reed-Muller logic circuits based on metaheuristic optimization algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5380–5393, 2022.

[25] C. F. Liu, J. F. Wang, and Y. J. Y. T. Leung, "Integrated bacteria foraging algorithm for cellular manufacturing in supply chain considering facility transfer and production planning," *Applied Soft Computing*, vol. 62, pp. 602–618, 2018.

[26] H. Y. Cai, L. Q. Ran, L. D. Zou, *et al.*, "Heuristic hybrid bacterial foraging algorithm for the pose detection of backlight units," *Automatic Control and Computer Sciences*, vol. 54, no. 3, pp. 229–237, 2020.

[27] S. Das, S. Dasgupta, A. Biswas, *et al.*, "On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 3, pp. 670–679, 2009.

[28] F. L. Ye, C. Y. Lee, Z. J. Lee, *et al.*, "Incorporating particle swarm optimization into improved bacterial foraging optimization algorithm applied to classify imbalanced data," *Symmetry*, vol. 12, no. 2, article no. 229, 2020.

[29] C. Zhang, Y. W. Yu, Y. F. Wang, *et al.*, "Takagi-Sugeno fuzzy neural network hysteresis modeling for magnetic shape memory alloy actuator based on modified bacteria foraging algorithm," *International Journal of Fuzzy Systems*, vol. 22, no. 4, pp. 1314–1329, 2020.

[30] Y. Liu, L. W. Tian, and L. N. Fan, "The hybrid bacterial foraging algorithm based on many-objective optimizer," *Saudi Journal of Biological Sciences*, vol. 27, no. 12, pp. 3743–3752, 2020.

[31] J. J. Hu, N. F. Jin, P. Li, *et al.*, "An improved bacterial foraging algorithm for multi-modal problems," *Journal of Physics:Conference Series*, vol. 1631, article no. 012069, 2020.

[32] W. Shao and Y. J. Zuo, "Computing the halfspace depth with multiple try algorithm and simulated annealing algorithm," *Computational Statistics*, vol. 35, no. 1, pp. 203–226, 2020.

[33] T. Wang, Y. L. Zhang, N. N. Xiong, *et al.*, "An effective edge-intelligent service placement technology for 5G-and-beyond industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4148–4157, 2022.

[34] Z. Shen, Q. H. Yang, and H. Jiang, "Multichannel neighbor discovery in Bluetooth low energy networks: Modeling and performance analysis," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2262–2280, 2023.

[35] F. S. Makri, Z. M. Psillakis, and A. N. Arapis, "On the concentration of runs of ones of length exceeding a threshold in a Markov chain," *Journal of Applied Statistics*, vol. 46, no. 1, pp. 85–100, 2019.

[36] M. Tripathi, L. K. Singh, S. Singh, *et al.*, "A comparative study on reliability analysis methods for safety critical systems using petri-nets and dynamic flowgraph methodology: A case study of nuclear power plant," *IEEE Transactions on Reliability*, vol. 71, no. 2, pp. 564–578, 2022.

[37] I. A. W. Abdul Razak, I. Z. Abidin, Y. K. Siah, *et al.*, "An hour ahead electricity price forecasting with least square support vector machine and bacterial foraging optimization algorithm," *The Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 2, pp. 748–755, 2018.

[38] M. A. Awadallah, "Variations of the bacterial foraging algorithm for the extraction of PV module parameters from nameplate data," *Energy Conversion and Management*, vol. 113, pp. 312–320, 2016.

[39] D. M. Xiao, Y. Wang, K. F. He, *et al.*, "A harmonic suppression method of variable frequency drive based on bacterial foraging algorithm," *International Journal of Computing Science and Mathematics*, vol. 7, no. 1, pp. 76–86, 2016.

[40] B. Subudhi and R. Pradhan, "Bacterial foraging optimization approach to parameter extraction of a photovoltaic module," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, pp. 381–389, 2018.

[41] J. Yi, D. Huang, S. Y. Fu, *et al.*, "Multi-objective bacterial foraging optimization algorithm based on parallel cell entropy for aluminum electrolysis production process," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2488–2500, 2016.

[42] M. A. Sahib, A. R. Abdulnabi, and M. A. Mohammed, "Improving bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution," *Alexandria Engineering Journal*, vol. 57, no. 4, pp. 3341–3349, 2018.

[43] P. Sindhuja and P. Ramamoorthy, "An improved fuzzy enabled optimal multipath routing for wireless sensor network," *Cluster Computing*, vol. 22, no. 5, pp. 11689–11697, 2019.

[44] D. M. Zambrano Zambrano, D. Vélez, Y. Daza, *et al.*, "Parametric analysis of BFOA for minimization problems using a benchmark function," *Enfoque UTE*, vol. 10, no. 3, pp. 67–80, 2019.

[45] S. Yang, "Logic synthesis and optimization benchmarks user guide version 3.0: Technical report," Microelectronics Center of North Carolina, http://www.lsi-cad.com/RM/RM2021.html, 1991.

Yuhao ZHOU is a Ph.D. candidate at the School of Software Engineering, Tongji University, Shanghai, China. He has authored some academic articles which have been published in refereed international journals, such as *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *ACM Transactions on Design Automation of Electronic Systems*, *Applied Soft Computing*, etc. His research interests include Reed-Muller circuits area optimization based on XNOR/OR, power optimization, low power integrated circuits (IC) design, computer aided design and intelligent optimization algorithm.
(Email: zhouyuhao@tongji.edu.cn)

Zhenxue HE received the Ph.D. degree in computer architecture from Beihang University, Beijing, China, in 2018. He is currently a full Associate Professor with Hebei Agricultural University, Baoding, China. He has authored or coauthored more than 40 articles in peer-reviewed journals and proceedings, such as *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on Services Computing*, *IEEE Sensors Journal*, and *Journal of Computer Science and Technology*. His research interests include low power IC design and optimization, multiple-valued logic circuits, combinatorial optimization, and intelligent algorithm.
(Email: hezhenxue@buaa.edu.cn)

Jianhui JIANG received the B.E., M.E., and Ph.D. degrees in traffic information engineering and control from Shanghai Tiedao University (in 2000, it was merged to Tongji University), Shanghai, China, in 1985, 1988, and 1999, respectively. During 1994–2000, he was an Associate Professor in computer science and technology at Shanghai Tiedao University. Since 2000, he has been a full Professor in computer science and technology at Tongji University, Shanghai, China. During 2007–2011, he was the Chair of the Department of Computer Science and Technology, Tongji University. Since 2011, he is the Associate Dean of the School of Software Engineering, Tongji University. He is the Vice Director of Technical Committee on Fault-tolerant Computing, CCF. He has served on several program committees of national or international symposiums or workshops. He has co-authored two books and published more than 200 technical papers. His current research interests include dependable systems and networks, software reliability engineering, and VLSI/SoC testing and fault-tolerance. He is a Senior Member of CCF.
(Email: jhjiang@tongji.edu.cn)

Xiaojun ZHAO received the B.S. degree from Sichuan Normal University, Chengdu, China, in 2013, and received the M.S. degree from Xidian University, Xi'an, China, in 2016. She is currently a Lecturer at Hebei Agricultural University. Her research interests include electronics design automation, intelligent algorithm, signal detection, and data processing.
(Email: jaja028@126.com)

Fan ZHANG received the Ph.D. degree in computer science from Hebei Agricultural University, Baoding, China, in 2013. She is currently a full Associate Professor with Hebei Agricultural University. She has authored or coauthored more than 20 articles in peer-reviewed journals and proceedings. Her research interests include low power IC design and optimization and agricultural informatization.
(Email: ellenzhang0911@126.com)

Limin XIAO received the B.S. degree in computer science (major) and physics (minor) from the Department of Computer Science, Tsinghua University, Beijing, China, in 1993, and the M.S. and Ph.D. degrees in computer science from the Institute of Computer Science, Chinese Academy of Sciences, Beijing, China, in 1996 and 1998, respectively. He is currently a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China. He is a Senior Member of the Chinese Computer Society, the Director of the Institute of Computer System Architecture. His main research areas include computer architecture, computer system software, high-performance computing, virtualization, and cloud computing. He is a Senior Member of CCF and a Senior member of IEEE.
(Email: 930111386@qq.com)

Xiang WANG received the Ph.D. degree in engineering from the School of Information Technology, Peking University, Beijing, China, in 2004. He is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. He has served on the Editorial Boards of several international journals, such as *IEEE Transactions on Very Large Scale Integration Systems*, *IEEE Transactions on Dependable and Secure Computing*, *Science China*, *Chinese Physics Letters*, *Journal of Semiconductor*, and *Journal of Electronics*. He is currently a TPC Member of the IEEE Conference Organizing Committee, Vice Chairman of the Conference, and Chairman of the Conference. His main research areas include very large-scale integration, micro-nano systems, genetic circuits, and aerospace information networks. He is a Senior Member of IEEE.
(Email: wxiang@buaa.edu.cn)