

Received January 13, 2019, accepted April 10, 2019, date of publication April 15, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2911355

A Polarity Optimization Algorithm Taking Into Account Polarity Conversion Sequence

ZHENXUE HE¹, JIA LIU¹, LIMIN XIAO², ZHISHENG HUO², AND XIANG WANG³

¹College of Information Science and Technology, Hebei Agricultural University, Baoding 071001, China

²School of Computer Science and Engineering, Beihang University, Beijing 100191, China

³School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Corresponding author: Zhenxue He (hezhenxue@buaa.edu.cn)

This work was supported in part by the Introducing Talent Research Project of Hebei Agricultural University under Grant YJ201829, in part by the National Natural Science Foundation of China under Grant 61772053, Grant 60973106, Grant 61232009, Grant 81571142, and Grant 31801782, in part by the China Postdoctoral Science Foundation under Grant 2018M641154, in part by the Scientific Science and Technology Research Projects of Universities in Hebei under Grant BJ2018012, in part by the Project of Hebei Natural Science Foundation under Grant G2018204093, and in part by the Baoding Science and Technology Support Plan under Grant 15ZG050.

ABSTRACT The polarity conversion sequence directly determines polarity conversion efficiency and then affects polarity optimization efficiency. However, few studies have focused on the polarity conversion sequence problem of Reed-Muller (RM) circuits. In this paper, we propose a continuous Hopfield neural network (CHNN)-based polarity conversion algorithm (CHNNPCA) for Mixed Polarity RM (MPRM) circuits, which uses the CHNN to solve the best polarity conversion sequence of polarity set waiting for evaluation before converting the polarity set. Moreover, based on the CHNNPCA, a polarity optimization algorithm (POA) is proposed to improve the polarity optimization efficiency of MPRM circuits. The experimental results on MCNC benchmark circuits show that for the large-scale polarity set, the CHNNPCA is superior to the mixed polarity conversion algorithm based on the tabular technique in terms of polarity conversion efficiency. Furthermore, compared to the traditional polarity optimization algorithm neglecting polarity conversion sequence, the POA has a considerable advantage in improving polarity optimization efficiency, especially for large-scale circuits. The POA can be extended to improve the polarity optimization efficiency of fixed polarity RM circuits.

INDEX TERMS Polarity conversion, polarity optimization, Reed-Muller circuits, continuous hopfield neural network.

I. INTRODUCTION

Digital logic circuits can be expressed in either AND/OR/NOT based Boolean logic or AND/XOR based Reed-Muller (RM) logic. Lots of research has shown that compared to circuits implemented using Boolean logic, circuits implemented by RM logic have significant advantage in terms of area, power, speed, and testability [1]–[3]. RM logic has gained increasing popularity, particularly as look-up-table based field programmable gate array technique has become increasingly available and the relative cost of XOR gates has become a non-critical factor restricting the use of RM design approaches [4]–[6].

Fixed Polarity RM (FPRM) expressions and Mixed Polarity RM (MPRM) expressions are the canonical representations of RM logic. The input variables of FPRM expression

only appear in complemented or uncomplemented form, but not both. The MPRM expressions are more compact than FPRM expressions because there are no restrictions on the polarity of input variables [7], [8]. Furthermore, for some classes of practical functions, MPRM expressions require fewer products than Sum-of-Products (SOP) expressions [9], [10]. The polarity optimization of MPRM circuits has been a hot research topic in the field of logic synthesis. However, the optimization of MPRM circuits is general is more complicated and difficult than the optimization of SOP expressions due to their tremendous polarity optimization space. Since the Genetic Algorithm (GA) has the advantage of simplicity, robustness and parallel in essence, it has been widely used in polarity optimization of RM circuits [11]–[14].

Recently, many polarity conversion algorithms have been proposed to derive the RM expression of target polarity. The polarity conversion algorithm based on tabular technique

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

is widely used because it puts no limits on the number of input variables and is easy to implement [15]–[17]. The mixed polarity conversion algorithm [14] indicates that when the MPRM expression of know polarity is converted to the MPRM expression of target polarity, the less different ternary bits between adjacent polarities, the fewer polarity conversion operations are required, the faster polarity conversion efficiency. Therefore, the polarity conversion sequence problem is Traveling Salesman Problem (TSP) [18]. Polarities are equivalent to the cities, and the different ternary bits between polarities are equivalent to the distances between cities. The best polarity conversion sequence is one with the minimum value of the sum of different ternary bits between adjacent polarities. Therefore, it will waste the polarity conversion time and reduce the polarity conversion efficiency to convert the polarity set in polarity random arrangement sequence. However, there are few studies on polarity optimization of MPRM circuits, which consider the polarity conversion sequence of the polarity set.

In this paper, we propose a Polarity Optimization algorithm (POA) of MPRM circuits, which considers the polarity conversion sequence. Compared with the existing polarity optimization algorithms of MPRM circuits, our main contributions are as follows.

1) We establish a mathematical model to describe the polarity conversion sequence problem by using graph theory. The point set denotes polarities, and the edge set denotes the different ternary bits between two polarities.

2) We propose a Continuous Hopfield Neural Network (CHNN) based Polarity Conversion Algorithm (CHNNPCA) for MPRM circuits, which uses CHNN to solve the best polarity conversion sequence of polarity set waiting for evaluation before converting the polarity set. The CHNNPCA can be applied to complete polarity set as well as incomplete polarity set, and can be extended to improve the polarity conversion efficiency of FPRM circuits. To the best of our knowledge, we are the first to use CHNN for the optimization of RM circuits.

3) We propose a polarity optimization algorithm called POA to improve the polarity optimization efficiency of MPRM circuits. We compare the POA with the traditional polarity optimization algorithm neglecting polarity conversion sequence. Experimental results demonstrate the effectiveness and superiority of POA. The POA can be extended to improve the polarity optimization efficiency of FPRM circuits.

The remainder of this paper is structured as follows. In Section II, we introduce the MPRM expression. The mathematical model for polarity conversion sequence problem is described in Section III. Using CHNN to solve polarity conversion sequence problem is described in Section IV. Section V presents a CHNN based polarity conversion algorithm. Section VI presents a polarity optimization algorithm for MPRM circuits. The experimental results are described in Section VII. Our conclusions are presented in Section VIII.

II. MPRM EXPRESSION

Any n -variable Boolean function can be represented canonically in SOP form as

$$f(x_{n-1}, x_{n-2}, \dots, x_0) = \sum_{i=0}^{2^n-1} a_i m_i \quad (1)$$

where Σ is an OR operator and m_i are the minterms, which can be expressed as

$$m_i = \dot{x}_{n-1} \dot{x}_{n-2} \dots \dot{x}_0, \dot{x}_j = \begin{cases} \bar{x}_j, & i_j = 0 \\ x_j, & i_j = 1 \end{cases} \quad (2)$$

where $j = n - 1, n - 2, \dots, 0$, a_i are the coefficient of the minterms and $a_i = 1$ or 0 , which corresponds to the presence or absence of minterms, respectively. The OR can be replaced by an XOR if all the variables are present in every term of Eq. (1). Alternatively, the function can be expressed by an MPRM expression as follows:

$$f^p(x_{n-1}, x_{n-2}, \dots, x_0) = \oplus \sum_{i=0}^{2^n-1} b_i \pi_i \quad (3)$$

where $\oplus \Sigma$ denotes modulo-2 addition and π_i represents the product term of the MPRM expression. $b_i \in \{0, 1\}$ represents whether or not π_i appears in the MPRM expression. $p = (p_{n-1} p_{n-2} \dots p_0)$ is the polarity and $i = (i_{n-1} i_{n-2} \dots i_0)$ is the subscript. The relationship between \dot{x}_j , p_i , and i_j can be expressed as follows:

TABLE 1. The value of \dot{x}_j .

$p_i = 0$	$p_i = 1$	$p_i = 2$
$\dot{x}_j = \begin{cases} 1, i_j=0 \\ x_j, i_j=1 \end{cases}$	$\dot{x}_j = \begin{cases} 1, i_j=0 \\ \bar{x}_j, i_j=1 \end{cases}$	$\dot{x}_j = \begin{cases} \bar{x}_j, i_j=0 \\ x_j, i_j=1 \end{cases}$

In an MPRM expression, each variable can appear as true, complemented, or mixed. The polarity of MPRM expression can be represented by replacing each variable with 0, 1, or 2, depending on whether the variable is true, complemented, or mixed. When a variable is true (complemented), it can be replaced with 0 (1). When a variable is present in both true and complemented forms, it can be replaced with 2. The polarity directly determines the simplicity or complexity of the expression, meaning it influences circuit performance in terms of power and area. Consequently, polarity optimization of MPRM circuits, which aims to find the polarity under which one circuit performance is the best within a particular polarity space, is a combination optimization problem. The polarity can be expressed as decimal equivalent of the resulting ternary number. The above description is illustrated in Example 1.

Example 1: Given a 3-variables MPRM expression $f^5(x_2, x_1, x_0) = x_2 \bar{x}_1 \oplus x_2 \bar{x}_1 \bar{x}_0 \oplus x_2 x_0$ with polarity 5. Since the decimal polarity 5 is equivalent to the ternary $(012)_3$, x_2 appears in true form, x_1 appears in complemented form, and x_0 appears in mixed form within this MPRM expression.

III. MATHEMATICAL MODEL TO DESCRIBE THE POLARITY CONVERSION SEQUENCE PROBLEM

For a polarity set with n polarities, the number of polarity conversion sequences is $n!$. Therefore, it is a combinatorial optimization problem to choose the best polarity conversion sequence, which corresponds to the minimum value of the sum of different ternary bits between adjacent polarities, from the $n!$ polarity conversion sequences. More precisely, it belongs to classical TSP.

We use the graph theory to describe the polarity conversion sequence problem. Let $G = (P, E)$ be a graph, where point set P denotes polarities, and edge set E denotes the different ternary bits between two polarities. Therefore, the polarity conversion sequence problem can be described as follows:

$$\text{Min} \sum_{r \neq s} b_{rs} c_{rs} \quad (4)$$

$$\sum_{s \neq r} c_{rs} = 1 \quad r \in P \quad (5)$$

$$\sum_{r \neq s} c_{rs} = 1 \quad s \in P \quad (6)$$

$$\sum_{r, s \in V} c_{rs} \leq |V| - 1 \quad V \in P \quad (7)$$

$$c_{rs} \in \{0, 1\} \quad r, s \in P \quad (8)$$

where b_{rs} denotes the different ternary bits between polarity r and polarity s , $c_{rs} \in \{0, 1\}$ denotes the presence or absence of the edge (r, s) in the polarity conversion sequence, and $|V|$ denotes the number of polarities in set V . Equation (4) denotes the object function of polarity conversion sequence problem, which aims at minimizing the sum of different ternary bets between adjacent polarities. Equations (5) and (6) are restrictions to guarantee each polarity in polarity conversion sequence is converted only once. Equations (7) and (8) are restrictions to guarantee that there is no sub-sequence. Therefore, the solutions satisfying the above restrictions compose a polarity conversion sequence, which visits all the polarities and converts each polarity only once.

IV. USING CHNN TO SOLVE POLARITY CONVERSION SEQUENCE PROBLEM

Numerous studies show that compared to traditionally combined optimization methods, CHNN has stronger computing power and storage capacity, especially in solving the optimization problem such as TSP [19]–[21]. Therefore, we use the CHNN to solve the polarity conversion sequence problem. To use CHNN to solve polarity conversion sequence problem, it is necessary to transform the restrictions and objective function into energy function, and use the output of CHNN to represent the efficient solution. As the network status changed, the efficient solution of polarity conversion sequence problem can be obtained when the energy function reaches a minimum.

Using CHNN to solve the best polarity conversion sequence of a polarity set with n polarities, the $n * n$ nerve cells are need to structure the $n * n$ permute matrix, where row

x denotes the polarity number, column j denotes the position of polarity in polarity conversion sequence. If v_{xj} is equal to 1, it represents that the polarity x appears at the j -th position of polarity conversion sequence.

Example 2: Given a polarity set with 5 polarities, which are represented by A, B, C, D and E, respectively. A conversion sequence $B \rightarrow D \rightarrow E \rightarrow A \rightarrow C$ is randomly chosen, the sum of different ternary bits between adjacent polarities can be expressed as follows:

$$S = d_{BD} + d_{DE} + d_{EA} + d_{AC} \quad (9)$$

where d_{BD} denotes the different ternary bits between polarity B and polarity D, d_{DE} denotes the different ternary bits between polarity D and polarity E, d_{EA} denotes the different ternary bits between polarity E and polarity A, d_{AC} denotes the different ternary bits between polarity A and polarity C.

TABLE 2. Permute matrix.

polarity	order				
	1	2	3	4	5
A	0	0	0	1	0
B	1	0	0	0	0
C	0	0	0	0	1
D	0	1	0	0	0
E	0	0	1	0	0

The polarity conversion sequence problem can be mapped into neural network by using a permute matrix shown in Table 2. The restrictions and the optimal condition of polarity conversion sequence problem are as follows:

Restriction (1): a polarity is converted only once, namely, each row only has one ‘1’;

Restriction (2): only one polarity is converted at a time, namely, each column only has one ‘1’;

Restriction (3): there are n polarities, namely, the sum of elements in permute matrix are equal to n ;

Optimal condition: solving the minimum value of the sum of different ternary bits between adjacent polarities.

The energy function is structured according to the above restrictions and optimal condition, as below:

If the row x satisfies restriction (1), the sum of multiplication of any two elements in x -th row is equal to 0, namely:

$$\sum_{x=1}^{n-1} \sum_{j=i+1}^n v_{xi} v_{xj} = 0 \quad (10)$$

Obviously, the sum of multiplication of any two elements of all the rows is equal to 0, namely:

$$\sum_{x=1}^n \sum_{i=1}^{n-1} \sum_{j=i+1}^n v_{xi} v_{xj} = 0 \quad (11)$$

Similarly, the following equation can be obtained according to restriction (2):

$$\sum_{i=1}^n \sum_{x=1}^{n-1} \sum_{y=x+1}^n v_{xi}v_{yi} = 0 \quad (12)$$

From the restriction (3), it is concluded that the sum of all the elements in permute matrix is equal to n , namely:

$$\sum_{x=1}^n \sum_{i=1}^n v_{xi} - n = 0 \quad (13)$$

The objective function of polarity conversion sequence can be obtained according to the optimal condition, namely:

$$J(v) = \min \left[\frac{1}{2} \sum_{x=1}^n \sum_{y=x+1}^n \sum_{i=1}^n d_{xy}v_{xi}(v_{y,i+1} + v_{y,i-1}) \right] \quad (14)$$

where d_{xy} denotes the different ternary bits between polarity x and polarity y . If polarity x and polarity y are next to each other, $V_{y,i+1}$ and $V_{y,i-1}$, one is 0 and the other is 1; if polarity x and polarity y are not next to each other, then both $V_{y,i+1}$ and $V_{y,i-1}$ are 0. Therefore, the sum of different ternary bits between adjacent polarities can be calculated according to the objective function.

Therefore, the polarity conversion sequence problem can be represented as the following optimization problem:

$$\min J(v) = \frac{1}{2} \sum_{x=1}^n \sum_{y=x+1}^n \sum_{i=1}^n d_{xy}v_{xi}(v_{y,i+1} + v_{y,i-1}) \quad (15)$$

$$s.t. J_1(v) = \sum_{x=1}^n \sum_{i=1}^{n-1} \sum_{j=i+1}^n v_{xi}v_{xj} = 0 \quad (16)$$

$$J_2(v) = \sum_{i=1}^n \sum_{x=1}^{n-1} \sum_{y=x+1}^n v_{xi}v_{yi} = 0 \quad (17)$$

$$J_3(v) = \left(\sum_{x=1}^n \sum_{i=1}^n v_{xi} - n \right)^2 = 0 \quad (18)$$

The energy function of CHNN is structured by putting the restrictions and optimal condition together, as below:

$$\begin{aligned} E = & \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^{n-1} \sum_{j=i+1}^n v_{xi}v_{xj} + \frac{B}{2} \sum_{i=1}^n \sum_{x=1}^{n-1} \sum_{y=x+1}^n v_{xi}v_{yi} \\ & + \frac{C}{2} \left(\sum_{x=1}^n \sum_{i=1}^n v_{xi} - n \right)^2 \\ & + \frac{D}{2} \sum_{x=1}^n \sum_{y=x+1}^n \sum_{i=1}^n d_{xy}v_{xi}(v_{y,i+1} + v_{y,i-1}) \end{aligned} \quad (19)$$

The link weight $w_{xi,yj}$ between nerve cells and output threshold value I_{xi} of nerve cells can be obtained by comparing equation (19) with standard energy function. As the network status changed, the polarity conversion sequence can be obtained according to the permute matrix consisting of network status $v_{i,j}$ when the energy function reaches a minimum.

V. CHNN BASED POLARITY CONVERSION ALGORITHM

Research indicates that the CHNN performs well in solving small-scale TSP with 10 cities, while its performance is poor when deals with large-scale TSP with 30 cities [22]. Literature [23] analyzed the reason why CHNN is difficult to obtain efficient solution from hyperspace perspective, and modified the energy function of CHNN, as below:

$$\begin{aligned} E = & \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^n \sum_{j=1, j \neq i}^n v_{xi}v_{xj} + \frac{B}{2} \sum_{x=1}^n \sum_{y=1, y \neq x}^n \sum_{i=1}^n v_{xi}v_{xi} \\ & + A_1 \left(\sum_{x=1}^n \sum_{i=1}^n v_{xi}^2 + \frac{C}{2} \sum_{x=1}^n \sum_{i=1}^n v_{xi} - n \right)^2 \\ & - \left(\frac{A_n - A + A_1}{n} \right) \sum_{x=1}^n \sum_{i=1}^n \sum_{y=1}^n \sum_{j=1}^n v_{xi}v_{yj} \\ & + \frac{D}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{j=1}^n d_{xy}v_{xj}(V_{y,i-1} + V_{y,i+1}) \end{aligned} \quad (20)$$

Accordingly, the link weight is modified, namely:

$$\begin{aligned} T_{xi,yj} = & -A\delta_{xy}(1 - \delta_{ij}) - B\delta_{ij}(1 - \delta_{xy}) - 2A_1\delta_{xy}\delta_{ij} - C \\ & + 2(AN - A + A_1)/N^2 - Dd_{xy}(\delta_{j,i+1} + \delta_{j,i-1}) \end{aligned} \quad (21)$$

From the Equations (20) and (21), it is concluded that the literature [23] can obtain efficient solution at the price of surprisingly complex energy function and link weight, which will extend the convergence time. Traditionally CHNN is sensitive to parameter and its convergence speed is slow. To eliminate the above flaw of CHNN, literature [24] proposed an improved algorithm of CHNN solving TSP and theoretically proved the effectiveness of the improved algorithm. The energy function of literature [24] is as follows:

$$\begin{aligned} E = & \frac{A}{2} \sum_{x=1}^N \left(\sum_{i=1}^N v_{xi} - 1 \right)^2 + \frac{A}{2} \sum_{i=1}^N \left(\sum_{x=1}^N v_{xi} - 1 \right)^2 \\ & + \frac{D}{2} \sum_{x=1}^N \sum_{y=1}^N \sum_{i=1}^N d_{xy}v_{xi}v_{y,i+1} \end{aligned} \quad (22)$$

And its dynamical equation can be expressed as follows:

$$\begin{aligned} \frac{dU_{xi}}{dt} = & -\frac{\partial E}{\partial V_{xi}} = -A \left(\sum_{i=1}^N v_{xi} - 1 \right) - A \left(\sum_{y=1}^N v_{yi} - 1 \right) \\ & - D \sum_{y=1}^N d_{xy}v_{y,i+1} \end{aligned} \quad (23)$$

In this section, based on the CHNN model proposed in [24], we propose a polarity conversion algorithm for MPRM circuits, called CHNNPCA, to improve polarity conversion efficiency. The difference between CHNNPCA and mixed polarity conversion algorithm based on tabular technique lies in the following two aspects: firstly, CHNNPCA uses CHNN proposed in [24] to solve the best polarity conversion sequence before converting polarity set; secondly,

each polarity is converted according to the obtained best polarity conversion sequence. Specifically, the steps to solve the best polarity conversion sequence by using CHNN are as follows (the parameter settings refer to literature [24]):

- (1) The variables and weights are initialized, namely, $t = 0, A = 1.5, D = 1.0$, where the A and D are weights;
- (2) The different ternary bits between n polarities in polarity set waiting for evaluation are calculated;
- (3) The input matrix $U_{xi}(t)$ of CHNN is initialized, namely, $U_{xi}(t) = U'_0 + \delta_{xi}$, where $x, i = 1, 2, \dots, n$ and $U'_0 = \frac{1}{2}U_0 \ln(n - 1), U_0 = 0.02, \delta_{xi}$ is a random value in $(-1, +1)$;
- (4) The output matrix $V_{xi}(t)$ of CHNN is calculated by using the Sigmod activation function, namely, $V_{xi}(t) = \frac{1}{2}(1 + \tanh(\frac{U_{xi}(t)}{U_0}))$;
- (5) If the maximum number of iteration is reached, the step (11) is performed. Otherwise, go to step (6);
- (6) The $\frac{dU_{xi}}{dt}$ is calculated according to the neural network dynamical equation;
- (7) The input matrix $U_{xi}(t + 1)$ of CHNN at the next moment is calculated, namely, $U_{xi}(t + 1) = U_{xi}(t) + \frac{dU_{xi}}{dt} \Delta T$, where $\Delta T = 0.01$;
- (8) The output matrix $V_{xi}(t + 1)$ of CHNN at the next moment is calculated, namely, $V_{xi}(t + 1) = \frac{1}{2}(1 + \tanh(\frac{U_{xi}(t+1)}{U_0}))$;
- (9) The energy function E is calculated;
- (10) The legality of polarity conversion sequence is checked, if the polarity conversion sequence is legal, the sum of different ternary bits between adjacent polarities corresponding to the polarity conversion sequence is stored and go to step (5); Otherwise, go to step (5) directly;
- (11) The best polarity conversion sequence corresponding to minimum value of the sum of different ternary bits between adjacent polarities is output according to the sum of different ternary bits between adjacent polarities;

It should be noted that we do not use $\frac{dE}{dt} = 0$ as the ending condition due to the fact that the point that satisfies $\frac{dE}{dt} = 0$ could be an inflection point, instead of a minimum value point. Moreover, if the point that satisfies $\frac{dE}{dt} = 0$ is minimum value point, keep up iteration will help algorithm to escape the local optimum, and then improve the global optimization ability.

VI. POLARITY OPTIMIZATION ALGORITHM TAKING INTO ACCOUNT POLARITY CONVERSION SEQUENCE

In this section, based on the proposed CHNNPCA and GA, we propose a polarity optimization algorithm, called POA, to improve polarity optimization efficiency of MPRM circuits. An overview of POA is depicted in Fig.1. The POA involves chromosome encoding scheme, fitness function, roulette wheel selection, crossover and mutation operation. We will introduce them in detail in the following sections.

A. CHROMOSOME ENCODING SCHEME

Since the polarity of MPRM expression can be represented by replacing each variable with 0, 1, or 2 depending on whether

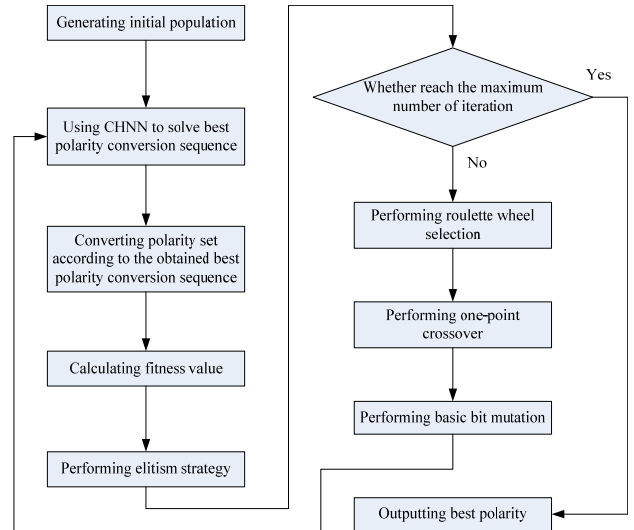


FIGURE 1. Overview of POA.

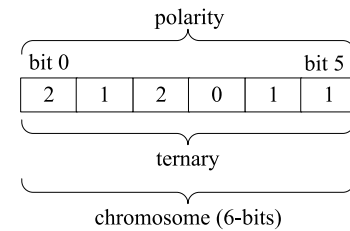


FIGURE 2. Chromosome encoding for example 3.

the variable is true, complemented, or mixed, we encode the polarity of MPRM expression as ternary form. The above description is illustrated in Example 3.

Example 3: Given a 6-variables MPRM expression, the ternary polarity “212011” can be encoded as 6-bits chromosome as shown in Fig.2.

B. FITNESS FUNCTION

Fitness function is used to evaluate quality of chromosomes. The higher the fitness value, the better the quality of chromosome. Therefore, the fitness function of POA can be defined as follows:

$$fitness(c_i) = 1.0/objective() \quad (24)$$

where $fitness()$ denotes fitness function, c_i denotes the i -th chromosome, $objective()$ denotes objective function corresponding to circuit performance.

C. ROULETTE WHEEL SELECTION

Roulette wheel selection is used to choose the excellent chromosomes to produce offspring population. Given a polarity set with n polarities, the probability that i -th polarity is selected is as follows:

$$p(i) = fitness(i) / \sum_{i=1}^n fitness(i) \quad (25)$$

where $fitness(i)$ denotes the fitness value of i -th polarity, $1 \leq i \leq n$. Furthermore, the cumulative probability of i -th polarity is as follows:

$$c(i) = \sum_{j=1}^i p(j) \quad (26)$$

where $c(i)$ denotes the cumulative probability of i -th polarity, $1 \leq i \leq n, 1 \leq j \leq i$.

D. OPERATION

We use one-point crossover to produce offspring population due to the ternary-encoded form of polarity. Fig.3 gives an illustration of one-point crossover. Suppose that ‘Parent 1’ and ‘Parent 2’ are two parent chromosomes selected for crossover. Firstly, crossover point is randomly generated. Secondly, child chromosomes ‘Child 1’ and ‘Child 2’ are generated by exchanging the section after the crossover point.

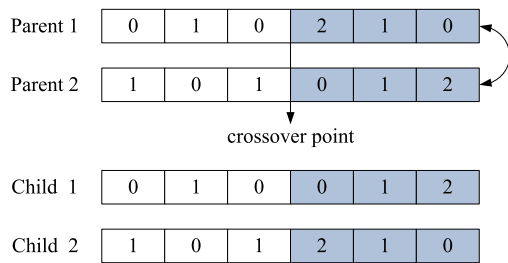


FIGURE 3. Demonstrating one-point crossover.

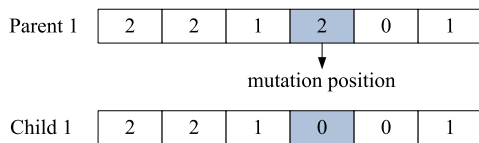


FIGURE 4. Demonstrating basic bit mutation.

We use the simplest basic bit mutation to change one or more gene values, namely, if the original value of selected gene is 0, then we change it to 1; if the original value of selected gene is 1, then we change it to 2; if the original value of selected gene is 2, then we change it to 0. Fig.4 gives an illustration of the basic bit mutation. Suppose that ‘Parent 1’ is parent chromosome selected for mutation. Firstly, mutation position is randomly generated. Secondly, child chromosome ‘Child 1’ is generated by changing the value of gene corresponding to mutation position from 1 to 0.

E. ALGORITHM DESCRIPTION

Based on the above description, the POA is described in Algorithm 1, where ‘ $popsiz$ ’ denotes population size, ‘ gen ’ denotes current number of iteration, and ‘ max_{gen} ’ denotes the maximum number of iteration.

VII. EXPERIMENTAL RESULTS

The POA had been implemented in C language, and the programs were compiled by the GNU C compiler. The results

Algorithm 1 POA

```

Input: Boolean circuit
Output: The best polarity
1: Read the Boolean circuit;
2: Parameters are initialized;
3: Initial population are generated randomly;
4: for  $gen = 0$  to  $max_{gen}$  do
5:   if  $gen == 0$  then
6:     CHNNPCA is performed;
7:     for  $i = 0$  to  $popsiz$  do
8:       Fitness value of chromosome is calculated;
9:     end for
10:    Elitism strategy is performed;
11:  end
12: if  $gen \neq 0$  then
13:   Roulette wheel selection is performed;
14:   One-point crossover is performed;
15:   Basic bit mutation is performed;
16:   CHNNPCA is performed;
17:   for  $i = 0$  to  $popsiz$  do
18:     Fitness value of chromosome is calculated;
19:   end for
20:   Elitism strategy is performed;
21: end
22: end for
23: The best polarity is output.
    
```

were obtained by using a PC with Intel Core i5 2.40 GHz with 4G RAM under Linux. Moreover, we used the MCNC benchmark circuits as experimental circuits. Firstly, we compared the CHNNPCA with the Mixed Polarity Conversion Algorithm (MPCA) [14] based on tabular technique to demonstrate the effectiveness of CHNNPCA in improving polarity conversion efficiency. Secondly, to verify the effectiveness of POA in improving the polarity optimization efficiency of MPRM circuits, we compared the POA with Traditional Polarity Optimization Algorithm (TPOA) [25] for MPRM circuits, which solve the best polarity by using GA and neglects the polarity conversion sequence.

A. COMPARISON OF CHNNPCA AND MPCA

In this section, we used the CHNNPCA and MPCA to convert three polarity sets to demonstrate the effectiveness of CHNNPCA. The first polarity set is a complete polarity set with 27 polarities of 3-variables circuit, the second polarity set is an incomplete polarity set with 100 polarities of 8-variables test circuit dc2, and the third polarity set is an incomplete polarity set with 600 polarities of 10-variables test circuit ex1010. Furthermore, we ran the CHNNPCA and MPCA 10 times due to the randomness of CHNN, and the maximum number of iterations of CHNN is set to be 2000.

The polarity conversion time of CHNNPCA and MPCA on the first polarity set is shown in Fig.5. From Fig.5 we can see that compared to the MPCA, the CHNNPCA took more

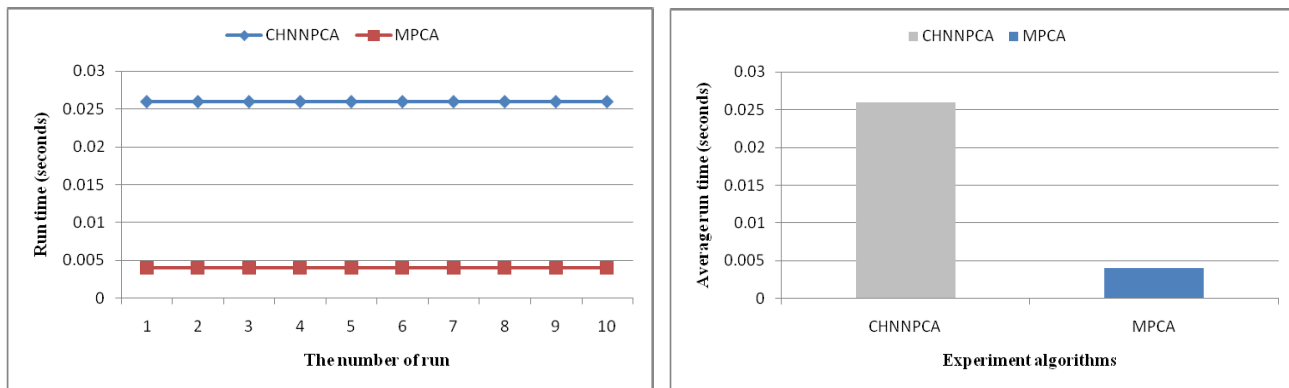


FIGURE 5. Polarity conversion time of CHNNPCA and MPCA on the first polarity set.

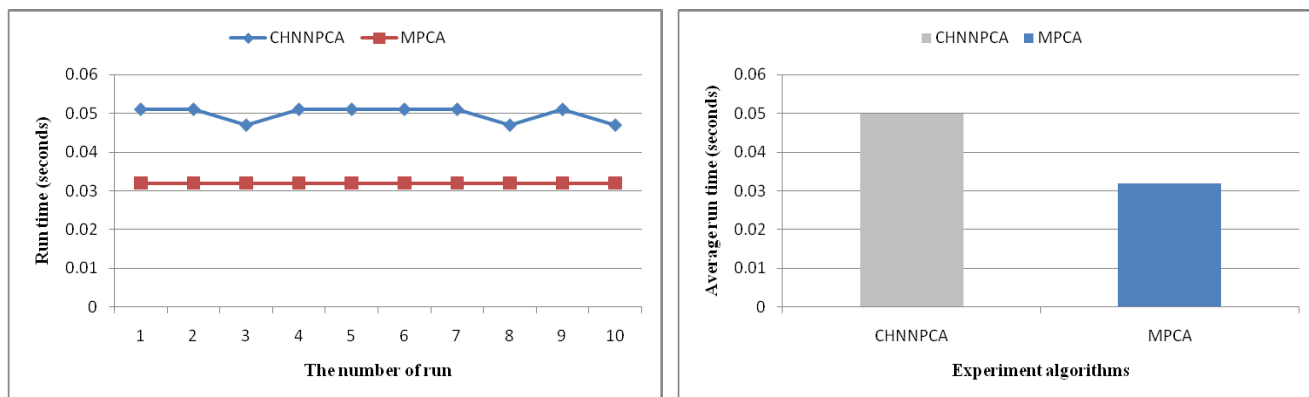


FIGURE 6. Polarity conversion time of CHNNPCA and MPCA on the second polarity set.

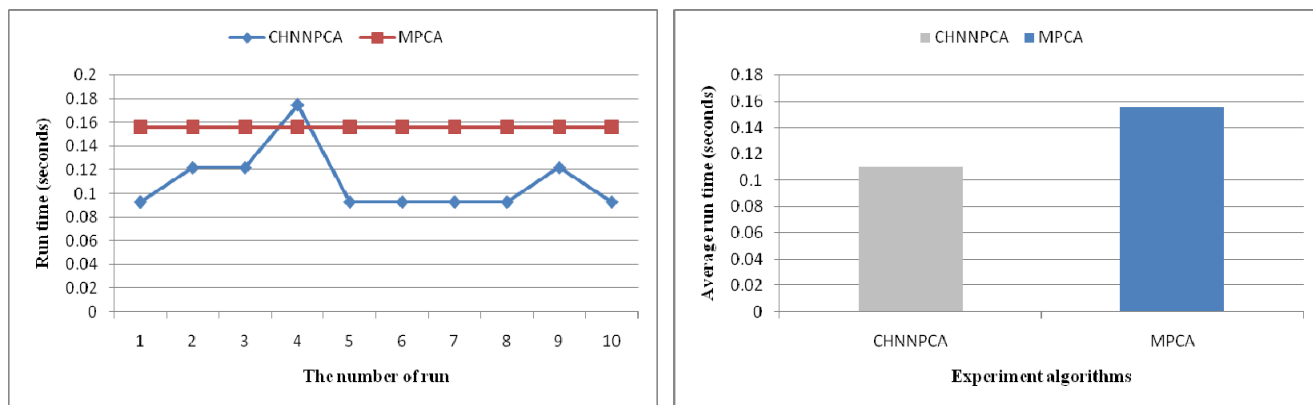


FIGURE 7. Polarity conversion time of CHNNPCA and MPCA on the third polarity set.

polarity conversion time. The main reason is that although the sum of different ternary bits between adjacent polarities of the polarity conversion sequence obtained by CHNN are less than the sum of different ternary bits between adjacent polarities of polarity random arrangement sequence, the time using CHNN to solve best polarity conversion sequence is longer than the time saved considering polarity conversion sequence, thus reducing the polarity conversion efficiency.

The polarity conversion time of CHNNPCA and MPCA on the second polarity set is shown in Fig.6. As shown in Fig.6, the polarity conversion efficiency of CHNNPCA is lower than that of MPCA. There are two possible reasons for this, one possibility is that although the CHNN succeeded in finding the best polarity conversion sequence, the time using CHNN to solve best polarity conversion sequence is longer than the time saved considering polarity conversion sequence;

TABLE 3. Parameters settings for POA and TPOA.

parameter	value
the size of population	80
the maximum number of iteration	80
the probability of crossover	0.8
the probability of mutation	0.05

another possibility is that the CHNN is trapped into local optimization or cannot find the efficient solution, thus wasting time solving the best polarity conversion sequence.

The polarity conversion time of CHNNPCA and MPCA on the third polarity set is shown in Fig. 7. As shown in Fig. 7, for the large-scale polarity set, the CHNNPCA outperforms the MPCA in terms of polarity conversion efficiency. Compared to MPCA, the highest percentage of polarity conversion time saved by CHNNPCA reached 40.38%, and the average percentage of polarity conversion time saved by CHNNPCA reached 29.49%. The main reason is that the time saved considering polarity conversion sequence is greater than the time using CHNN to solve best polarity conversion sequence, thus making the CHNNPCA has higher polarity conversion efficiency.

Moreover, we could also see that for the fourth experiment, the polarity conversion time of CHNNPCA is longer than that of MPCA. This is primarily because the CHNN had not found the efficient solutions or the CHNN only found the lower quality solutions (namely, the sum of different ternary bits between adjacent polarities of the polarity conversion sequence obtained by CHNN are approximately equal to the

sum of different ternary bits between adjacent polarities of polarity random arrangement sequence, thus wasting time solving the best polarity conversion sequence and reducing the polarity conversion efficiency.

B. COMPARISON OF POA AND TPOA

In this section, we set the power minimization as the polarity optimization goal of MPRM circuits, and used the power estimation model proposed in [25] to compute circuit power accurately. Moreover, 24 MCNC benchmark circuits are randomly selected to obtain the persuasive experimental results. Furthermore, we ran the POA and TPOA 10 times due to the randomness of GA. The parameters of two algorithms are summarized in Table 3.

The comparison of POA and TPOA is shown in Table 4, where column 1 shows the circuit name, column 2 shows the number of input variables, column 3 shows the minimum switching activity obtained by POA and TPOA, column 4 shows the best polarity corresponding to minimum switching activity, columns 5 and 6 show the average time (in second) of POA and TPOA, column 7 shows the percentage of time saved by POA compared to TPOA, which is defined as

$$Save\% = ((TPOA_{time} - POA_{time}) / TPOA_{time}) * 100\% \quad (27)$$

From the Table 4, we can see that the test circuits may have one or more best polarities, and the best polarities obtained by POA and TPOA are the same. We could also see that for the circuits with fewer variables, such as squar5, rd84 and 9sym, compared to TPOA, the POA considering polarity conversion sequence not only failed to improve polarity optimization efficiency, but also extended the polarity optimization time.

TABLE 4. Comparison of TPOA and POA.

Name	Inputs	SA	best polarity	Time(s)		
				TPOA	POA	Save(%)
squar5	5	2.49	66, 87	0.49	2.25	-359.18
m1	6	9.05	460, 703, 622	0.98	3.04	-210.20
Z5xp1	7	64.25	57,912	4.78	13.51	-182.63
rd84	8	82.55	2, 274	5.27	15.24	-189.18
m3	8	20.20	1765, 426, 2391	3.30	7.65	-131.82
9sym	9	21.06	1284, 1769, 3829	2.97	6.30	-121.12
ex1010	10	24.79	42681, 59048	2.49	2.53	-1.61
ex1020	10	168.01	2074, 8436, 15895	6.46	5.12	20.74
t4	12	79.46	24387, 35972	6.84	4.38	35.96
newapla	12	105.90	308773	13.55	8.07	40.44
14_4color	14	123.30	2393764	7.97	8.69	-9.03
dk48	15	36.15	5019472, 9031578	4.34	4.32	0.46
b2	16	47.02	26213380	24.83	17.55	29.32
spla	16	56.31	1161821, 5221045	21.14	11.28	46.64
table5	17	100.73	128963012	19.68	10.40	47.15
src1	18	304.18	25905535	60.70	32.51	46.44
in2color	19	146.44	714768549, 902995532	37.22	35.46	4.73
pcl	19	19.26	105341547, 161239925	97.76	42.03	57.01
mark1	20	193.48	26352617	127.27	73.15	42.52
t1	21	75.93	8501474018	80.29	34.66	56.83
mux	21	96.62	3705242062, 3971888614	142.21	53.37	62.47
cm150a	21	83.90	6117835932, 7074138196	136.97	140.98	-2.93
duke2	22	176.82	8918752281, 16772852888	234.06	86.42	63.08
cordic	23	537.48	25341154116	274.41	193.25	29.58

The above results can be explained by the following two possible reasons:

(1) CHNN did not find the efficient solution or only found the general solution (namely, there is little difference between the obtained polarity conversion sequence and polarity random arrangement sequence) in one generation or some generations of POA, thereby wasting the time solving best polarity conversion sequence;

(2) Although the CHNN can obtain the best polarity conversion sequence every time, the time using CHNN to solve best polarity conversion sequence is longer than the time saved considering polarity conversion sequence.

Moreover, for the circuits such as dk48, in2color and cm150a, compared to TPOA, POA has no advantage over the TPOA, because the time using CHNN to solve best polarity conversion sequence is approximately equal to the time saved considering polarity conversion sequence. It is worthwhile to mention that for the large-scale circuits such as pcle, t1, mux and duke2, compared to TPOA, the POA has considerable advantage in improving the polarity optimization efficiency of MPRM circuits. Compared to TPOA, the highest percentage of polarity conversion time saved by POA reached 63.08%.

VIII. CONCLUSION

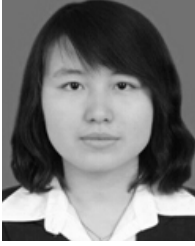
In this paper, we have proposed a polarity optimization algorithm called POA to improve the polarity optimization efficiency of MPRM circuits. The POA uses the proposed CHNNPCA to improve polarity conversion efficiency, and is based on the GA to solve the best polarity. Experimental results show that for the large-scale polarity set, the CHNNPCA is an efficient algorithm in improving the polarity conversion efficiency. Moreover, the comparison with traditional polarity optimization algorithm neglecting polarity conversion sequence shows that the POA has considerable advantage in improving the polarity optimization efficiency of MPRM circuits. Future work will focus on further improving the efficiency of CHNNPCA to convert large-scale polarity set.

REFERENCES

- [1] S. Shirinzadeh, M. Soeken, P.-E. Gaillardon, and R. Drechsler, "Logic synthesis for RRAM-based in-memory computing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 7, pp. 1422–1435, Jul. 2018.
- [2] S. D. Kumar, H. Thapliyal, and A. Mohammad, "FinSAL: FinFET-based secure adiabatic logic for energy-efficient and DPA resistant IoT devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 1, pp. 110–122, Jan. 2017.
- [3] C. Chen, B. Lin, and M. Zhu, "Verification method for area optimization of mixed-polarity Reed-Muller logic circuits," *J. Eng. Sci. Technol. Rev.*, vol. 11, no. 1, pp. 28–34, 2018.
- [4] Z.-X. He et al., "A power and area optimization approach of mixed polarity Reed-Müller expression for incompletely specified Boolean functions," *J. Comput. Sci. Technol.*, vol. 32, no. 2, pp. 297–311, 2017.
- [5] Q. Zhang, P. Wang, J. Hu, and H. Zhang, "Cube-based synthesis of ESOPs for large functions," *Chin. J. Electron.*, vol. 27, no. 3, pp. 527–534, Jan. 2018.
- [6] R. Ueno, N. Homma, T. Aoki, and S. Morioka, "Hierarchical formal verification combining algebraic transformation with PPRM expansion and its application to masked cryptographic processors," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E100-A, no. 7, pp. 1396–1408, 2017.
- [7] X. Wang, M. Li, Z. He, W. Wang, C. Zhou, and Z. Zha, "PAOA: A power and area optimization approach of Reed-Müller logic circuits," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst.*, Aug. 2017, pp. 1394–1397.
- [8] H. Z. Yu, P. J. Wang, H. H. Zhang, and K. Wan, "Optimization of MPRM circuits based on ternary diversity particle swarm optimization," *Acta Electron. Sinica*, vol. 45, no. 7, pp. 1601–1607, 2017.
- [9] D. Debnath and T. Sasao, "Exact minimization of fixed polarity Reed-Müller expressions for incompletely specified functions," in *Proc. Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2000, pp. 247–252.
- [10] C.-C. Tsai and M. Marek-Sadowska, "Boolean functions classification via fixed polarity Reed-Müller forms," *IEEE Trans. Comput.*, vol. C-46, no. 2, pp. 173–186, Feb. 1997.
- [11] D. Bu, J. Jiang, and W. Luo, "Two-phase GA based area and SER trade-off algorithm for MPRM circuits," *J. Comput.-Aided Des. Comput. Graph.*, vol. 29, no. 10, pp. 1924–1934, 2017.
- [12] C. K. Vijayakumari, R. K. James, and P. Mythili, "A GA based simple and efficient technique to design combinational logic circuits using universal logic modules," *J. Circuits, Syst. Comput.*, vol. 25, no. 7, 2016, Art. no. 1650074.
- [13] A. Das and S. N. Pradhan, "Shared Reed-Müller decision diagram based thermal-aware AND-XOR decomposition of logic circuits," *VLSI Des.*, vol. 2016, pp. 1–14, Dec. 2015.
- [14] B. A. Al Jassani, N. Urquhart, and A. E. A. Almaini, "Manipulation and optimisation techniques for Boolean logic," *IET Comput. Digit. Techn.*, vol. 4, no. 3, pp. 227–239, May 2010.
- [15] Y. Wang, L. Wang, and Y. Xia, "A fast Reed-Müller fixed polarity conversion algorithm for large circuits," *J. Comput.-Aid Desig. Comput. Graph.*, vol. 26, no. 11, pp. 2091–2098, 2014.
- [16] M. Yang and J. Lai, "Optimisation of mixed polarity Reed-Müller functions," *J. Softw.*, vol. 8, no. 11, pp. 2770–2774, 2013.
- [17] R. S. Stankovic and B. J. Falkowski, "Spectral interpretation of the fast tabular technique for fixed-polarity Reed-Müller expressions," *Int. J. Electron.*, vol. 87, no. 6, pp. 641–648, 2000.
- [18] Z. He et al., "An efficient and fast polarity optimization approach for mixed polarity Reed-Müller logic circuits," *Frontiers Comput. Sci.*, vol. 11, no. 4, pp. 728–742, 2017.
- [19] L. García, P. M. Talaván, and J. Yáñez, "Improving the Hopfield model performance when applied to the traveling salesman problem," *Soft Comput.*, vol. 21, no. 14, pp. 3891–3905, 2017.
- [20] E. J. Teoh, K. C. Tan, H. J. Tang, C. Xiang, and C. K. Goh, "An asynchronous recurrent linear threshold network approach to solving the traveling salesman problem," *Neurocomputing*, vol. 71, nos. 7–9, pp. 1359–1372, 2008.
- [21] K. Smith, "An argument for abandoning the travelling salesman problem as a neural-network benchmark," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1542–1544, Nov. 1996.
- [22] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological*, vol. 52, no. 3, pp. 141–152, 1985.
- [23] S. V. B. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield model," *IEEE Trans. Neural Netw.*, vol. 1, no. 2, pp. 204–215, Jun. 1990.
- [24] J. Zheng and S. Sun, "A modified algorithm and theoretical analysis for Hopfield network solving TSP," *Acta Electron. Sinica*, vol. 23, no. 1, pp. 73–78, 1995.
- [25] X. Wang, Y. Lu, Y. Zhang, Z. Zhao, T. Xia, and L. Xiao, "Power optimization in logic synthesis for mixed polarity Reed-Müller logic circuits," *Comput. J.*, vol. 58, no. 6, pp. 1306–1313, 2015.



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. He has authored or coauthored over 20 papers in peer-reviewed journals and proceedings. His research interests include low power integrated circuit design and optimization, multiple-valued logic circuits, and intelligent algorithm. He is a member of the China Computer Federation.



JIA LIU was born in Tangshan, Hebei, China, in 1988. She received the M.E. degree from Hebei University, Baoding, China, in 2019. She is currently a Staff with the College of Information Science and Technology, Hebei Agricultural University. Her research interests include electronic design automation, intelligent algorithm, finance data mining, and data analysis.



ZHISHENG HUO received the M.S. degree from the College of Computer Science, Shenyang Aerospace University, Shenyang, China, in 2012, and the Ph.D. degree in computer architecture from Beihang University, Beijing, China, in 2018, where he currently holds a postdoctoral position with the School of Computer Science and Engineering. His research focuses on big data storage and distributed storage systems.



LIMIN XIAO is currently a Professor with the School of Computer Science and Engineering, Beihang University, China. His main research areas include computer architecture, computer system software, high performance computing, virtualization, and cloud computing. He is a Senior Member of the China Computer Federation.



XIANG WANG is currently a Professor with the School of Electronic and Information Engineering, Beihang University, China. His main research areas include very large-scale integration, micro-nano systems, genetic circuits, and aerospace information networks. He is a Senior Member of the Chinese Institute of Electronics and the Chinese Society of Micro-Nano Technology.

...