He ZX, Xiao LM, Ruan L *et al.* A power and area optimization approach of mixed polarity Reed-Muller expression for incompletely specified Boolean functions. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(2): 297–311 Mar. 2017. DOI 10.1007/s11390-017-1723-1

A Power and Area Optimization Approach of Mixed Polarity Reed-Muller Expression for Incompletely Specified Boolean Functions

Zhen-Xue He [1,2], Li-Min Xiao[1,2,*], *Senior Member, CCF*, Li Ruan [1,2], *Senior Member, CCF*, Fei Gu [2]
Zhi-Sheng Huo [1,2], Guang-Jun Qin [3], *Member, CCF*, Ming-Fa Zhu [1,2], *Senior Member, CCF*
Long-Bing Zhang [4,5], Rui Liu [3,6], and Xiang Wang [3]

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*

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

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

[4] *State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences Beijing 100190, China*

[5] *University of Chinese Academy of Sciences, Beijing 100049, China*

[6] *National Engineering Research Center for Science and Technology Resources Sharing Service, Beihang University Beijing 100191, China*

E-mail: {hezhenxue, xiaolm, ruanli, gufei, huozhisheng1122, qingj, zhumf}@buaa.edu.cn; lbzhang@ict.ac.cn {lr, wxiang}@buaa.edu.cn

Received December 19, 2015; revised January 4, 2017.

Abstract The power and area optimization of Reed-Muller (RM) circuits has been widely concerned. However, almost none of the exiting power and area optimization approaches can obtain all the Pareto optimal solutions of the original problem and are efficient enough. Moreover, they have not considered the don't care terms, which makes the circuit performance unable to be further optimized. In this paper, we propose a power and area optimization approach of mixed polarity RM expression (MPRM) for incompletely specified Boolean functions based on Non-Dominated Sorting Genetic Algorithm II (NSGA-II). Firstly, the incompletely specified Boolean function is transformed into zero polarity incompletely specified MPRM (ISMPRM) by using a novel ISMPRM acquisition algorithm. Secondly, the polarity and allocation of don't care terms of ISMPRM is encoded as chromosome. Lastly, the Pareto optimal solutions are obtained by using NSGA-II, in which MPRM corresponding to the given chromosome is obtained by using a chromosome conversion algorithm. The results on incompletely specified Boolean functions and MCNC benchmark circuits show that a significant power and area improvement can be made compared with the existing power and area optimization approaches of RM circuits.

Keywords power and area optimization, Reed-Muller (RM) circuit, Pareto optimal solution, don't care term, chromosome conversion

1 Introduction

Digital logic functions can be expressed in either AND/OR/NOT based Boolean logic or AND/XOR or OR/XNOR based Reed-Muller (RM) logic. For some circuits, such as the arithmetic circuits, parity check circuits, and communication circuits, the RM representation is more efficient than the Boolean represen-

Regular Paper

This work is supported by the National Natural Science Foundation of China under Grant Nos. 60973106, 61370059, 61232009, and 81571142, Beijing Natural Science Foundation under Grant No. 4152030, the Fundamental Research Funds for the Central Universities of China under Grant Nos. YWF-15-GJSYS-085 and YWF-14-JSJXY-14, the Fund of the State Key Laboratory of Computer Architecture of China under Grant No. CARCH201507, the Open Project Program of National Engineering Research Center for Science and Technology Resources Sharing Service (Beihang University), and the Fund of the State Key Laboratory of Software Development Environment of China under Grant No. SKLSDE-2016ZX-15.

*Corresponding Author

tation in power, area, speed and testability[1-2]. Notably, functions which do not produce efficient solutions in Boolean representation can often be realized efficiently in RM representation[3]. Recently, RM logic has attracted wide attention, and many synthesis methods of RM circuits have been proposed. They can be mainly classified into polarity conversion[4-7], polarity searching[8-10], and circuit performance optimization[11-14].

With the advent of the era of deep sub-micron integrated circuit (IC) and the feature size of chip shrinking to nanometers, the fast-growing power and area has become the main bottleneck restricting the development of IC. At present, the IC design has changed from uniform pursuit to high-density and high-performance to consider the power and area together. Accordingly, many power and area optimization approaches of RM circuits have been proposed. However, almost all of them are based on weighted sum method (WSM). WSM suggests converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular compromise solution at a time, and it needs to be run many times according to the different weights set by the designer. Additionally, it is difficult to determine the weight coefficients in practice. Most notably, the WSM has two fatal flaws. It cannot obtain the Pareto optimal solutions in the non-convex regions of Pareto front and uniform distributed Pareto optimal solutions[15]. Therefore, the existing WSM-based power and area optimization approaches cannot obtain all the Pareto optimal solutions and are not efficient enough. Moreover, researches have shown that the RM circuits can be further optimized by using the don't care terms properly[16-17]. However, few researches use the don't care terms to optimize the power and area of mixed polarity RM circuits.

In this paper, we propose an NSGA-II (non-dominated sorting genetic algorithm II)[18] based power and area optimization approach (NSGA-II-PAOA) of MPRM (mixed polarity RM expression) for incompletely specified Boolean functions. Compared with existing power and area optimization approaches, our main contributions are as follows.

1) We propose a power and area optimization approach of mixed polarity RM circuits. It fully makes use of the advantage of don't care terms in optimizing circuit performance and NSGA-II in solving multi-objective optimization problem. Moreover, it can be extended to optimize the power and area of fixed polarity RM logic circuits.

2) We propose an incompletely specified MPRM (ISMPRM) acquisition algorithm that can transform the incompletely specified Boolean function into the zero polarity ISMPRM. It enables to treat the incompletely specified Boolean functions efficiently without any constraints on the number of variables.

3) We encode the polarity and allocation of don't care terms as chromosome and propose a chromosome conversion algorithm that can convert the zero polarity ISMPRM to the MPRM corresponding to the given chromosome.

4) We compare NSGA-II-PAOA with the existing power and area optimization approaches on incompletely specified Boolean functions and MCNC benchmark circuits. Experimental results show that NSGA-II-PAOA outperforms the existing power and area optimization approaches in terms of the number of Pareto optimal solutions and solving efficiency.

The rest of the paper is organized as follows. Section 2 provides related work. A power and area optimization approach of MPRM for incompletely specified Boolean functions is described in detail in Section 3. Experimental results are presented in Section 4. Conclusions are drawn in Section 5.

2 Related Work

2.1 Power and Area Optimization Approaches of RM Circuits

Recently, much work has been carried out in power and area optimization of RM circuits. Almost all of them are based on WSM (e.g., [19-23]). To get power-area trade-off, these approaches allocate the weights for the power and area cost functions in the fitness function design, which can be defined as follows:

$$fitness(i) = \alpha/Power_{\cos t}(i) + (1-\alpha)/Area_{\cos t}(i),$$

where $fitness(i)$ represents the fitness value corresponding to chromosome i, α represents the weight, $0 \leqslant \alpha \leqslant 1$, $Power_{\cos t}(i)$ and $Area_{\cos t}(i)$ represent the power and the area corresponding to chromosome i respectively.

However, the WSM-based power and area optimization approach cannot obtain all the Pareto optimal solutions of the original problem due to that WSM cannot obtain the Pareto optimal solutions in the non-convex regions of Pareto front, which can be illustrated by the trigonometric linear combinations (TLC) problem. The

TLC problem can be represented as follows:

$$F(x) = \frac{\cos\theta}{\sin\theta + \cos\theta} f_1(x) + \frac{\sin\theta}{\sin\theta + \cos\theta} f_2(x),$$

where $\theta \in [0, \frac{\pi}{2}]$. As shown in Fig.1(a), a value of θ corresponds to a Pareto optimal solution. Under the condition that the value of θ is determined, to minimize $F(x)$ is to search the coordinate points with minimum f_2 intercept. Since the intercept d_1 of points A and C is less than the intercept d_2 of point B, the Pareto solutions in non-convex regions (such as B) cannot be obtained, as shown in Fig.1(b).

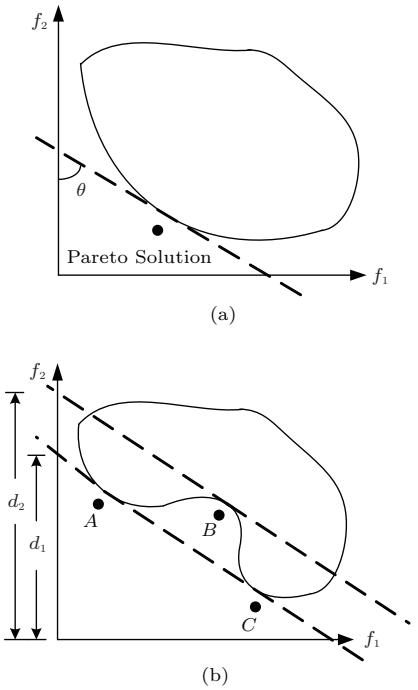


Fig.1. Demonstrating the TLC problem.

2.2 Optimization Approaches of RM Circuits Considering the Don't Care Terms

Researches have shown that using the don't care terms properly could simplify the expression and optimize the circuit performance[24-25]. Recently, many researchers have shown interest in using the don't care terms to optimize RM circuits (e.g., [24-28]).

However, almost all of the existing optimization approaches considering the don't care terms focused on the minimization of RM expressions and polarity conversion, without using the don't care terms to optimize the power and the area of RM circuits simultaneously. Although there was a power and area optimization approach considering the don't care terms[20], it only applies to fixed polarity RM circuits.

3 Power and Area Optimization Approach of MPRM for Incompletely Specified Boolean Functions

In this section, we introduce a novel optimization approach called NSGA-II-PAOA to obtain the mixed polarity RM expressions (MPRMs) with lower power and smaller area. Firstly, the incompletely specified Boolean function is transformed into zero polarity ISMPRM by using a novel ISMPRM acquisition algorithm. Secondly, the polarity and allocation of don't care terms is encoded as a chromosome. Finally, the Pareto optimal solutions (chromosomes) are generated by using NSGA-II, in which MPRM corresponding to the given chromosome is obtained by using a chromosome conversion algorithm.

The ISMPRM acquisition algorithm, chromosome encoding scheme, chromosome conversion algorithm, power and area fitness functions, and crossover and mutation operations are discussed in details as follows. Due to space constraints, the roulette wheel selection, fast non-dominated sorting and calculation of the crowing distances will not be covered again here.

3.1 ISMPRM Acquisition Algorithm

In order to transform the incompletely specified Boolean function into the zero polarity ISMPRM, we propose an ISMPRM acquisition algorithm (IAA), which is an extension to the tabular technique[29]. It is summarized in Algorithm 1. The minterm or product term is represented by the row of the table. Moreover,

Algorithm 1. IAA

Input: an n-variable incompletely specified Boolean function with t don't care terms

Output: zero polarity ISMPRM

1: Represent the incompletely specified Boolean function using the tabular notation
2: Set the flag f of minterms to 1 and the flag f of don't care terms to d_i, $1 \leqslant i \leqslant t$
3: Let $j = n$
4: If any row of the table satisfies $< c_n ... c_{j+1} 0 c_{j-1} ... c_1 f >$, generate the new row $< c_n ... c_{j+1} 1 c_{j-1} ... c_1 f >$
5: If the newly generated row is equal to any existing row except for the flag, we perform the following operations:
1) Performing the XOR operation on the flags of the two rows
2) Assigning the operation result to the flag of already existing row
3) Deleting the newly generated row
Otherwise, put the newly generated row into the trail of the original table
6: If $j > 1$ then let $j = j - 1$, go to step 4. Otherwise, output the zero polarity ISMPRM

the column of the table denotes the variable of Boolean function or MPRM. Each cell of the table is either 0 or 1, which represents the state of each variable in minterm or product term. The above description is illustrated in example 1.

Example 1. Given a 4-variable incompletely specified Boolean function with four don't care terms $f(x_4, x_3, x_2, x_1) = \Sigma m(1, 5, 11, 14) + \Sigma d(2, 7, 10, 12)$, the zero polarity ISMPRM can be obtained by the following steps.

Step 1: construct a table representing the incompletely specified Boolean function.

x_4	x_3	x_2	x_1	Flag
0	0	0	1	1
0	1	0	1	1
1	0	1	1	1
1	1	1	0	1
0	0	1	0	d_1
0	1	1	1	d_2
1	0	1	0	d_3
1	1	0	0	d_4

Step 2: for x_4, if there are rows $< 0c_3c_2c_1 >$ in the table, generate new rows $< 1c_3c_2c_1 >$.

x_4

	Boolean function						Generated terms				
	x_4	x_3	x_2	x_1	Flag		x_4	x_3	x_2	x_1	Flag
	0	0	0	1	1		1	0	0	1	1
	0	1	0	1	1		1	1	0	1	1
	1	0	1	1	1	*1	1	0	1	0	d_1
	1	1	1	0	1		1	1	1	1	d_2
	0	0	1	0	d_1						
	0	1	1	1	d_2						
*1	1	0	1	0	d_3						
	1	1	0	0	d_4						

Step 3: perform the XOR operation on the flags of the identical two rows and add any remaining new rows to the foot of the table; for x_3, if there are rows $< c_40c_2c_1 >$ in the table, generate new rows $< c_41c_2c_1 >$.

x_3

	Transformed Function						Generated Terms				
	x_4	x_3	x_2	x_1	Flag		x_4	x_3	x_2	x_1	Flag
	0	0	0	1	1	*1	0	1	0	1	1
*1	0	1	0	1	1	*2	1	1	1	1	1
	1	0	1	1	1		0	1	1	0	d_1
*3	1	1	1	0	1	*3	1	1	1	0	$d_3 \oplus d_1$
	0	0	1	0	d_1	*4	1	1	0	1	1
	0	1	1	1	d_2						
	1	0	1	0	$d_3 \oplus d_1$						
	1	1	0	0	d_4						
	1	0	0	1	1						
*4	1	1	0	1	1						
*2	1	1	1	1	d_2						

Step 4: Perform the XOR operation on the flags of the identical two rows and add any remaining new rows to the foot of the table. For x_2, if there are rows $< c_4c_30c_1 >$ in the table, generate new rows $< c_4c_31c_1 >$.

x_2

	Transformed Function						Generated Terms				
	x_4	x_3	x_2	x_1	Flag		x_4	x_3	x_2	x_1	Flag
	0	0	0	1	1		0	0	1	1	1
	0	1	0	1	0	*1	0	1	1	1	0
*3	1	0	1	1	1	*2	1	1	1	0	d_4
*2	1	1	1	0	$\overline{d_3} \oplus \overline{d_1}$	*3	1	0	1	1	1
	0	0	1	0	d_1	*4	1	1	1	1	0
*1	0	1	1	1	d_2						
	1	0	1	0	$d_3 \oplus d_1$						
	1	1	0	0	d_4						
	1	0	0	1	1						
	1	1	0	1	0						
*4	1	1	1	1	$\overline{d_2}$						
	0	1	1	0	d_1						

Step 5: perform the XOR operation on the flags of the identical two rows and add any remaining new rows to the foot of the table; for x_1, since there are rows $< c_4c_3c_20 >$ in the table, generate new rows $< c_4c_3c_21 >$.

x_1

	Transformed Function						Generated Terms				
	x_4	x_3	x_2	x_1	Flag		x_4	x_3	x_2	x_1	Flag
	0	0	0	1	1	*1	1	1	1	1	$\overline{d_3} \odot \overline{d_1} \oplus d_4$
	0	1	0	1	0	*2	0	0	1	1	d_1
*3	1	0	1	1	0	*3	1	0	1	1	$d_3 \oplus d_1$
	1	1	1	0	$\overline{d_3} \odot \overline{d_1} \oplus d_4$	*4	1	1	0	1	d_4
	0	0	1	0	d_1	*5	0	1	1	1	d_1
*5	0	1	1	1	d_2						
	1	0	1	0	$d_3 \oplus d_1$						
	1	1	0	0	d_4						
	1	0	0	1	1						
*4	1	1	0	1	0						
*1	1	1	1	1	$\overline{d_2}$						
	0	1	1	0	d_1						
*2	0	0	1	1	1						

Step 6: perform the XOR operation on the flags of the identical two rows and add any remaining new rows to the foot of the table; the zero polarity ISMPRM can be represented as below.

x_4	x_3	x_2	x_1	Flag
0	0	0	1	1
0	1	0	1	0
1	0	1	1	$d_3 \oplus d_1$
1	1	1	0	$\overline{d_3} \odot \overline{d_1} \oplus d_4$
0	0	1	0	d_1
0	1	1	1	$d_2 \oplus d_1$
1	0	1	0	$d_3 \oplus d_1$
1	1	0	0	d_4
1	0	0	1	1
1	1	0	1	d_4
1	1	1	1	$\overline{d_2} \oplus (\overline{d_3} \odot \overline{d_1} \oplus d_4)$
0	1	1	0	d_1
0	0	1	1	$\overline{d_1}$

3.2 Chromosome Encoding Scheme

The polarity and allocation of don't care terms is encoded as chromosome due to the fact that both the polarity and the allocation of don't care terms could affect the circuit performance. Since the polarity denotes the appearance form of variable in MPRM, the first half (polarity) of chromosome is represented in ternary form. Furthermore, since the allocation of don't care terms indicates whether or not each don't care term is to be written to the expression, the second half (allocation of don't care terms) of chromosome is represented in binary form. The above description is illustrated in example 2.

Example 2. Given a 4-variable ISMPRM with four don't care terms, the polarity and the allocation of don't care terms of the ISMPRM can be encoded as 8-bit chromosome, as shown in Fig.2.

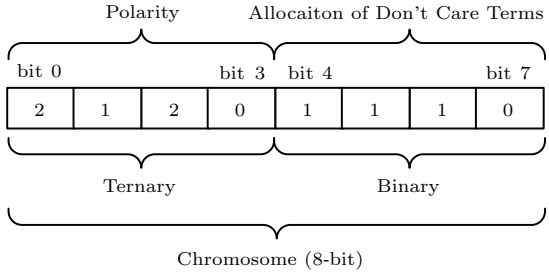


Fig.2. Details of chromosome for example 2.

Fig.2 shows that bit 0∼bit 3 denote polarity $(2120)_3 = (69)_{10}$. Bit 4∼bit 7 denote the allocation of don't care terms, namely, the fourth don't care term is abandoned, whereas the remaining three don't care terms are written to the expression.

3.3 Chromosome Conversion Algorithm

In this subsection, we describe a chromosome conversion algorithm (CCA) to transform the zero polarity ISMPRM into MPRM corresponding to the given chromosome, so as to calculate the fitness value of chromosome. The main idea behind CCA is that: firstly, the flag in each row is calculated according to the allocation of don't care terms in given chromosome; secondly, the zero polarity MPRM is obtained by removing a row whose flag is zero; lastly, the zero polarity MPRM is converted into MPRM corresponding to the given chromosome according to the polarity in given chromosome. The above description is illustrated in example 3.

Example 3. The zero polarity ISMPRM obtained by step 6 in example 1 is served as the given zero polarity ISMPRM. The chromosome "21201110" in example 2 is served as the given chromosome, and the corresponding MPRM can be obtained by the following steps.

Step 1: since the allocation of don't care terms in given chromosome is "1110", the flags of don't care terms d_1, d_2, d_3, d_4 are: $d_1 = 1,\ d_2 = 1,\ d_3 = 1,\ d_4 = 0$, respectively.

Step 2: calculate the flag in each row based on d_1, d_2, d_3, d_4.

x_4	x_3	x_2	x_1	Flag
0	0	0	1	1
0	1	0	1	0
1	0	1	1	0
1	1	1	0	1
0	0	1	0	1
0	1	1	1	0
1	0	1	0	0
1	1	0	0	0
1	0	0	1	1
1	1	0	1	0
1	1	1	1	1
0	1	1	0	1
0	0	1	1	0

Step 3: obtain the zero polarity MPRM by removing a row whose flag is 0.

x_4	x_3	x_2	x_1	
0	0	0	1	(x_1)
1	1	1	0	$(x_4x_3x_2)$
0	0	1	0	(x_2)
1	0	0	1	(x_4x_1)
1	1	1	1	$(x_4x_3x_2x_1)$
0	1	1	0	(x_3x_2)

The zero polarity MPRM can be expressed as follows:

$$f(x_4, x_3, x_2, x_1) = x_1 \oplus x_4x_3x_2 \oplus x_2 \oplus x_4x_1 \oplus x_4x_3x_2x_1 \oplus x_3x_2.$$

Step 4: the zero polarity MPRM is converted to the MPRM corresponding to polarity "2120" by using the mixed polarity conversion algorithm[29]. Due to space constraints, the polarity conversion process will not be covered again here. MPRM that corresponds to chromosome "21201110" is:

x_4	x_3	x_2	x_1	
0	1	0	1	$(\bar{x}_4\bar{x}_2x_1)$
1	1	1	0	(x_4x_2)
1	0	1	0	$(x_4\bar{x}_3x_2)$
1	1	1	1	$(x_4x_2x_1)$

The resulting MPRM is presented by the following equation:

$$f(x_4, x_3, x_2, x_1) = \overline{x}_4\overline{x}_2x_1 \oplus x_4x_2 \oplus x_4\overline{x_3}x_2 \oplus x_4x_2x_1.$$

3.4 Power and Area Fitness Functions

3.4.1 Power Fitness Function

With the rapid development of semi-conductor technology and the continuous improvement of integration, power has become a major bottleneck which restricts the development of IC. Power in CMOS circuits can be broadly divided into two categories: dynamic power and leakage power.

Dynamic Power Fitness Function. Dynamic power results mainly from the charging and discharging of the load capacitances[30], which can be expressed as follows:

$$p = 0.5V_{dd}^2 f_{\text{clk}} \sum_{i=1}^{n} C_{\text{L}}^i SW^i,$$

where V_{dd} is the supply voltage, f_{clk} is the clock frequency, C_{L}^i is the load capacitance at the output of gate i, and SW^i (referred to as switching activity) is the number of transitions per cycle at gate i. Power consumption in a logic circuit can be best modeled by the switching activity on the circuit nodes. Therefore, we use the switching activity to estimate the power. The power estimation model[13] considering temporal correlations is used to estimate the circuit power accurately. Since the lower the power, the better the quality of chromosome, the dynamic power fitness function is defined:

$$fitness_{\text{dynamic\_power}}(i) = 1/SW(i),$$

where $SW(i)$ represents the switching activity corresponding to chromosome i.

Leakage Power Fitness Function. At present, the feature size of transistors has advanced to the deep sub-micron and the leakage power has made a significant contribution to the total power. However, few researches have focused on optimizing the leakage power of RM circuits. To the best of our knowledge, [31] is the only work to consider the leakage power of RM circuits, which is used to estimate the leakage power of RM circuits. Accordingly, the leakage power is defined as:

$$Leakage(i) = AND_{\text{leakage}}(i) + XOR_{\text{leakage}}(i),$$

where $Leakage(i)$ represents the leakage power of chromosome i. $AND_{\text{leakage}}(i)$ and $XOR_{\text{leakage}}(i)$ denote the leakage power of all the AND gates and XOR gates corresponding to chromosome i respectively. For more information about the calculation of leakage power of AND gates and XOR gates, please refer to [31]. Similarly, the smaller the leakage power, the better the quality of chromosome. Therefore, the leakage power fitness function is defined as:

$$fitness_{\text{leakage\_power}}(i) = 1/Leakage(i).$$

3.4.2 Area Fitness Function

The multi-input AND gates and XOR gates need to be decomposed into two-input AND gates and XOR gates to estimate the power[19]. Therefore, the area is quantified in terms of the total sum of the number of two-input AND gates and XOR gates, which is defined as follows:

$$Area(i) = Num\_AND(i) + Num\_XOR(i),$$

where $Area(i)$ represents the area corresponding to chromosome i. $Num\_AND(i)$ and $Num\_XOR(i)$ represent the number of AND gates and XOR gates corresponding to chromosome i respectively. Since the smaller the area, the better the quality of chromosome, the area fitness function is defined as:

$$fitness_{\text{area}}(i) = 1/Area(i).$$

3.5 Crossover and Mutation Operations

3.5.1 Crossover Operation

The one-point crossover is used to produce offspring. It is worthwhile to note that if the crossover point appears at the polarity (allocation of don't care terms) in parent chromosomes, the one-point crossover does not work for the allocation of don't care terms (polarity) in children chromosomes, which can be illustrated by Fig.3 and Fig.4.

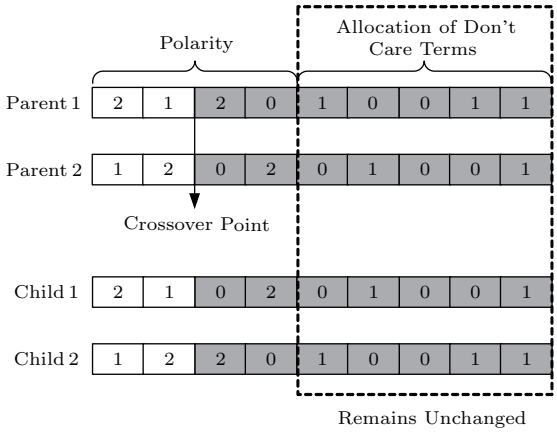


Fig.3. When the crossover point appears at polarity, the allocation of don't care terms will remain unchanged after one-point crossover.

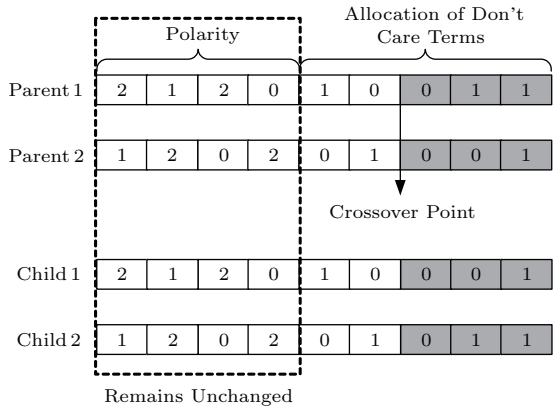


Fig.4. When the crossover point appears at the allocation of don't care terms, the polarity will remain unchanged after one-point crossover.

In order to make the one-point crossover do work for the polarity as well as the allocation of don't care terms, we perform the one-point crossover on the polarity and allocation of don't care terms, which is illustrated by example 4.

Example 4. Suppose the two parent chromosomes selected for crossover are "parent 1" and "parent 2" as shown in Fig.5.

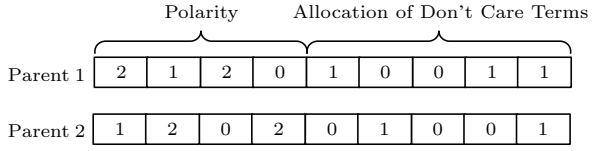


Fig.5. Two parent chromosomes selected for crossover.

As shown in Fig.6, firstly, the crossover points 1 and 2 for the polarity and the allocation of don't care terms are randomly generated; secondly, the children chromosomes are generated by performing the one-point crossover on the polarity and the allocation of don't care terms. Fig.6 shows that the one-point crossover does work for the polarity as well as the allocation of don't care terms.

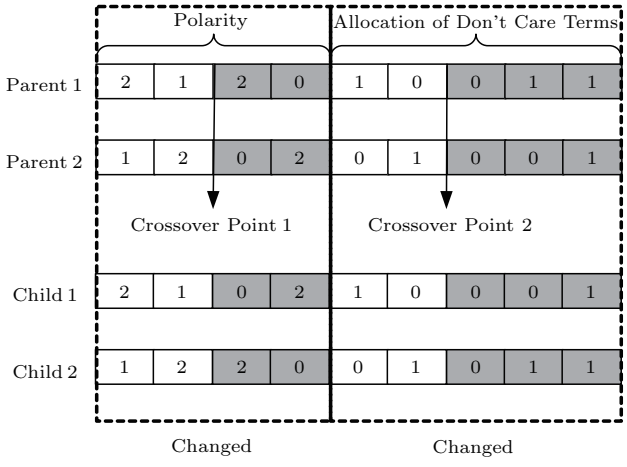


Fig.6. Example demonstrating one-point crossover.

3.5.2 Mutation Operation

The simplest bit mutation is used to maintain genetic diversity in population[32]. Similarly, in order to make the bit mutation do work for the polarity as well as the allocation of don't care terms, we perform the bit mutation on the polarity and allocation of don't care terms, respectively. Since the allocation of don't care terms is encoded as binary, 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 0. It should be noted that the polarity is encoded as ternary. Therefore, the bit mutation for the polarity needs to be modified, 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.

3.5.3 Algorithm Description

Based on the above description, NSGA-II-PAOA is illustrated in Algorithm 2, in which "*popsize*" represents the size of population, "*2popsize*" represents the size of combined population, "*iteration*" represents the current number of iteration, and "*Maxiteration*" represents the maximum number of iteration.

4 Experimental Results

NSGA-II-PAOA had been implemented in C language, and the programs were compiled by the GNU C complier. The results were obtained by using a PC with Intel Core i7 3.40 GHz with 4 G RAM under Linux. In order to validate the effectiveness of NSGA-II-PAOA in optimizing power and area, firstly, we compared NSGA-II-PAOA with the area and power optimization approach[20] (APOA) which searches the best polarity and then searches the best allocation of don't care terms; secondly, we compared NSGA-II-PAOA with the WSM-based power and area optimization approach[19] (WSM-PAOA) which also considers the don't care terms.

Algorithm 2. NSGA-II-PAOA

Input: incompletely specified Boolean function
Output: Pareto optimal solutions

```
IAA(incompletely specified Boolean function);
Encode the polarity and the allocation of don't care terms as chromosome;
Establish the dynamic power, leakage power and area fitness functions;
Initialize the parameters;
Randomly generate the initial population;
for i = 1 to popsize do
    CCA(zero polarity ISMPRM);
    Dynamic-power-fitness(MPRM);
    Leakage-power-fitness(MPRM);
    Area-fitness(MPMR);
end for
Non-dominated-sorting(initial population);
Roulette wheel selection;
One point crossover;
Bit mutation;
for iteration = 1 to Maxiteration do
    Combine(parent population, child population);
    for i = 1 to 2popsize do
        CCA(zero polarity ISMPRM);
        Dynamic-power-fitness(MPRM);
        Leakage-power-fitness(MPRM);
        Area-fitness(MPMR);
    end for
    Fast non-dominated sorting;
    Calculate the crowding distance;
    Form parent population;
    Roulette wheel selection;
    One point crossover;
    Bit mutation;
end for
Output the Pareto optimal solutions
```

4.1 Comparison of NSGA-II-PAOA and APOA

In order to calculate the switching activity accurately, 11 input signal probabilities were generated under the random function, which are 0.64, 0.03, 0.17, 0.91, 0.35, 0.57, 0.85, 0.15, 0.48, 0.53, 0.24, respectively. For different input variables, the input signal probabilities were chosen from the left to the right in above list. For example, for 3-variable incompletely specified Boolean function $f(x_3, x_2, x_1)$, signal probabilities for x_3, x_2, x_1 were 0.17, 0.03, 0.64, respectively. Moreover, we ran NSGA-II-PAOA and APOA 10 times and took the best obtained results as experimental data. To guarantee the fairness of experiment, the parameters were set based on [20], which are shown in Table 1.

Table 1. Parameters of NSGA-II-PAOA and APOA

Parameter	NSGA-II-PAOA	APOA
Population size	100.00	100.00
Maximum number of iteration	200.00	200.00
Local number of iteration	-	20.00
Weight in fitness function	-	0.50
Crossover probability	0.90	-
Maximum crossover probability	-	0.80
Minimum crossover probability	-	0.40
Mutation probability	0.01	-
Maximum mutation probability	-	0.10
Minimum mutation probability	-	0.01
Maximum local search value	-	1.10
Minimum local search value	-	1.01
Local search crossover probability	-	0.10
Local search mutation probability	-	0.40

The comparison between NSGA-II-PAOA and APOA is listed in Table 2. In this table, $f(n, t, d, s)$ represents n-variable function with t minterms and d don't care terms, where s represents the seeds for the function generator. Columns 2∼4 show the area, switching activity (SW) and leakage power (microwatts) obtained by APOA, respectively. Column 5 shows the average time (in seconds) for generating MPRM using APOA. Column 6 shows the Pareto optimal solutions obtained by NSGA-II-PAOA, where "A" represents the solution with minimum area, "S" represents the solution with minimum switching activity, and "L" represents the solution with minimum leakage power. Columns 7∼9 show the area, switching activity and leakage power (microwatts) corresponding to the Pareto optimal solutions, respectively. Column 10 shows the average time for generating the Pareto optimal solutions using NSGA-II-PAOA. Columns 11∼14 denote the percentage of area saved ($Save_{\text{area}}$), switching activity saved ($Save_{\text{sw}}$), leakage power saved ($Save_{\text{leak}}$) and time saved ($Save_{\text{time}}$), which are defined as:

$$Save_{\text{area}} = \frac{APOA_{\text{area}} - NSGAIIPAOA_{\text{area}}}{APOA_{\text{area}}},$$
$$Save_{\text{sw}} = \frac{APOA_{\text{sw}} - NSGAIIPAOA_{\text{sw}}}{APOA_{\text{sw}}} \times 100\%,$$
$$Save_{\text{leak}} = \frac{APOA_{\text{leak}} - NSGAIIPAOA_{\text{leak}}}{APOA_{\text{leak}}} \times 100\%,$$
$$Save_{\text{time}} = \frac{APOA_{\text{time}} - NSGAIIPAOA_{\text{time}}}{APOA_{\text{time}}} \times 100\%,$$

where $APOA_{\text{area}}$ and $NSGAIIPAOA_{\text{area}}$ represent the area obtained by APOA and NSGA-II-PAOA respectively, $APOA_{\text{sw}}$ and $NSGAIIPAOA_{\text{sw}}$ represent the switching activity obtained by APOA and NSGA-II-PAOA respectively, $APOA_{\text{leak}}$ and $NSGAIIPAOA_{\text{leak}}$ represent the leakage power obtained by APOA and NSGA-II-PAOA respectively, and

Table 2. Comparison of NSGA-II-PAOA and APOA

$f(n,t,d,s)$	APOA				NSGA-II-PAOA					Save (%)			
	Area	SW	Leak	Time (s)	C	Area	SW	Leak	Time (s)	Area	SW	Leak	Time (s)
6, 12, 40, 50	51	5.82	10.24	7.16	A	35	4.93	11.80	5.27	35.37	15.29	−15.23	26.40
					S	53	4.11	9.09		−3.92	29.38	11.23	
					L	46	5.85	8.72		9.80	−0.52	14.84	
7, 20, 90, 5	86	5.53	15.22	9.43	A	36	5.48	15.29	6.60	58.14	0.90	−0.46	30.01
					S	69	3.11	16.60		19.77	43.76	−9.07	
					L	88	5.60	12.64		2.33	−1.27	16.95	
8, 8, 240, 60	17	1.94	9.50	17.38	A	16	2.05	8.37	12.54	5.88	−5.67	11.89	27.85
					S	47	1.47	9.56		−176.47	24.23	−0.63	
					L	23	1.66	6.44		−35.29	14.43	32.21	
8, 25, 220, 50	73	4.86	75.12	23.49	A	45	3.40	76.17	16.31	38.36	30.04	−1.40	30.57
					S	91	3.23	85.99		−24.66	33.54	−14.47	
					L	76	5.08	74.82		−4.11	−4.53	0.40	
8, 100, 90, 10	291	13.27	114.55	21.20	A	225	13.76	146.03	15.43	22.68	−3.69	−27.48	27.22
					S	248	12.26	152.80		14.78	7.61	−33.39	
					L	291	13.27	114.55		0.00	0.00	0.00	
9, 200, 100, 5	639	33.78	94.61	48.97	A	516	36.29	107.67	26.70	19.25	−7.43	−13.80	45.48
					S	585	31.13	95.04		8.45	7.84	−0.45	
					L	653	43.60	93.52		−2.19	−29.07	1.15	
9, 250, 70, 5	1 053	56.92	174.66	68.49	A	1 047	54.29	175.34	31.16	0.57	4.62	−0.39	54.50
					S	1 133	49.86	170.25		−7.60	12.40	2.52	
					L	1 063	58.27	161.89		−0.95	−2.37	7.31	
10, 300, 150, 10	1 401	112.79	281.63	223.90	A	1 392	112.37	286.20	120.51	0.64	0.37	−1.62	46.18
					S	1 546	99.08	257.26		−10.35	12.16	8.65	
					L	1 670	112.84	246.35		−19.20	−0.04	12.53	
10, 500, 70, 25	2 147	195.48	130.65	379.22	A	1 972	210.04	128.26	162.04	8.15	−7.45	1.83	57.27
					S	2 148	194.53	131.17		−0.05	0.49	−0.40	
					L	2 076	199.10	126.38		3.31	−1.85	3.27	
11, 1000, 80, 1	4 058	264.66	348.62	951.43	A	3 670	271.51	354.24	375.60	9.56	−2.59	1.61	60.52
					S	4 058	264.66	348.62		0.00	0.00	0.00	
					L	3 815	302.11	322.31		5.99	−14.15	7.55	

$APOA_{\text{time}}$ and $NSGAIIPAOA_{\text{time}}$ represent the average time spent on APOA and NSGA-II-PAOA respectively.

From the simulation results on 10 randomly generated incompletely specified Boolean functions, we can find that both the NSGA-II-PAOA and APOA can generate the Pareto optimal solutions. Since these Pareto optimal solutions are incomparable, there is no comparison between NSGA-II-PAOA and APOA in terms of solution quality. Additionally, we found that the number of Pareto optimal solutions obtained by NSGA-II-PAOA is larger than that obtained by APOA. Therefore, compared with APOA, NSGA-II-PAOA can provide the designer more options. Moreover, it is worth mentioning that NSGA-II-PAOA is faster than APOA. The greatest time saving reached 60.52% and the average time saving reached 40.60%, which are attributed to the fact that APOA is divided into two phases: the best polarity searching phase neglecting don't care terms and the phase searching the best allocation of don't care terms. Since APOA needs to perform optimal searching twice and NSGA-II-PAOA only needs to perform once, NSGA-II-PAOA is faster than APOA.

4.2 Comparison of NSGA-II-PAOA and WSM-PAOA

To further assess the effectiveness of NSGA-II-PAOA, we compared NSGA-II-PAOA with WSM-PAOA. Since WSM-PAOA is based on WSM, we need to allocate the weights for the dynamic power, leakage power and area cost functions. The fitness function of WSM-PAOA is defined as:

$$fitness(i) = (\frac{w_1}{SW(i)} + \frac{w_2}{Leakage(i)} + \frac{w_3}{Area(i)}) \times \beta,$$

where $SW(i)$, $Leakage(i)$ and $Area(i)$ represent the switching activity, leakage power (microwatts) and area corresponding to chromosome i, respectively. Weights w_1, w_2 and w_3 can be set by the designer with $w_1 + w_2 + w_3 = 1$. The constant β is used to prevent the fitness value from being too small to participate in the subsequent computations. In this experiment, β is set to 100.

Moreover, we need to assign more values to weights w_1, w_2 and w_3 to obtain more solutions. There are many different combinations of w_1, w_2 and w_3, which satisfy $w_1 + w_2 + w_3 = 1$. But for the sake of space and for the distribution of weights, we select seven diffe-

rent combinations of w_1, w_2 and w_3, namely, (1, 0, 0), (0.5, 0.5, 0), (0, 1, 0), (0, 0.5, 0.5), (0, 0, 1), (0.5, 0.25, 0.25) and (0.5, 0, 0.5) for each function or circuit in the following experiments.

Eight randomly generated incompletely specified Boolean functions and eight randomly selected MCNC benchmark circuits were tested on NSGA-II-PAOA and WSM-PAOA respectively. Moreover, in order to calculate the switching activity accurately, 18 input signal probabilities were generated under the random function, which are 0.11, 0.73, 0.94, 0.62, 0.29, 0.15, 0.23, 0.67, 0.14, 0.38, 0.54, 0.06, 0.80, 0.46, 0.37, 0.30, 0.02, 0.97, respectively. Furthermore, in order to determine the parameters and then obtain the better experimental results, we conducted a series of experiments with different parameters. The parameters of two approaches are given in Table 3.

Table 3. Parameters of NSGA-II-PAOA and WSM-PAOA

Parameter	Value
Size of population	100.00
Number of iteration	200.00
Probability of crossover	0.90
Probability of mutation	0.01

Figs.7∼14 show the comparison of NSGA-II-PAOA and WSM-PAOA on eight randomly generated incompletely specified Boolean functions, respectively. The X-axis denotes area, the Y-axis denotes switching activity, and the Z-axis denotes leakage power. Moreover, the red circle points and blue square points represent the Pareto optimal solutions obtained by NSGA-II-PAOA and WSM-PAOA, respectively. Table 4 shows the comparison of NSGA-II-PAOA and WSM-PAOA on eight randomly selected MCNC benchmark circuits, respectively. Column 1 shows the circuit name. Column 2 shows the number of don't care terms added to circuits (dc). Column 3 shows the number of Pareto optimal solutions obtained by WSM-PAOA (num). Columns 4∼6 show the area, switching activity (SW) and leakage power corresponding to the Pareto optimal solutions obtained by WSM-PAOA, respectively. Column 7 shows the total run time of WSM-PAOA under seven different weights. Column 8 shows the number of Pareto optimal solutions obtained by NSGA-II-PAOA. Columns 9∼11 show the area, switching activity and leakage power corresponding to the Pareto optimal solutions obtained by NSGA-II-PAOA, respectively. Column 12 shows the run time of NSGA-II-PAOA. Column 13 shows the multiple relationship between NSGA-II-PAOA and WSM-PAOA in terms of the number of solutions. Column 14 shows the percentage of time saved by NSGA-II-PAOA compared with WSM-PAOA, which is defined as

$$Save_{\text{time}} = \frac{WSMPAOA_{\text{time}} - NSGAIIPAOA_{\text{time}}}{WSMPAOA_{\text{time}}} \times 100\%.$$

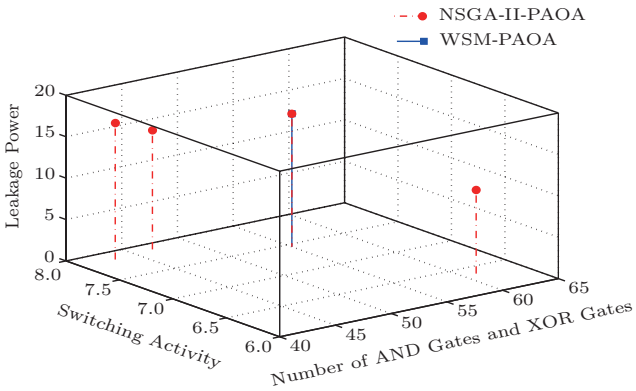


Fig.7. Comparison results on $f(6, 15, 40, 25)$.

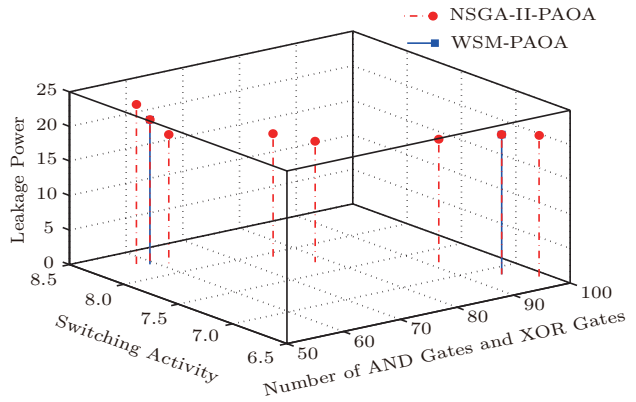


Fig.8. Comparison results on $f(7, 30, 80, 5)$.

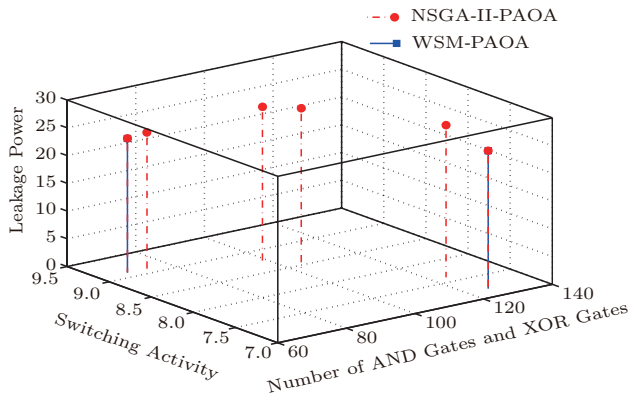


Fig.9. Comparison results on $f(7, 35, 40, 5)$.

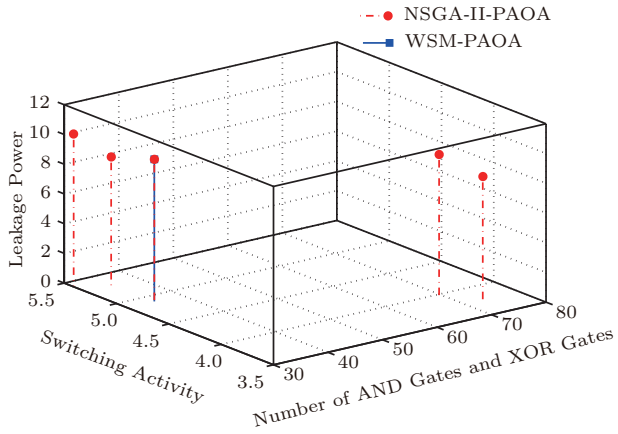


Fig.10. Comparison results on $f(8, 15, 230, 25)$.

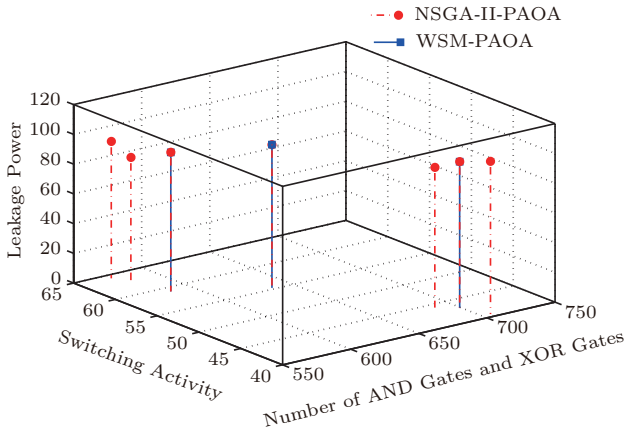


Fig.11. Comparison results on $f(9, 200, 50, 5)$.

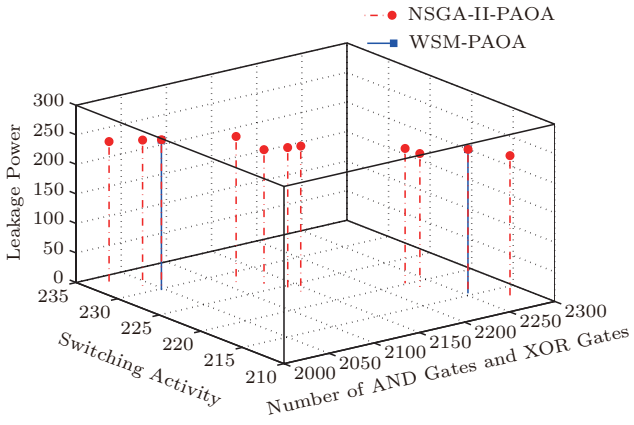


Fig.12. Comparison results on $f(10, 500, 30, 10)$.

Figs.7∼14 and Table 4 demonstrate that compared with WSM-PAOA, NSGA-II-PAOA can obtain more Pareto optimal solutions. For the eight incompletely specified Boolean functions, the maximum multiple reached 5.5 times. For the eight MCNC benchmark circuits, the maximum multiple reached four times. Moreover, the Pareto optimal solutions obtained by WSM-PAOA are not evenly distributed, whereas the Pareto optimal solutions obtained by NSGA-II-PAOA are well-distributed. Moreover, NSGA-II-PAOA has higher efficiency than WSM-PAOA. For the eight MCNC benchmark circuits, the greatest time saving reached 81.73% and the average time saving reached 72.94%. The above results can be explained by the following reasons.

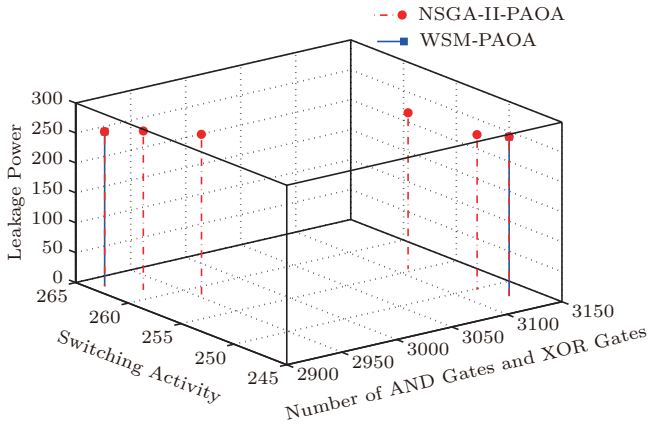


Fig.13. Comparison results on $f(10, 700, 40, 5)$.

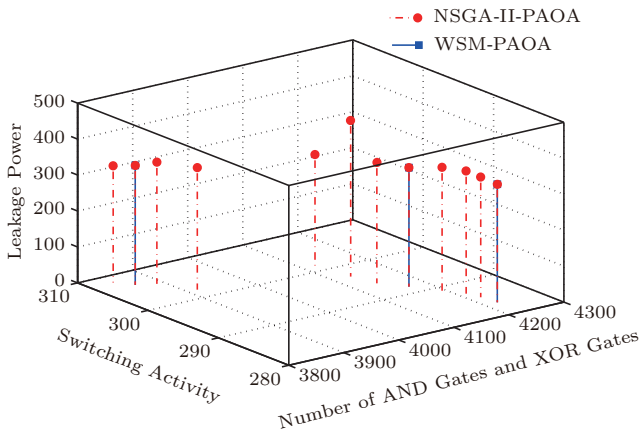


Fig.14. Comparison results on $f(11, 1\,000, 30, 1)$.

1) WSM cannot obtain the Pareto optimal solutions in the non-convex regions of Parent front and a uniform distribution set of Pareto optimal solutions in Pareto front. Therefore, WSM-PAOA has disadvantage of being unable to generate more Pareto optimal solutions.

2) In order to obtain the Pareto optimal solutions as many as possible, WSM-PAOA needs to run as many times as the value of the weight set by the designer, whereas NSGA-II-PAOA can generate a set of Pareto optimal solutions in one single run. Therefore, NSGA-II-PAOA has higher efficiency.

Table 4. Comparison of WSM-PAOA and NSGA-II-PAOA

Circuit	*dc*	WSM-PAOA					NSGA-II-PAOA					Number of	Time
		num	Area	SW	Leak	Time (s)	*num*	Area	SW	Leak	Time (s)	Solutions	Saved (%)
xor5	20	1	32	1.92	35.23	21.98	3	32	1.91	36.40	7.54	3.00	65.70
								32	1.92	35.23			
								40	1.96	34.78			
m1	50	2	32	9.06	41.50	20.02	4	32	9.06	41.50	4.63	2.00	76.87
			40	9.05	41.67			32	9.07	41.46			
								40	9.05	41.67			
								41	9.02	41.42			
rd84	190	2	258	98.69	184.52	99.75	6	256	97.33	184.79	26.98	3.00	72.95
			269	82.73	182.23			258	98.69	184.52			
								260	96.83	182.11			
								261	83.75	184.25			
								266	92.36	183.92			
								269	82.73	182.23			
clip	300	3	747	125.38	323.81	115.01	8	747	125.38	323.81	41.62	2.67	63.81
			888	94.80	324.52			750	123.21	323.68			
			935	128.25	321.09			751	123.21	322.45			
								862	92.80	325.76			
								888	94.80	324.52			
								927	118.56	322.70			
								935	128.25	321.09			
								937	116.40	323.17			
ex1010	250	1	216	24.80	147.63	39.95	4	214	24.80	153.33	9.77	4.00	75.54
								216	24.80	147.63			
								263	26.15	140.57			
								264	27.49	138.50			
dk48	200	1	176	36.15	108.26	34.19	3	175	35.94	110.57	8.48	3.00	75.20
								176	36.15	108.26			
								178	36.15	105.83			
14_4color	320	2	680	125.62	243.75	71.26	4	680	125.62	243.75	20.15	2.00	71.72
			688	125.63	241.28			683	125.60	243.14			
								687	125.63	242.50			
								688	125.63	241.28			
src1	280	2	763	62.65	190.44	94.36	5	761	61.27	193.81	17.24	2.50	81.73
			775	79.40	176.59			763	62.65	190.44			
								774	60.19	178.32			
								775	79.40	176.59			
								778	83.05	172.64			

3) Compared with the most multi-objective optimization methods, NSGA-II was able to find a much better spread of solutions and a better convergence to the true Pareto optimal front. Consequently, NSGA-II-PAOA can generate distributed uniformly Pareto optimal solutions.

5 Conclusions

In this paper, we proposed a novel approach that can optimize the power and the area of mixed polarity RM circuits simultaneously. The approach uses the don't care terms to further optimize the circuit performance. Moreover, the Pareto optimal solutions obtained by the approach are more diverse on the multi-objective space than those obtained by the existing power and area optimization approaches. The experimental results showed that the approach can optimize the power and the area of mixed polarity RM circuits quickly and effectively. The obtained MPRMs can be selected by the designer based on the importance of objectives.

The optimization of RM circuits, which considers the don't care terms, is a computationally hard problem, because the optimization space will increase exponentially with the increase of input variables and don't care terms. In future, we will study much more ef-

ficient optimization approaches considering the don't care terms. Moreover, we will introduce the delay optimization into our approach, so as to make the circuit comprehensive performance better.

References

[1] Wang P J, Li K P, Zhang H H. PMGA and its application in area and power optimization for ternary FPRM circuit. *Journal of Semiconductors*, 2016, 37(1): 015007.

[2] Xiao L M, He Z X, Ruan L, Zhang R, Xia T S, Wang X. Optimization of best polarity searching for mixed polarity Reed-Muller logic circuit. In *Proc. the 28th IEEE International System-on-Chip Conference*, Sept. 2015, pp.275-280.

[3] Wang X, Lu Y, Zhang Y, Zhao Z X, Xia T S *et al*. Probabilistic modeling during power estimation for mixed polarity Reed-Muller logic circuits. In *Proc. IEEE International Conference on Green Computing and Communications*, Aug. 2013, pp.1414-1418.

[4] Yu H Z, Jiang Z D, Wang P J, Li K P. GA-DTPSO algorithm and its application in area optimization of mixed polarity XNOR/OR circuits. *Journal of Computer-Aided Design & Computer Graphics*, 2015, 27(5): 946-952. (in Chinese)

[5] Bu D L, Jiang J H. Dual logic based polarity conversion and optimization of mixed polarity RM circuits. *Acta Electronica Sinica*, 2015, 43(1): 79-85. (in Chinese)

[6] Wang P J, Wang Z H, Xu R, Jiang Z D, Wang D S. Conversion algorithm for MPRM expansion. *Journal of Semiconductors*, 2014, 35(3): 035007.

[7] Wang Y H, Wang L Y, Xia Y S. A fast Reed-Muller fixed polarity conversion algorithm for large circuits. *Journal of Computer-Aided Design & Computer Graphics*, 2014, 26(11): 2091-2098. (in Chinese)

[8] Fu Q, Wang P J, Tong N, Wang M B, Zhang H H. Multi-strategy discrete particle swarm optimization. *Acta Electronica Sinica*, 2016, 44(5): 1202-1207. (in Chinese)

[9] Vijayakumari C K, Mythili P, James R K, Akhil K S. Optimal design of combinational logic circuits using genetic algorithm and Reed-Muller universal logic modules. In *Proc. International Conference on Embedded Systems*, July 2014.

[10] Li K P, Wang P J, Zhang H H. The search of the best power of ternary FPRM circuit based on simulated annealing genetic algorithm. *Journal of Zhejiang University* (*Science Edition*), 2016, 43(2): 190-199. (in Chinese)

[11] Das A, Pradhan S N. Shared Reed-Muller decision diagram based thermal-aware AND-XOR decomposition of logic circuits. *VLSI Design*, 2016: Article ID3191286.

[12] Zhang Q W, Wang P J, Hu J. Exact minimization of ESOP expressions based on hierarchical hypercube. *Journal of Computer-Aided Design & Computer Graphics*, 2016, 28(1): 172-179. (in Chinese)

[13] Wang X, Lu Y, Zhang Y, Zhao Z X, Xia T S *et al*. Power optimization in logic synthesis for mixed polarity Reed-Muller logic circuits. *The Computer Journal*, 2015, 58(6): 1307-1313.

[14] Das A, Pradhan S N. Thermal aware FPRM based AND-XOR network synthesis of logic circuits. In *Proc. the 2nd International Conference on Recent Trends in Information Systems*, Sept. 2015, pp.497-502.

[15] Kim I Y, de Weck O L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 2005, 29(2): 149-158.

[16] Debnath D, Sasao T. Exact minimization of fixed polarity Reed-Muller expressions for incompletely specified functions. In *Proc. Asia and South Pacific Design Automation Conference*, Jan. 2000, pp.247-252.

[17] Habib M K. A new approach to generate fixed-polarity Reed-Muller expansions for completely and incompletely specified functions. *Int. J. Electronics*, 2002, 89(11): 845-876.

[18] Deb K, Agarwal S, Pratap A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.

[19] Zhang H H, Wang P J. Polarity optimization of XNOR/OR circuit area and power based on weighted sum method. In *Proc. the 9th IEEE International Conference on ASIC*, Oct. 2011, pp.341-344.

[20] Wang P J, Wang D S, Jiang Z D, Zhang H H. Area and power optimization of ISFPRM circuits based on PSGA algorithm. *Acta Electronica Sinica*, 2013, 41(8):1542-1548. (in Chinese)

[21] Chaudhury S, Chattopadhyay S. Fixed polarity Reed-Muller network synthesis and its application in AND-OR/XOR-based circuit realization with area-power tradeoff. *IETE Journal of Research*, 2008, 54(5): 353-363.

[22] Wu W J, Wang P J, Zhang X Y, Wang L L *et al*. Search for the best polarity of multi-output RM circuits based on QGA. In *Proc. the 2nd International Symposium on Intelligent Information Technology Application*, Dec. 2008, pp.279-282.

[23] Xia Y, Ali B, Almaini A E A. Area and power optimization of FPRM function based circuits. In *Proc. International Symposium on Circuits and Systems*, May 2003, pp.329-332.

[24] Wang D S, Wang P J. Algorithm about minimization of MPRM expansions including don't care terms. *Journal of Zhejiang University* (*Science Edition*), 2014, 41(1): 38-42. (in Chinese)

[25] Wang D S, Wang P J, Sun F, Yu H Z. Fixed-polarity conversions for logic functions include don't care terms. *Journal of Circuits and Systems*, 2013, 18(1): 117-121. (in Chinese)

[26] Al Jassani B A, Urquhart N, Almaini A E A. Minimization of incompletely specified mixed polarity Reed-Muller functions using genetic algorithm. In *Proc. the 3rd International Conference on Signals, Circuits and Systems*, Nov. 2009.

[27] Debatosh D, Tsutomu S. Exact minimization of FPRMs for incompletely specified functions by using MTBDDs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer*, 2005, 88-A(12): 3332-3341.

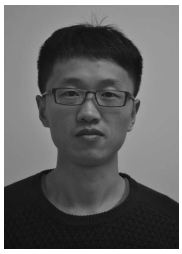
[28] Wang D S, Wang P J. Power optimization of incompletely specified fixed polarity Reed-Muller circuits. In *Proc. the 11th International Conference on Solid-State and Integrated Circuit Technique*, Oct.29-Nov.1, 2012.

[29] Almaini A E A, Mckenzie L. Tabular techniques for generating kronecker expansions. *IEE Proceedings—Computers and Digital Techniques*, 1996, 143(4): 205-212.

[30] Xia Y S, Wu X W, Almaini A E A. Power minimization of FPRM functions based on polarity conversion. *Journal of Computer Science and Technology*, 2003, 18(3): 325-331.

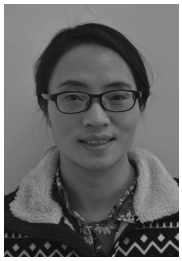
[31] Pradhan S N, Chattopadhyay S. AND-XOR network synthesis with area-power trade-off. In *Proc. the 3rd International Conference on Industrial and Information Systems*, Dec. 2008.

[32] Chu Z F, Xia Y S, Wang L Y. Cell mapping for nanohybrid circuit architecture using genetic algorithm. *Journal of Computer Science and Technology*, 2012, 27(1): 113-120.



Zhen-Xue He is a Ph.D. candidate in the School of Computer Science and Engineering, Beihang University, Beijing. He received his M.S. degree in computer application technology from Northwest Normal University, Lanzhou, in 2014. His research interests include electronic design automation, low power design and artificial intelligence.

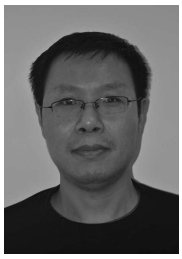
Li-Min Xiao is a professor of the School of Computer Science and Engineering, Beihang University, Beijing. He is a senior member of CCF. His main research areas are computer architecture, computer system software, high performance computing, virtualization and cloud computing.



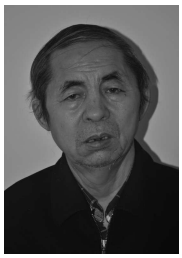
Li Ruan is a lecturer of the School of Computer Science and Engineering, Beihang University, Beijing. She is a senior member of CCF. Her main research areas are virtualization and cloud computing, computer system software, high performance computer.

Fei Gu is a Ph.D. candidate in the School of Computer Science and Engineering, Beihang University, Beijing. He received his M.S. degree in computer science and technology from Nanjing University of Aeronautics and Astronautics, Nanjing, in 2014, and his B.S. degree in software engineering from Yangzhou University, Yangzhou, in 2011. His research focuses on sensor network, healthcare system and data mining.

Zhi-Sheng Huo is a Ph.D. candidate in the School of Computer Science and Engineering, Beihang University, Beijing. He received his M.S. degree in computer science from Shenyang Aerospace University, Shenyang, in 2012. His research focuses on big data storage and distributed storage system.



Guang-Jun Qin is a postdoctoral researcher of the School of Electronic and Information Engineering, Beihang University, Beijing. He is a member of CCF. His main research areas are computer architecture and safety critical computer system.



Ming-Fa Zhu is a professor of the School of Computer Science and Engineering, Beihang University, Beijing. He is a senior member of CCF. His main research areas are computer architecture, parallel computing, high performance computer system and network, and artificial intelligence.

Long-Bing Zhang is an associated professor of Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing. He is the associate director of Research Center for Microprocessor, ICT, CAS, Beijing. He received his Ph.D. degree in computer architecture from University of Science and Technology of China, Hefei, in 2002. His research focuses on microprocessor design and parallel processing.

Rui Liu is an associated professor of the School of Computer Science and Engineering, Beihang University, Beijing. His main research interests include e-Science, data management and data mining, and web information retrieval.

Xiang Wang is a professor of the School of Electronic and Information Engineering, Beihang University, Beijing. His main research areas are very large scale integration, micro-nano system, genetic circuits and aerospace information networks.

Appendix

A.1 MPRM

Any n-variable Boolean function may be represented canonically in a sum-of-products (SOP) form as (A1):

$$f(x_{n-1}, x_{n-2}, ..., x_0) = \sum_{i=0}^{2^n-1} a_i m_i, \qquad \text{(A1)}$$

where Σ is the OR operator, m_i are the minterms which can be expressed as $m_i = \dot{x}_{n-1}\dot{x}_{n-2}...\dot{x}_0$; if $i_j = 0$, then $\dot{x}_j = \overline{x_j}$, and if $i_j = 1$, then $\dot{x}_j = x_j$.

a_i is the coefficient of minterms, $a_i = 1$ or 0 represents the presence or absence of minterms, respectively. OR can be replaced by XOR if all the variables are present in every term of (A1). Alternatively, the function can be expressed by MPRM as follows:

$$f^p(x_{n-1}, x_{n-2}, ..., x_0) = \oplus \sum_{i=0}^{2^n-1} b_i \pi_i, \qquad \text{(A2)}$$

where $\oplus\Sigma$ denotes the modulo-2 addition, $\pi_i = \dot{x}_{n-1}\dot{x}_{n-2}...\dot{x}_0$ represents the product terms of the MPRM. $b_i \in \{0, 1\}$ represents whether π_i appears in the function or not. $p = (p_{n-1}p_{n-2}...p_0)$ is polarity. $i = (i_{n-1}i_{n-2}...i_0)$ is subscript. The relationship among $\dot{x}_j$, p_i and i_j can be expressed in Table A1.

Table A1. Value of $\dot{x}_j$

	$i_j = 0$	$i_j = 1$
$p_i = 0$	$\dot{x}_j = 1$	$\dot{x}_j = x_j$
$p_i = 1$	$\dot{x}_j = 1$	$\dot{x}_j = \overline{x_j}$
$p_i = 2$	$\dot{x}_j = \overline{x_j}$	$\dot{x}_j = x_j$

In MPRM, each variable can appear to be true, complemented or both at the same time. The polarity of MPRM can be represented by replacing each variable by 0, 1, or 2 depending on whether the variable is used in true, complement or mixed, respectively. When a variable is used in true(complemented), it can be replaced by 0(1). When a variable is present in both true and complemented forms, it can be replaced by 2. Therefore, for an n-variable MPRM, it has 3^n different polarities.

The polarity will directly determine the form of expression, and thus influence circuit performance. Consequently, the polarity optimization of RM circuits is to search the best polarity, which corresponds to an optimal circuit performance, from a particular polarity space.

A.2 ISMPRM

An n-variable incompletely specified Boolean function may be represented canonically in an SOP form as (A3):

$$f(x_{n-1}, x_{n-2}, ..., x_0) = \sum_{i=0}^{2^n-1} a_i m_i + \sum_{i=0}^{2^n-1} d_i m_i, \qquad \text{(A3)}$$

where Σ is OR operator, m_i is the minterm which can be expressed as $m_i = \dot{x}_{n-1}\dot{x}_{n-2}...\dot{x}_0$. a_i is the coefficient of minterms, $a_i = 1$ or 0 represents the presence or absence of minterms, respectively. d_i is the coefficient of the don't care terms, $d_i = 1$ or 0 represents the presence or absence of don't care terms, respectively.

Accordingly, ISMPRM can be expressed as follows:

$$f^p(x_{n-1}, x_{n-2}, ..., x_0) = \oplus \sum_{i=0}^{2^n-1} b_i e_i \oplus \sum_{i=0}^{2^n-1} d'_i e_i. \qquad \text{(A4)}$$

Compared with (A2), (A4) introduces $\sum_{i=0}^{2^n-1} d'_i e_i$, $d'_i \in \{0, 1\}$ represents whether the don't care term e_i appears in MPRM or not.

For an n-variable ISMPRM with r don't care terms, whether or not each don't care term is to be written to the expression can be represented by a string of binary numbers $< b_{r-1}b_{r-2}...b_i...b_0 >$ (called allocation of don't care terms). $b_i = 0$ represents the don't care term d_i is not written to the expression, while $b_i = 1$ represents d_i is written to the expression. The allocation of don't care terms does not affect the circuit function, but affects the circuit structure, and thus affects circuit performance. Consequently, the RM circuits can be optimized by determining the allocation of don't care terms properly.