ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Engineering Journals
Available on IEEE Xplore®

Research Article
https://doi.org/10.1631/ENG.ITEE.2026.0146

Multiobjective optimization for mixed polarity Reed–Muller logic circuits based on a multi-strategy quantum wolf pack algorithm

Chengcheng CHEN[1], Yuhao ZHOU[2], Juncai HE[3], Dongge QIN[4], Zhanhui SHI[5]✉, Wei ZHANG[6], Jicong DUAN[7], Zhenxue HE[3], Limin XIAO[8]

[1]*School of Future Transportation, Guangzhou Maritime University, Guangzhou 510725, China*
[2]*School of Automation and Electrical Engineering, Linyi University, Linyi 276000, China*
[3]*Key Laboratory of Agricultural Big Data of Hebei Province, Hebei Agricultural University, Baoding 071001, China*
[4]*School of Integrated Circuits, Southeast University, Nanjing 210000, China*
[5]*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China*
[6]*School of Artificial Intelligence and Big Data, Suzhou University, Suzhou 234000, China*
[7]*School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212003, China*
[8]*School of Computer Science and Engineering, Beihang University, Beijing 100191, China*

Abstract: Area and power need to be considered at all stages of integrated circuit design. Area and power cooperative optimization for mixed polarity Reed–Muller (MPRM) circuits aims to find the Pareto non-dominated sets with minimum area and power. Existing multiobjective optimization approaches for MPRM circuits have a long runtime and cannot handle large-scale circuits. In this paper, we propose a multi-strategy quantum wolf pack algorithm (MQWPA), which includes quantum encoding, dual-population strategy, dynamic reverse learning strategy, mutation strategy, and termination criterion. Moreover, based on the proposed MQWPA, we propose an efficient area and power optimization approach (EAPOA), which searches for XNOR/OR-based MPRM circuits with Pareto non-dominated solution sets. Experimental results show that compared with state-of-the-art evolutionary algorithms, MQWPA can search for optimal solutions in multiobjective test problems; compared with commonly used Boolean minimization methods, EAPOA has advantages in area and power consumption. After field-programmable gate array technology and standard cell mapping, by selecting area as the optimal cost implementation, we obtain average improvements in the actual area of 16.33% and 2.27%, respectively.

Key words: Cooperative optimization; Mixed polarity Reed–Muller; Quantum wolf pack algorithm; Pareto non-dominated sets

1 Introduction

With the development of integrated circuits, the size of complementary metal-oxide-semiconductor (CMOS) transistors has reached the nanometer scale; thus, reducing area and power has become the key problem in integrated circuit design (Belot et al., 2021; Yin et al., 2021; Park et al., 2022). An effective way to solve these problems is to simplify the circuits as much as possible to optimize area and power of the circuits. In the process of digital circuit design, the logic level plays an important role in connecting the register transfer level and the physical level, and its electronic design automation tools are the most mature.

Early digital circuit design mostly focused on Boolean logic. In fact, digital circuits can be implemented by Reed–Muller (RM) logic based on XNOR/XOR operations, and RM logic can realize a complete set of arbitrary logic functions (Wang X et al., 2013). In some specific areas of digital circuit design, such as field-programmable gate array (FPGA), multi-valued logic systems, error detection, and coding/decoding theory, RM logic can simplify complex Boolean functions into a more efficient form and help simplify the analysis of the behavior of circuits in the event of failure (Wang X et al., 2013; Abu Lebdeh et al., 2017; Wang L and Xie, 2020). RM logic functions can be divided into fixed polarity RM (FPRM) expressions and mixed polarity RM (MPRM) expressions according to the value form of variables. Polarity determines the representation of the MPRM logic circuits, which in turn affects power and area of the circuits. For a logic function with D input variables, there are 3^D mixed polarities and 2^D fixed polarities, respectively, which correspond to 3^D mixed polarity expressions and 2^D fixed polarity expressions, and

✉ Zhanhui SHI, shizhanhui@zjut.edu.cn
Chengcheng CHEN, https://orcid.org/0000-0002-6014-5135
Zhanhui SHI, https://orcid.org/0000-0003-0770-6690
CLC number: TP18

Received: May 15, 2026; Revision accepted: Aug. 4, 2026;
Crosschecked: Sept. 1, 2026; Published online: Oct. 6, 2026

© The Authors 2026. Published by Zhejiang University Press Co., Ltd.
This is an open access article distributed under the terms of the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

all fixed polarity expressions are contained in the mixed polarity expressions (Xia et al., 2003; Yang M et al., 2011; Zhou et al., 2022b). Therefore, MPRM logic circuits provide a broader range of polarity configurations for optimization than FPRM logic circuits. However, the corresponding optimization process is more complex.

Area and power optimization for XNOR/OR-based MPRM circuits is more difficult. First, after benchmark tests, some XNOR/OR-based logic expressions are more concise and offer greater optimization potential in practical applications (Yu HZ et al., 2013; Abu Lebdeh et al., 2017; Chen HW et al., 2021; Wagle et al., 2021). However, most research focused on XOR circuits (Wang X et al., 2015; Das and Pradhan, 2018; Zhou et al., 2024; Wu et al., 2025) and Exclusive-OR Sum-of-Products (ESOP)-based FPRM circuits (Drechsler et al., 1996, 2000; Kalay et al., 2000; Qin et al., 2023). These approaches can be applied to circuits with multiple input variables, but most rely on conventional optimization strategies and do not explicitly consider computation time. In addition, the conversion to XNOR logic by De Morgan's law would be more time-consuming. Second, the above works are all based on single-objective optimization (i.e., area or power optimization). However, with the advent of the post-Moore era, the main frequency as well as the transistor density of each new generation has almost doubled, so power and area need to be considered in the logic synthesis stage. Finally, existing approaches for optimizing area and power of XNOR/OR-based MPRM circuits are inefficient and cannot handle large-scale circuits with multiple input variables (Yu HZ et al., 2017; Yan et al., 2020). Yu HZ et al. (2017) proposed a multiobjective particle swarm optimization algorithm based on the weighted sum method (WSM) for area and power optimization. However, the flight speed of the particles easily exceeds the boundary value and it is difficult to obtain the Pareto optimal frontier solution set by the WSM (Yan et al., 2020). An area and power optimization approach based on Pareto non-dominated sorting genetic algorithm was proposed (He et al., 2017). However, the objectives are mutually constrained by decision variables and the mutation control parameters are sensitive, making the approach time-consuming. In addition, the area and power optimization approaches described above are less effective in dealing with large-scale circuits with multiple input variables. Zhou et al. (2022a) proposed an improved bacterial foraging algorithm (IBFA) for area and power optimization of MPRM circuits. However, this approach still suffers from several limitations. First, it employs a conventional ternary encoding scheme combined with a complex global experience-driven search strategy, which limits the improvement in optimization efficiency for large-scale circuits. Second, the designed convergence mechanism mainly targets bi-objective optimization problems and lacks scalability to many-objective optimization scenarios. Furthermore, the effectiveness of this method has not been validated through practical implementation flows, such as standard cell mapping and FPGA-based technology mapping. Therefore, developing an optimization framework that can simultaneously enhance search efficiency and maintain global exploration capability in high-dimensional discrete polarity spaces remains an important research challenge.

Area and power optimization for XNOR/OR-based MPRM logic circuits is a combinatorial optimization problem. Most research focuses on single-objective optimization, which is inefficient in dealing with large-scale circuits and cannot converge to the true Pareto optimal frontier. To solve these problems, we propose an area and power optimization approach for MPRM logic circuits based on a multi-strategy quantum wolf pack algorithm (MQWPA). Compared with existing area and power optimization approaches, our main contributions are as follows:

1. We propose an MQWPA, which follows a simplicity- and efficiency-oriented philosophy. Rather than introducing complex hybrid mechanisms, each component is designed to serve a specific functional role in ternary MPRM polarity optimization, ensuring low computational overhead and scalability.

2. We propose an efficient area and power optimization approach (EAPOA), which uses MQWPA to search Pareto optimal solution sets with minimum area and power. The effectiveness of MQWPA and EAPOA is verified by comprehensive experiments.

3. We conduct detailed experiments on the validity of MQWPA and EAPOA, including parameter verification, comparison on the IEEE Congress on Evolutionary Computation (CEC) test sets, verification of EAPOA on area and power, verification of EAPOA on convergence, verification of EAPOA on execution time, technology mapping, and FPGA mapping. Experimental results show that our proposed methods are superior to the current state-of-the-art multiobjective optimization approaches and can be used as a promising logic synthesis method. The source code and data are publicly available at https://gitee.com/Yuhao_Zhou.

2 Problem formulation

2.1 Area estimation model

In XNOR/OR-based MPRM logic circuits, a D-variable logic function for polarity P can be represented by the MPRM expression:

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

where the ternary representation of P is $(p_{D-1}p_{D-2}\cdots p_0)$, and the binary form of subscript i can be expressed as $i=(i_{D-1}i_{D-2}\cdots i_0)$. $d_i \in \{0, 1\}$, with 0 and 1 corresponding to the presence or absence of maxterms, respectively. "$\odot\Pi$" is an XNOR operator. $S_i = \bar{x}_{D-1,i} + \bar{x}_{D-2,i} + \cdots + \bar{x}_{0,i}$. x_i denotes the i^{th} input variable, and $\bar{x}_i$ denotes the complement of x_i. The multi-input XNOR and OR terms need to be decomposed into two-input OR terms and two-input XNOR terms, expressed as

$$\text{XNOR_SUM} = \sum_{i=0}^{2^D-1} \bar{d}_i - 1, \tag{2}$$

$$\text{OR_SUM} = \sum_{i=0}^{2^D-1} \left(\bar{d}_i \left(\sum_{k=0}^{D-1} \bar{i}_k - 1 \right) \right), \tag{3}$$

where XNOR_SUM and OR_SUM represent the sum of XNOR terms and OR terms in the expressions, respectively. i_k denotes the k^{th} bit in the binary representation of $i=(i_{D-1}i_{D-2}\cdots i_0)$, and $i_k \in \{0, 1\}$. Finally, the XNOR/OR-based mixed polarity RM logic circuits' area E_{area} can be expressed as

$$\begin{aligned} E_{\text{area}} &= \text{OR_SUM} + \text{XNOR_SUM} \\ &= \sum_{i=0}^{2^D-1} \bar{d}_i \sum_{k=0}^{D-1} \bar{i}_k - 1 . \end{aligned} \tag{4}$$

2.2 Power estimation model

In digital CMOS circuits consisting of n_g gates, the total dynamic power E_{pow} can be expressed as

$$E_{pow}=\frac{1}{2}V_{dd}^2 f_{clk}\sum_{i=1}^{n_g}C_L^i E_{s\omega d}^i, \tag{5}$$

where V_{dd} represents the voltage, f_{clk} is the clock frequency, and C_L^i is the output load capacitance of gate i. $E_{s\omega d}^i$ (referred to as switching activity) is the number of transitions per cycle at gate i. In the logic synthesis stage, only $E_{s\omega d}^i$ is controllable and is directly proportional to E_{pow}, so $E_{s\omega d}^i$ is used as an approximate indicator of E_{pow} at the logic synthesis stage. The switching activity of gate circuits can be obtained from the output signal probability $\Pr(\xi_i)$ and is expressed as

$$E_{s\omega d}^i=2\cdot\Pr(\xi_i)[1-\Pr(\xi_i)], \tag{6}$$

$$\Pr(\xi_i)_{OR}=\Pr(x)+\Pr(y)-\Pr(x)\cdot\Pr(y), \tag{7}$$

$$\Pr(\xi_i)_{XNOR}=1+2\cdot\Pr(x)\cdot\Pr(y)-\Pr(x)-\Pr(y), \tag{8}$$

where ξ_i denotes the output signal of the i^{th} gate, and $\Pr(\xi_i)$ denotes the probability that the output signal of the i^{th} gate is logic 1. $\Pr(x)$ and $\Pr(y)$ denote the probabilities that the two input signals x and y are logic 1. Eqs. (7) and (8) are the output signal probabilities of the two-input OR gate and the two-input XNOR gate after logical decomposition, respectively. Power consumption is approximated using switching activity at the logic synthesis stage. This metric mainly reflects dynamic switching behavior and is widely used as an early-stage proxy for power estimation.

2.3 Multiobjective optimization problem

Let $N(U)$ be the simplified and decomposed logic circuit and U be the set of two-input gates. The numbers of two-input OR gates and two-input XNOR gates in U are counted as m_{OR} and m_{XNOR}, respectively; the output signal probabilities of two-input OR gates and two-input XNOR gates are counted as l_{OR} and l_{XNOR}, respectively. Then, the multiobjective optimization problems of MPRM circuits can be described as

$$A(p)=m_{OR}+m_{XNOR}, \tag{9}$$

$$E(p)=\sum_{y\in U}2l_{OR,y}+\sum_{z\in U}2l_{XNOR,z}, \tag{10}$$

where p is the circuit polarity, $A(p)$ performs the circuit area estimation, representing the total number of all two-input gates of the polarity P in the MPRM circuit, and $E(P)$ performs the circuit power estimation and represents the total switching activities of polarity P in the MPRM circuit. In the MPRM circuits, area and power often show inconsistent change trends, and may even conflict. It is often difficult to achieve comprehensive optimization of multiple objectives by using WSM to set a single weight. Therefore, the multiobjective decision function for the polarity optimization of the MPRM circuit is set as

$$\min_p(F(p))=\min_p(A(p),E(p)). \tag{11}$$

2.4 Wolf pack algorithm

Currently, many intelligent algorithms are used to deal with expensive optimization problems, thanks to the fact that the evolutionary mechanisms of intelligent algorithms are suitable for high-dimensional optimization. The wolf pack algorithm (WPA) was first proposed by Yang CG et al. (2007). WPA simulates the predatory behavior of wolves and their prey distribution. WPA has high precision and strong robustness, and is suitable for solving complex functions with multiple peaks (Xing et al., 2019; Xu et al., 2021). Chen YB et al. (2018) first proposed a modified two-part wolf pack algorithm (MTWPA) for a multiobjective combinatorial optimization problem, which has excellent performance in solving the problem of cooperative task assignment of multiple unmanned aerial vehicles (UAVs). The main operators of WPA are described as follows:

1. Wandering behavior. S_num wolves with the largest objective function value in the solution space except for the head wolf are regarded as detective wolves. If the current fitness value Y_i of detective wolf i is larger than that of the head wolf Y_{lead}, then $Y_{lead}=Y_i$; if Y_i is smaller than Y_{lead}, then the detective wolf moves once in h directions in steps of $step_a$, calculates Y_i at this time, and returns to the original position. The position of detective wolf i in the d-dimensional space after moving to the κ ($\kappa=1, 2, \cdots, h$) directions is

$$x_{i,d}^{\kappa}=x_{i,d}+\sin(2\pi\kappa/h)\cdot step_a^d, \tag{12}$$

where $x_{i,d}$ indicates the current position of the i^{th} detective wolf in the d-dimensional space.

2. Running behavior. If the prey taste Y_i perceived by fierce wolf i is greater than Y_{lead}, then $Y_{lead}=Y_i$ and the fierce wolf initiates the calling behavior. If Y_i is smaller than Y_{lead}, then fierce wolf i continues to attack until its distance from the head wolf becomes smaller than the threshold d_{near}, which is defined as follows:

$$d_{near}=\frac{1}{D\omega}\sum_{d=1}^{D}\left|\max_d-\min_d\right|, \tag{13}$$

where D is the dimension of the solution space, and ω ($0<\omega<1$) is the distance determination factor. $\max_d$ and $\min_d$ denote the upper and lower bounds of the d-dimensional search space, respectively.

3. Sieging behavior. The fierce wolves unite with the detective wolves to round up the prey and capture it. For the k-generation wolves, the position of prey in the d-dimensional space is G_d^k, and the sieging behavior of wolves is expressed as

$$x_{i,d}^{k+1}=x_{i,d}^k+\lambda\cdot step_c^d\cdot\left|G_d^k-x_{i,d}^k\right|, \tag{14}$$

where λ ($0<\lambda<1$) is a uniformly distributed random number.

3 Overview of the MQWPA

The polarity encoding code consists of 0, 1, and 2 and is constrained to be a combinatorial optimization problem. However, WPA is proposed based on solving single-objective-based continuous optimization problems and cannot be directly used to solve multiobjective discrete combinatorial optimization problems. This means that WPA cannot fully use the information carried by the individual dimensions in solving the polarity optimization problems, and it cannot fully exploit the best performance. Therefore, to solve the above problems, we propose the MQWPA to solve multiobjective optimization problems. The design of MQWPA follows a simplicity- and

efficiency-oriented philosophy. MQWPA uses quantum bits to encode individuals, quantum superposition states and quantum collapse to express information, and quantum rotation gates to update individuals, which gives it a different search mechanism. Therefore, quantum coding is suitable for solving high-dimensional combinatorial optimization problems, which benefit from the discrete optimization space of quantum coding. MQWPA includes quantum encoding, dual-population strategy, dynamic reverse learning strategy (DRLS), mutation strategy, and termination criterion.

3.1 Main framework

The combination of quantum theory and swarm intelligence algorithms can increase population diversity, which is one of the hot topics in optimization theory research and application (Yi et al., 2018; He et al., 2021; Zhou et al., 2025). In addition, the effective combination of quantum theory and evolutionary algorithms can accelerate the convergence rate of the algorithm. We introduce quantum computing in WPA to deal with high-dimensional combinatorial optimization problems. The positions of individuals are encoded using quantum bit probability amplitudes, the individuals are updated using quantum rotation gates, DRLS is employed to improve the global search ability of the algorithm, and the mutation strategy is used to introduce individual mutations. In addition, the dual-population strategy realizes the parallel search of populations, thus improving the local search ability of the algorithm.

The overall framework of MQWPA is described in Algorithm 1. Let the population size be n, the number of iterations be T, and the dimensionality of the problem be D. In each iteration, MQWPA mainly consists of three parts: (1) fitness evaluation and non-dominated sorting, requiring $O(nD)$; (2) quantum rotation update and population evolution, requiring $O(nD)$; (3) DRLS and mutation operations applied to all individuals, requiring $O(nD)$. Therefore, the overall time complexity of MQWPA is $O(TnD)$. Since the dual-population strategy only splits the population into two independent subgroups, it does not change the asymptotic complexity but improves practical parallel efficiency.

Algorithm 1 Framework of the MQWPA

Input: Population size n and Pareto non-dominated solution set N;
Output: N;

```
Randomly generate the initial population;
for i=1: n do
    N is calculated according to the fitness value;
end for
The optimal position of the current best individual is set as the head wolf;
Divide wolves into two subpopulations according to the
dual-population strategy;
for t=0: T (maximum runtime)
    for i=1: n do
        Search for excellent solutions;
        Update the head wolf;
    end for
    for i=1: n do
        Update N;
    end for
    for i=1: n do
        fitness(x_i) = S_area,i / Σ_{i=1}^{n} S_area,i + S_power,i / Σ_{i=1}^{n} S_power,i;
        Calculate the normalized fitness values;
        Use DRLS to produce the next generation;
        Use the mutation strategy for all individuals;
    end for
end for
Output N;
```

3.2 Quantum encoding

In quantum algorithms, quantum bits are based on the concept of bit and superposition state (i.e., the quantum bit is the smallest unit of information storage); its value can be 0 or 1 and any superposition state of both (Tsvelikhovskiy et al., 2026). A quantum bit is represented as

$$|\Psi\rangle=\alpha|0\rangle+\beta|1\rangle, \tag{15}$$

where $|\Psi\rangle$ is the quantum superposition state, and α and β are complex numbers related to the probability amplitude of the corresponding state. $|\alpha|^2$and $|\beta|^2$ are the probabilities that the bit is 0 and 1, respectively, and the normalization condition $|\alpha|^2+|\beta|^2=1$ is satisfied. Let α=cosθ and β=sinθ. Quantum bits can be expressed as $[\cos\theta, \sin\theta]^{\mathrm{T}}$, where θ is the phase of the quantum bits. A D-bit quantum chromosome can be expressed as follows:

$$\begin{bmatrix}\alpha_1 & \alpha_2 \cdots \alpha_D\\ \beta_1 & \beta_2 \cdots \beta_D\end{bmatrix}. \tag{16}$$

Quantum evolution is an important process to realize quantum computing, which is usually operated by quantum rotation gates, which are used to adjust the state of the quantum bits. Single-phase quantum bit rotation gates are defined as

$$\begin{bmatrix}\alpha_i'\\ \beta_i'\end{bmatrix}=\begin{bmatrix}\cos\theta_i & -\sin\theta_i\\ \sin\theta_i & \cos\theta_i\end{bmatrix}\begin{bmatrix}\alpha_i\\ \beta_i\end{bmatrix}, \tag{17}$$

where $[\alpha_i, \beta_i]^{\mathrm{T}}$ is the i^{th} quantum bit of an individual in the quantum algorithm. θ_i is the rotation angle, whose size and direction are determined according to the preset adjustment strategy. Detailed quantum computation can be found in He et al. (2021). Next, we will focus on introducing quantum theory for WPA.

3.2.1 Wolf location encoding

The ternary quantum encoding is specifically designed to match the three-valued polarity representation in MPRM circuits, making it more suitable than binary encoding schemes. We extend from a two-dimensional Hilbert space $\mathbb{C}^2$ to a three-dimensional Hilbert space $\mathbb{C}^3$. Chromosomes are encoded in polarity through ternary quantum bits, which are represented as

$$|\Psi\rangle=\alpha|0\rangle+\beta|1\rangle+\gamma|2\rangle, \tag{18}$$

where α, β, and γ are complex numbers related to the probability amplitude of the corresponding state.

Each probability amplitude is parameterized in spherical coordinates using two angles to ensure normalization:

$$\begin{cases}\alpha_{i,j}=\cos\theta_{i,j},\\ \beta_{i,j}=\sin\theta_{i,j}\cos\varphi_{i,j}\Rightarrow\alpha_{i,j}^2+\beta_{i,j}^2+\gamma_{i,j}^2=1,\\ \gamma_{i,j}=\sin\theta_{i,j}\sin\varphi_{i,j},\end{cases}\tag{19}$$

where $\theta_{i,j}=\pi\cdot\mathrm{rand}(\cdot)$ and $\varphi_{i,j}=2\pi\cdot\mathrm{rand}(\cdot)$. θ is the rotation angle, and φ is the disturbance angle. In quantum WPA, suppose that in a D-dimensional solution space, the i^{th} individual position can be expressed as

$$\begin{aligned}&\boldsymbol{P}_i^{(0)}=[\alpha_{i,1},\alpha_{i,2},\cdots,\alpha_{i,D}],\\ &\boldsymbol{P}_i^{(1)}=[\beta_{i,1},\beta_{i,2},\cdots,\beta_{i,D}],\\ &\boldsymbol{P}_i^{(2)}=[\gamma_{i,1},\gamma_{i,2},\cdots,\gamma_{i,D}].\end{aligned}\tag{20}$$

3.2.2 Location update

Updates of wolf pack positions are made using quantum rotation gates. The update of the quantum rotation gates is achieved by changing the rotation angle and is defined as

$$\boldsymbol{U}(\theta_{0,1})=\begin{bmatrix}\cos\theta_{0,1}&-\sin\theta_{0,1}&0\\ \sin\theta_{0,1}&\cos\theta_{0,1}&0\\ 0&0&1\end{bmatrix},\tag{21}$$

$$\boldsymbol{U}(\theta_{0,2})=\begin{bmatrix}\cos\theta_{0,2}&0&-\sin\theta_{0,2}\\ 0&1&0\\ \sin\theta_{0,2}&0&\cos\theta_{0,2}\end{bmatrix},\tag{22}$$

$$\boldsymbol{U}(\theta_{0,3})=\begin{bmatrix}1&0&0\\ 0&\cos\theta_{0,3}&-\sin\theta_{0,3}\\ 0&\sin\theta_{0,3}&\cos\theta_{0,3}\end{bmatrix}.\tag{23}$$

The advantage of the above update lies in the ability to maintain the normalized orthogonal matrix and control the probability flow of each pair of states at each angle. For the sake of clarity, we normalize the rotation angle θ to Δ. The selection rule for Δ is

$$\begin{cases}\text{If } s=0,b=1\rightarrow\boldsymbol{U}_{0,1}(+\Delta),\\ \text{If } s=1,b=0\rightarrow\boldsymbol{U}_{0,1}(-\Delta),\\ \text{If } s=0,b=2\rightarrow\boldsymbol{U}_{0,2}(+\Delta),\\ \text{If } s=2,b=0\rightarrow\boldsymbol{U}_{0,2}(-\Delta),\\ \text{If } s=1,b=2\rightarrow\boldsymbol{U}_{0,3}(+\Delta),\\ \text{If } s=2,b=1\rightarrow\boldsymbol{U}_{0,3}(-\Delta),\end{cases}\tag{24}$$

where s represents the value of each dimension of the current individual, and b represents the value of each dimension of the current best individual. Then, the updates for the rotation angle θ and the disturbance angle φ are as follows:

$$\Delta\theta_{i,j}=\theta_{i,j}+\phi_{i,j}^{\theta}\left(\theta_{k,j}-\theta_{i,j}\right),\quad\Delta\varphi_{i,j}=\varphi_{i,j}+\phi_{i,j}^{\varphi}\left(\varphi_{k,j}-\varphi_{i,j}\right),\tag{25}$$

where $\phi_{i,j}$ is a random number between [−1, 1], $k\in[1, N]$, and $k\neq i$. The candidate solution Y_i is obtained by sampling each wolf i according to the following definition, and the leader wolf "lead" is calculated. During the wandering phase, the detective wolves perform exploratory rotation updates based on $Y_i\rightarrow Y_i'$ (Y_i' denotes the exploratory candidate solution generated from Y_i during the wandering phase); during the running phase, the fierce wolves perform calling rotation updates based on $Y_i\rightarrow Y_{\mathrm{g}}$ (Y_{g} denotes the guided candidate solution generated during the running phase); during the sieging phase, the fierce wolves perform sieging rotation updates based on Y_i, $Y_{\mathrm{g}}\rightarrow$lead.

$$\text{lead}=\arg\min_i f(Y_i),\tag{26}$$

where $f(Y_i)$ denotes the fitness value of wolf i evaluated at position Y_i.

Example 1 Given a polarity $\boldsymbol{P}$, the quantum encoding of that polarity is as follows:

$$\boldsymbol{P}=\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}=\begin{bmatrix}0.80\\ 0.30\\ 0.52\end{bmatrix}.\tag{27}$$

Here, $p_0=\alpha^2=0.64$, $p_1=\beta^2=0.09$, and $p_2=\gamma^2=0.2704$. Then, the quantum bit s is measured to $|0\rangle$, namely, s=0; assume that the current dimension of the optimal individual is 2, i.e., b=2. Following the rotation rule and using $\boldsymbol{U}(\theta_{0,2})$ with a small step size Δ=0.20, we obtain $\cos\Delta\approx0.9801$ and $\sin\Delta\approx0.1987$. Therefore, the updated probability amplitude is $p_0'=\alpha^{2\prime}=0.463$, $p_1'=\beta^{2\prime}=0.09$, and $p_2'=\gamma^{2\prime}=0.447$.

3.3 Dynamic reverse learning strategy

The search process in WPA is guided by the head wolf, detective wolves, and fierce wolves to balance exploration and exploitation. However, for high-dimensional discrete ternary polarity optimization problems, maintaining population diversity and avoiding premature convergence are critical challenges due to the highly constrained and non-convex search space. Therefore, DRLS is introduced as an integral component of the overall search mechanism, rather than an isolated operator, to enhance global exploration capability in the ternary polarity space while preserving the structural consistency of the optimization process. DRLS has been proven to have a better ability to jump out of local optima in continuous optimization problems (Long et al., 2020). However, DRLS is based on multi-dimensional space and dynamic changes, so DRLS is more advantageous in combinatorial optimization problems. The DRLS process is described as follows:

Definition 1 Reverse point (Rahnamayan et al., 2008) Suppose that $\boldsymbol{X}=(x_1, x_2, \cdots, x_D)$ is a point in D-dimensional space, $x_j\in[a_j, b_j]$, j=1, 2, ⋯, D. Its reverse point is denoted by $\boldsymbol{X}'=(x_1', x_2', \cdots, x_D')$, where $x_j'=a_j+b_j-x_j$. Here, a_j and b_j denote the lower and upper bounds of the j^{th} decision variable, respectively.

Definition 2 Base point (Yu F et al., 2014) If there are several points $\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_m$ in the D-dimensional space, for any point $\boldsymbol{X}=(x_1, x_2, \cdots, x_D)$ and its opposite point $\boldsymbol{X}'=(x_1', x_2', \cdots, x_D')$, the Euclidean distances from $\boldsymbol{X}$ and $\boldsymbol{X}'$ to $\boldsymbol{o}_i$ (i=1, 2, ⋯, m) are d_i and d_i', respectively. The corresponding scaling factor associated with $\boldsymbol{o}_i$ is defined as $k_i=d_i/d_i'$, and i=1, 2, ⋯, m. The point $\boldsymbol{o}_i$ is referred to as the base point associated with the scaling factor k_i. Then $\boldsymbol{o}_i$ is called the base point of $\boldsymbol{X}$ and $\boldsymbol{X}'$ when k=i.

Example 2 Suppose that there is an individual $\boldsymbol{P}$ with height h, whose projection on the coordinate axis is x^* (x^* is the global optimal individual), the base point is $\boldsymbol{o}$ (i.e., take the base point as the midpoint of $[a, b]$), and a point $\boldsymbol{P}'$ with height h' can be obtained by DRLS, whose projection on the coordinate axis is x'^*. Therefore, the inverse individual x'^* of the global optimal individual x^* can be obtained based on DRLS, as shown in Fig. 1.

In Fig. 1, the global optimal individual reverse learning strategy is obtained as the base point of $\boldsymbol{o}$ to obtain its corresponding inverse point x'^*, which can be derived from DRLS as

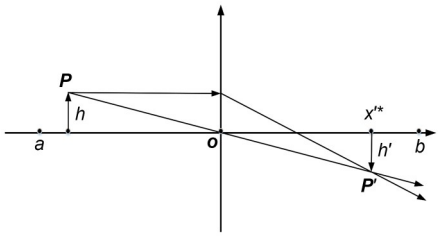


Fig. 1 Geometric illustration of DRLS

$$\frac{\frac{a+b}{2}-x^*}{x'^*-\frac{a+b}{2}}=\frac{h}{h'}. \tag{28}$$

Let $h/h'=k$, where k is called the zoom factor. By transforming Eq. (28), we obtain the formula for the inverse point x'^* as

$$x'^*=\frac{a+b}{2}+\frac{a+b}{2k}-\frac{x^*}{k}. \tag{29}$$

Obviously, when k=1, Eq. (29) can be simplified to

$$x'^*=a+b-x^*. \tag{30}$$

Eq. (30) is the general reverse learning strategy acting on x^* and centered on $[-(a+b)/2, (a+b)/2]$. From Eqs. (29) and (30), it is clear that the general reverse learning strategy is only a special case of DRLS. The new candidates obtained based on the general reverse learning strategy are fixed, and by adjusting k, the new candidates obtained based on DRLS are dynamic, which further enhances the population diversity.

Generally, the reverse learning strategy shown in Eq. (30) can be extended to the D-dimensional space to obtain

$$x_j'^*=\frac{a_j+b_j}{2}+\frac{a_j+b_j}{2k}-\frac{x_j^*}{k}, \tag{31}$$

where x_j^* and $x_j'^*$ are the j^{th} dimension components of x^* and x'^*, respectively, and a_j and b_j are the j-dimensional components of the upper and lower bounds of the decision variable, respectively.

3.4 Mutation strategy

Traditional WPA does not make full use of effective information carried by individuals and does not exchange effective information with the head wolf, which reduces the population diversity and convergence rate. In fact, the head wolf (i.e., global optimal solution) contains a lot of useful information that can be used, especially for high-dimensional combinatorial optimization problems, where each dimension of the individual is crucial.

To solve this problem, we propose a mutation strategy for solving high-dimensional combinatorial optimization problems. Mutation strategy consists of two steps, the first of which is the individual's mutation as shown below:

$$\pi''=\begin{cases}\text{Move}(\pi_j^t, d),\ d=z((\pi_{\text{he}})_j^t-\pi_j^t), \text{if rand}(\cdot)<0.3,\\ \text{Move}(\pi_j^t, d),\ d=z((\pi_{\text{de}})_j^t-\pi_j^t), \text{if } 0.3\le\text{rand}(\cdot)<0.6,\\ \text{Move}(\pi_j^t, d),\ d=z((\pi_{\text{fi}})_j^t-\pi_j^t), \text{otherwise},\end{cases} \tag{32}$$

where π_{he}, π_{de}, and π_{fi} represent the head wolf, detective wolf, and fierce wolf respectively, $(\pi_{\text{he}})_j^t$, $(\pi_{\text{de}})_j^t$, and $(\pi_{\text{fi}})_j^t$ represent the j^{th} dimension of each wolf at the t^{th} iteration respectively, π' is the updated wolf, rand$(\cdot)$ is a random value in the range of 0–1, and z is the control coefficient, set to 1 in this paper. Move(π_j^t, d) means that the j^{th} element in π moves to the right or left by $|d|$ units from the current position, with the symbol "+" meaning to right and "–" to left. If it exceeds the left boundary during the process, it continues to move from the right boundary to the left; the same applies when it exceeds the right boundary.

Example 3 As shown in Fig. 2, these individuals are the head wolf π_{he}, detective wolf π_{de}, and fierce wolf π_{fi} with D dimensions, respectively. Suppose that rand$(\cdot)$=0.4 and d=2 for the first dimension. Then, the detective wolf is shifted by 2 dimensions. After the first step of the mutation strategy is executed in all dimensions, the second step of the mutation strategy is the exchange of information between π_{de} and π_{fi} and the head wolf π_{he}. The information exchange with the head wolf is completed by recording the dimension j that is moved in π_{de} and π_{fi}, and then assigning the j^{th} dimension of the head wolf to the j^{th} dimension of π_{de} and π_{fi}, as shown by the red dashed lines in Fig. 2.

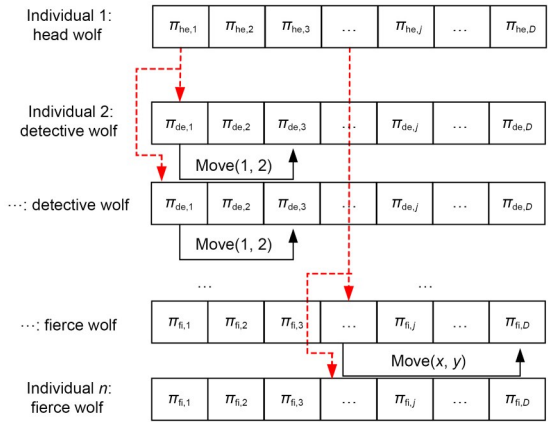


Fig. 2 Overview of the mutation strategy

3.5 Dual-population strategy

The key point of multiobjective optimization problems is to find more Pareto non-dominated solution sets. However, the individual non-dominated computation consumes a lot of time during the operation of the algorithm. To reduce the time consumption of the huge individual fitness computation, an efficient approach is to use a parallel architecture that allows multiple processors to share the time consumption that is unbearable for a single processor. In this paper, the use of a dual-population strategy is another major means of improving the algorithm performance (Chen J et al., 2025). Each subpopulation mainly preserves the solutions corresponding to independently converged decision variables. The subpopulation not only focuses on improving the convergence performance of the solutions but also maintains the diversity of the solution set in the independently converged decision subspace. As shown in Algorithm 2, we give the overall framework of the dual-population strategy. Algorithm 2 only needs to traverse the population once, and the time complexity is $O(n)$.

Algorithm 2 Dual-population strategy

Input: Two subpopulations of size $\frac{n}{2}$ and fitness functions $f(x)$ and $g(x)$;

Output: New individuals i and j;

1. **for** i=1: $\frac{n}{2}$ **do**
2. Calculate $f(i)=1/S_{\text{power},i}$;
3. For each wolf, the location encoding is generated;
4. Wolves begin to evolve according to DRLS and the mutation strategy;
5. **end for**
6. **for** j=1: $\frac{n}{2}$ **do**
7. Calculate $g(j)=1/S_{\text{area},j}$;
8. For each wolf, the location encoding is generated;
9. Wolves begin to evolve according to DRLS and the mutation strategy;
10. **end for**
11. **return** i and j;

In this paper, we deal with the minimization of area and power, so we take S_{area} and S_{power} as fitness functions of subpopulations, respectively. We standardize the quantum encoding scheme, where each subpopulation is an independent convergent subspace that is unaffected between subpopulations in the process of independent evolution. Finally, we normalize the fitness function as described in Algorithm 1. The normalization process serves to enable information exchange between two subpopulations to generate more Pareto non-dominated solution sets.

3.6 Termination criterion

The proposed MQWPA is fundamentally different from IBFA (Zhou et al., 2022a) in terms of its optimization mechanism. IBFA relies on bacterial foraging behaviors combined with conventional ternary encoding, whereas MQWPA introduces a quantum encoding-based search framework to represent and update polarity solutions. This mechanism enables a larger diversity of candidate solutions and improves the exploration capability in the exponentially increasing polarity space. Therefore, MQWPA is not merely an improvement of the previous IBFA, but a new optimization framework that is more suitable for large-scale MPRM circuits, especially when handling multiple objectives while maintaining optimization efficiency and solution quality. The proposed MQWPA integrates quantum rotation update, DRLS, mutation strategy, and dual-population cooperation. At each iteration, the head wolf is selected from the current non-dominated set according to the fitness function. Since area and power are minimized, the Pareto dominance relation ensures that any updated solution replacing the head wolf must not be worse in both objectives. Therefore, the archive of non-dominated solutions is monotonically non-deteriorating during evolution. Moreover, the quantum rotation gate update ensures a probabilistic but bounded search within the feasible solution space, while DRLS introduces reverse sampling that increases the probability of escaping local optima. The mutation strategy further prevents premature convergence by maintaining population diversity. Thus, MQWPA forms a non-decreasing Pareto archive sequence and converges asymptotically toward a stable Pareto frontier under finite search space conditions.

It is common to take the number of iterations as the termination standard of an algorithm, but because of the different evolutionary strategies of each algorithm, it will result in repeated internal recursion of each algorithm in the process of iteration. In quantum computing, time is a more standard reference. In addition, the time superiority of each approach is verified in the follow-up experiments of this paper. Therefore, MQWPA uses a predetermined maximum runtime as a stop condition. It should be emphasized that the proposed MQWPA is a quantum-inspired classical optimization algorithm rather than an algorithm implemented on a real quantum computer. The term "quantum" in this work refers to the use of quantum-inspired representations (e.g., probability amplitude encoding and rotation-based updates), which are simulated on classical hardware.

4 Area and power optimization approach for XNOR/OR-based MPRM logic circuits

4.1 Chromosome encoding

EAPOA mainly includes three kinds of operators oriented to individual coding, including chromosome encoding, DRLS encoding, and mutation strategy encoding. Examples of the three operations are shown below.

Example 4 A control logic unit "con" of three input variables is read and a polarity $\boldsymbol{P}$ with three chromosome bits is defined in the population. The quantum bit s is measured to $|1\rangle$, $|0\rangle$, and $|2\rangle$. The quantum encoding polarity $\boldsymbol{P}$ is encoded as shown in Fig. 3.

Example 5 (DRLS operation for MPRM) As shown in Fig. 4a, suppose that a "dc1" circuit with four dimensions is read and the optimal polarity is defined as f_{best}=(1220). The proposed DRLS is used to calculate the inverse individual. If the DRLS is met, f_{best} is reversed to the child polarity.

Example 6 (Mutation strategy operation for MPRM) As shown in Fig. 4b, suppose that the individual's polarity is (1220), and the polarity bit that needs to be migrated is x_1. The proposed mutation strategy is used to calculate the shifted dimensions. If the information exchange condition is met, the polarity bit will be migrated, thus assigning the j^{th} dimension of the head wolf to the j^{th} dimension of π_{de} and π.

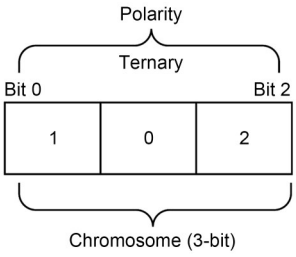


Fig. 3 Polarity encoding

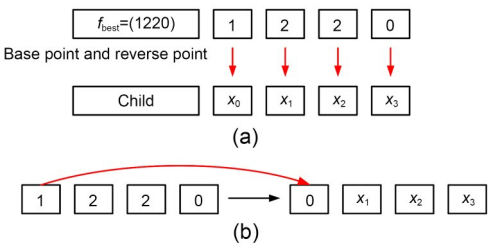


Fig. 4 Overview of MPRM operations: (a) DRLS operation; (b) mutation strategy operation

4.2 Algorithm description

Most research on polarity optimization of RM logic circuits is carried out from single-objective optimization. However, the combined consideration of area and power will certainly help improve the overall performance of the circuits. As described in Section 2, the multiobjective optimization of MPRM logic circuits is a combinatorial explosion problem, and to minimize MPRM logic circuits, we propose an EAPOA. EAPOA uses MQWPA to find the Pareto optimal solution set (i.e., the minimum area and power) in a vast polarity space. EAPOA is shown in Fig. 5, and the steps are described in detail below.

Step 1: read the Boolean logic circuit of the D input variables and initialize EAPOA parameters.

Step 2: wolves (i.e., polarities of MPRM expressions) are encoded as quantum bits, and the dimension of the individual is the number of input variables.

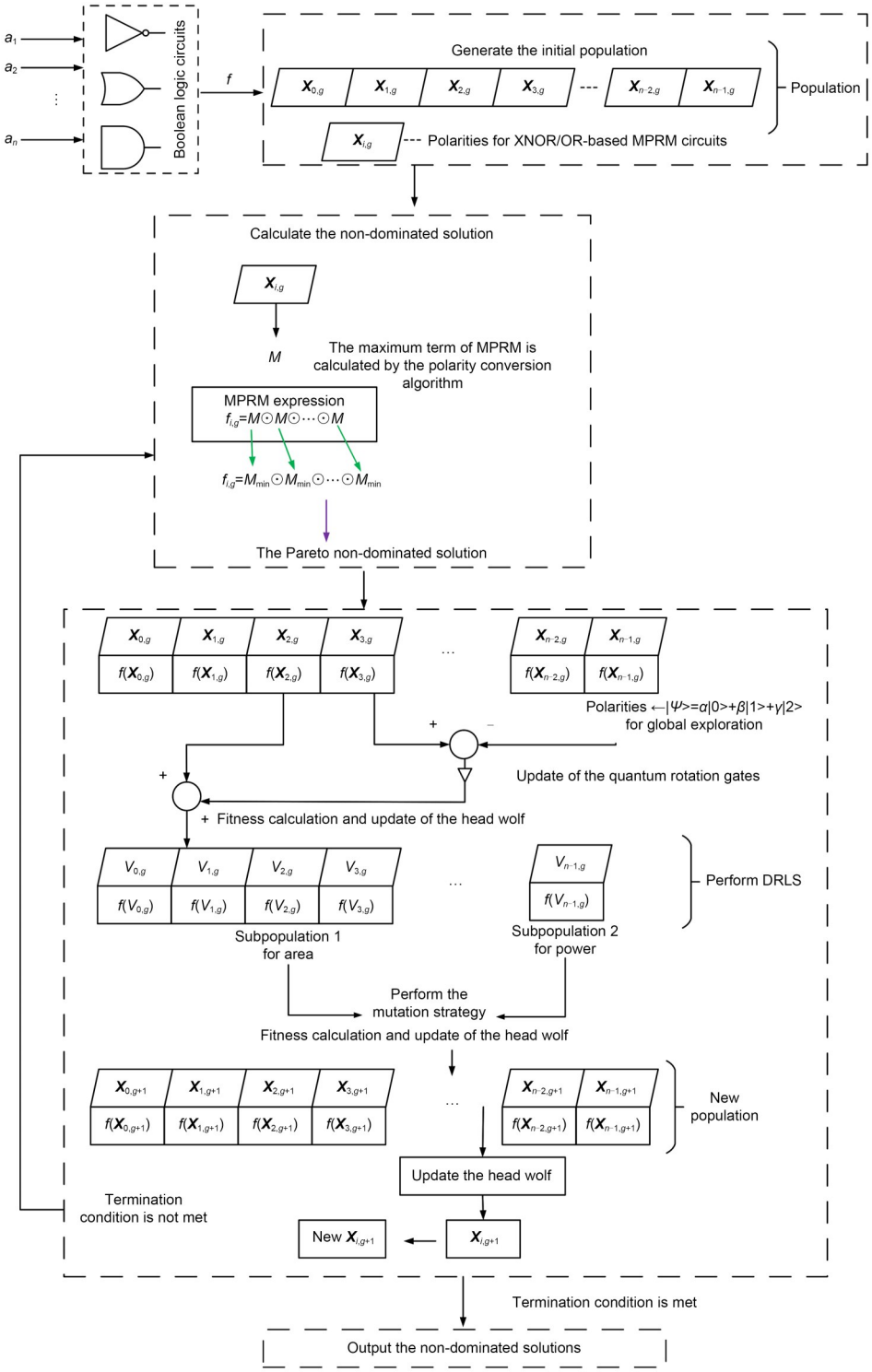


Fig. 5 Overview of the EAPOA

Step 3: convert the Boolean logic circuits into MPRM expressions, and compute the non-dominated solution $\boldsymbol{X}_{i,g}$ at the g^{th} iteration.

Step 4: updates of wolf pack positions are made using quantum rotation gates. The head wolf is selected according to the fitness function, which is defined as

$$\text{fitness}(x_i)=S_{\text{area},i}\Big/\sum_{i=1}^{n}S_{\text{area},i}+S_{\text{power},i}\Big/\sum_{i=1}^{n}S_{\text{power},i}, \tag{33}$$

where fitness(x_i) denotes the fitness value of polarity $\boldsymbol{X}_{i,g}$, $S_{\text{area},i}$ denotes the area of the polarity $\boldsymbol{X}_{i,g}$, and $S_{\text{power},i}$ denotes the power of the polarity $\boldsymbol{X}_{i,g}$. It is worth mentioning that the head wolf must be selected from the non-dominated set.

Step 5: the two subpopulations search in their respective convergent decision subspaces to maintain the diversity of solution sets in the target space and decision subspace.

Step 6: individual $\boldsymbol{X}_i$ in the two subpopulations performs DRLS to obtain $\boldsymbol{X}_i'=(x_1', x_2', \cdots, x_D')$. $\boldsymbol{X}_i'$ provides a basis for the next generation of individual variation and enhances the local search ability of the algorithm.

Step 7: the two subpopulations mix to form a new population and perform the mutation strategy. The purpose of the mutation strategy is to exchange information between individuals and make full use of information from all dimensions of the head wolf.

Step 8: if the termination condition is met, the Pareto non-dominated set is output; otherwise, jump to step 3.

5 Experiments

Both our proposed algorithms (MQWPA and EAPOA) are verified in this section. All experiments are performed using a set of Microelectronics Center of North Carolina (MCNC) benchmarks, and the characteristics of the circuits can be found in Yang S (1991). The proposed EAPOA is run on a computer with a 2.20 GHz Intel processor and 8 GB of RAM, which has 6 physical processing cores and 12 logical threads. It is worth noting that since hardware-level quantum implementation requires specific circuit designs and constraint analyses, we do not consider quantum computers here. All algorithms are compiled using the Visual Studio compiler in C language. Through repeated control variable experiments, EAPOA's parameters are shown in Table 1. We perform detailed comparisons of existing works. To verify the performance of EAPOA in searching for the Pareto optimal solution with minimum area and power, we compare it with the WSM-based area and power optimization approach (Yu HZ et al., 2017) and the non-dominated sorting genetic algorithm-II (NSGA-II)-based area and power optimization approach (He et al., 2017).

Table 1 Parameter settings for EAPOA

Parameter	Value
Total number of wolves	30
Subpopulation size	15
Wolf movement step size	1
Acceptable time (s)	60

5.1 Verification of MQWPA, MTWPA, NBFA, MOEA/D-SCS, and MGSO-BFA on benchmark functions

To test the performance of MQWPA, we select five dual-objective functions for numerical experiments, which are commonly used multiobjective test functions (Zitzler et al., 2000). A detailed description of the test functions is shown in Table 2. To ensure the fairness of the experiments, we not only compare with state-of-the-art multiobjective WPA, but also with other heuristic multiobjective optimization algorithms. Specifically, the comparison algorithms include MTWPA (Chen YB et al., 2018), novel bacterial foraging algorithm (NBFA) (Niu et al., 2013), decomposition-based multiobjective evolutionary algorithm with a self-organizing collaborative scheme (MOEA/D-SCS) (Zhu et al., 2023), and multiobjective glowworm swarm optimization and bacterial foraging algorithm coupling algorithm (MGSO-BFA) (Wang YC et al., 2019). In addition, to further eliminate the influence of experimental environments on the algorithm, the experimental results of MGSO-BFA refer to Wang YC et al. (2019). The inverted generational distance (IGD) index is selected to evaluate the performance of the algorithm, which is mainly used to measure the convergence and distribution of the Pareto optimal solution set. The parameters of the comparison algorithms refer to the publications. The experimental results are shown in Table 3.

Table 3 shows the experimental results of 10 runs, where STD represents the standard deviation. From Table 3, compared with MTWPA, NBFA, MOEA/D-SCS, and MGSO-BFA, MQWPA achieves competitive results on the tested ZDT problems. Specifically,

Table 2 Test function descriptions

Benchmark function	Dim	Objective functions
ZDT1	30	$f_1(x)=x_1;\ f_2(x)=g(x)\left[1-\sqrt{\frac{x_1}{g(x)}}\right];$ $g(x)=1+9\left(\frac{1}{D-1}\sum_{i=2}^{D}x_i\right)$
ZDT2	30	$f_1(x)=x_1;\ f_2(x)=g(x)\left[1-\left(\frac{x_1}{g(x)}\right)^2\right];$ $g(x)=1+9\left(\frac{1}{D-1}\sum_{i=2}^{D}x_i\right)$
ZDT3	30	$f_1(x)=x_1;$ $f_2(x)=g(x)\left[1-\sqrt{\frac{x_1}{g(x)}}-\frac{x_1}{g(x)}\sin(10\pi x_1)\right];$ $g(x)=1+9\left(\frac{1}{D-1}\sum_{i=2}^{D}x_i\right)$
ZDT4	10	$f_1(x)=x_1;\ f_2(x)=g(x)\left[1-\sqrt{\frac{x_1}{g(x)}}\right];$ $g(x)=1+10(D-1)+\sum_{i=2}^{D}\left[x_i^2-10\cos(4\pi x_i)\right]$
ZDT6	10	$f_1(x)=1-\exp(-4x_1);\ f_2(x)=g(x)\left[1-\left(\frac{f_1}{g(x)}\right)^2\right];$ $g(x)=1+9\left(\sum_{i=2}^{D}\frac{x_i}{D-1}\right)^{1/4}$

Dim: dimension

in the ZDT1 and ZDT3 test problems, it can be seen that MQWPA can obtain better non-dominated sets when dealing with a continuous Pareto frontier or a discontinuous Pareto frontier. In the ZDT4 problem, due to the multi-modal characteristics of ZDT4, the convergence of the Pareto frontier of MQWPA is weaker than that of MOEA/D-SCS, but the distribution of the Pareto frontier of MQWPA is better than that of the MOEA/D-SCS. Furthermore, on the ZDT2 and ZDT6 benchmark problems, the proposed MQWPA achieves a performance that is very close to MOEA/D. As a variant of MOEA/D, MOEA/D-SCS incorporates a self-organizing cooperative mechanism to enhance information interaction among subproblems, which is conceptually similar to the multi-strategy cooperation mechanism adopted in MQWPA. Therefore, MOEA/D-SCS is selected as an additional comparison algorithm for continuous optimization problems to ensure a fair and comprehensive evaluation.

Although MOEA/D-SCS demonstrates strong performance on continuous multiobjective problems, MQWPA exhibits superior convergence stability and a more uniform Pareto solution distribution. More importantly, due to its discrete quantum encoding and polarity-based representation, MQWPA can be naturally extended to combinatorial optimization problems without requiring continuous-to-discrete transformation. The test results of the above ZDT test functions are used to make a statistical boxplot on the IGD index. As can be seen from Fig. 6, the convergence distribution of MQWPA is better than those of the comparison algorithms. The distribution is concentrated in a standard range.

To verify the effectiveness of the proposed DRLS, mutation strategy, and dual-population strategy, ablation experiments are conducted. Taking the complete MQWPA as the baseline, three ablation variants are constructed by removing the DRLS, mutation strategy, and dual-population strategy, and are denoted as NDRLS, NMS, and NDPS respectively. The corresponding mechanisms are replaced with a classical opposition-based learning strategy, a conventional mutation operator, and a single-population strategy, respectively, while all other algorithmic procedures and parameter settings are kept unchanged. To ensure the statistical reliability of the results, each algorithm is independently run 10 times on each benchmark circuit. Table 4 reports the mean and standard deviation of the IGD values obtained from the 10 independent runs.

The ablation results show that the complete MQWPA achieves the best overall performance among all compared variants. Although NDRLS slightly outperforms MQWPA on ZDT3, this result is mainly related to the discontinuous Pareto frontier of ZDT3. The DRLS introduces stronger adaptive perturbations to improve global exploration, but such perturbations may occasionally affect local exploitation near discontinuous Pareto regions. Nevertheless, MQWPA still achieves the best overall performance across most benchmark problems, indicating that DRLS is effective in improving the general robustness and search capability of the algorithm.

Compared with NDRLS, MQWPA obtains lower IGD values, indicating that DRLS effectively improves the global exploration ability and helps the population escape from local optima. Compared with NMS, MQWPA shows better convergence stability, which demonstrates that the proposed mutation strategy can enhance information exchange among individuals and reduce premature convergence. Compared with NDPS, MQWPA achieves a more uniform distribution of Pareto solutions, indicating that the dual-population strategy plays an important role in maintaining diversity and balancing convergence and distribution.

5.2 Verification of NSGA-II, WSM, and EAPOA on area and power

To demonstrate the effectiveness of EAPOA in searching for XNOR/OR-based MPRM logic circuits with minimum area and power,

Table 4 Ablation studies on ZDT problems

Benchmark function	Index	NDRLS	NMS	NDPS	MQWPA
ZDT1	IGD	0.014 195	0.014 415	0.012 332	**0.010 841**
	STD	0.009 051	0.004 427	0.004 485	0.003 939
ZDT2	IGD	0.040 069	0.040 076	0.040 163	**0.036 719**
	STD	0.027 379	0.027 111	0.027 753	0.024 515
ZDT3	IGD	**0.072 821**	0.083 328	0.083 321	0.077 877
	STD	0.064 157	0.072 362	0.071 617	0.065 457
ZDT4	IGD	0.379 739	0.383 211	0.403 125	**0.305 410**
	STD	0.119 770	0.091 329	0.095 533	0.046 021
ZDT6	IGD	0.081 672	0.075 368	0.086 701	**0.071 509**
	STD	0.062 783	0.050 342	0.064 559	0.049 905

The best results are in bold

Table 3 IGD and STD results of five algorithms on ZDT problems

Benchmark function	Index	Value				
		MQWPA	MTWPA	NBFA	MOEA/D-SCS	MGSO-BFA
ZDT1	IGD	**1.0841E−02**	7.6404E−02	2.5234E−01	3.0598E−02	4.3310E−01
	STD	3.9385E−03	1.1564E−01	2.4621E−01	2.5631E−02	2.3720E+00
ZDT2	IGD	3.6719E−02	1.0923E−01	2.9778E−01	**3.4816E−02**	4.5950E−01
	STD	2.4515E−02	9.4008E−02	1.1852E−01	2.6270E−02	2.3168E+00
ZDT3	IGD	**7.7877E−02**	1.2555E−01	2.0945E−01	8.0863E−02	3.0680E−01
	STD	6.5457E−02	9.9981E−02	1.1238E−01	6.6560E−02	2.7706E+00
ZDT4	IGD	3.0541E−01	3.4454E−01	2.9714E−01	**2.6899E−01**	4.8980E−01
	STD	4.6021E−02	9.2099E−02	8.2923E−02	1.2379E−01	2.6827E+00
ZDT6	IGD	7.1509E−02	1.5410E−01	7.9494E−02	**7.0695E−02**	4.0140E−01
	STD	4.9905E−02	1.3730E−01	7.9270E−02	7.8970E−02	2.1985E+00

The best results are in bold

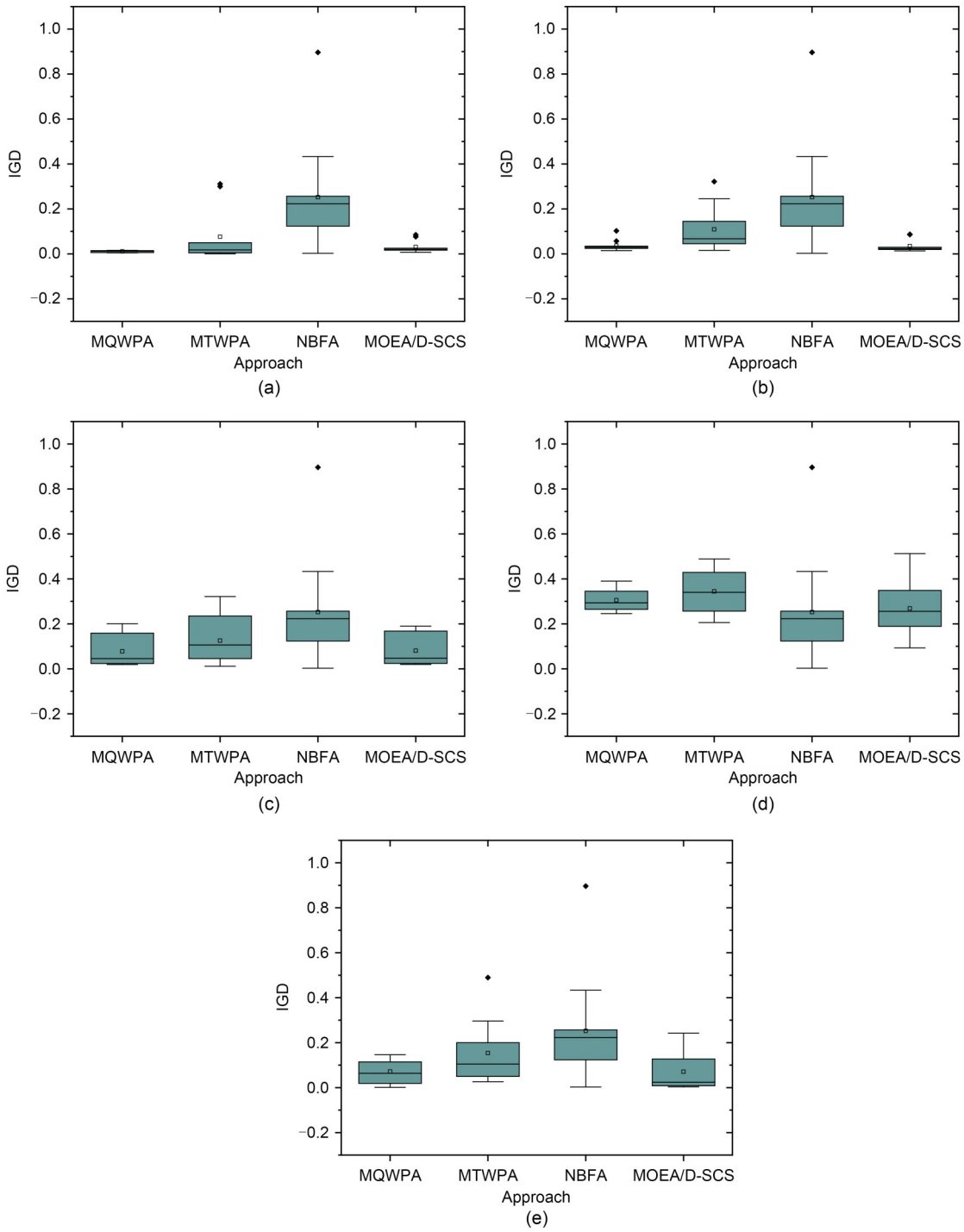


Fig. 6 Statistical boxplots on ZDT test functions: (a) ZDT1; (b) ZDT2; (c) ZDT3; (d) ZDT4; (e) ZDT6

we compare existing works, i.e., WSM (Yu HZ et al., 2017; Yan et al., 2020) and NSGA-II (He et al., 2017), both of which are commonly used area and power optimization approaches. Moreover, the corresponding published studies provide detailed circuit mapping procedures and complete experimental configurations, which ensure reproducibility and make them suitable as fair and representative baselines for comparison in MCNC benchmark circuits. During the iterative process, the chromosome update is also based on the cross-exchange of each dimension. Neither of the above two methods uses quantum coding. The experimental circuits are randomly selected, and the experimental results are shown in Table 5. Here, scale represents the size of the test circuits (that is, the number of polarities of the original test circuits). Area is uniformly determined by the number of gates in a two-input logic gate, and power is determined by the process-independent switching activity $E^i_{s\omega d}$. The scale for a test circuit with D input variables is 3^D. In this paper, we test the circuit input variables up to D=65.

From Table 5, we can see that EAPOA can search for a smaller area and $E^i_{s\omega d}$, and can search for more Pareto non-dominated solution sets. For example, the optimization rates of area and power of EAPOA in table5 circuit compared to WSM are 30% and 23%, respectively; by adopting equal-weighted synthesis fitness, the

optimization rates of area and E^i_{sod} of EAPOA in table5 circuit compared to NSGA-II are 37% and 12%, respectively. EAPOA can find multiple optimal solutions with good distribution and boundaries, and the Pareto optimal solutions dominate the solution sets searched by other comparison approaches. The experimental results can be explained by the following reasons:

Table 5 Experimental results on area and power

Circuit	Scale	WSM		NSGA-II		EAPOA	
		Area	E^i_{sod} (Hz)	Area	E^i_{sod} (Hz)	Area	E^i_{sod} (Hz)
rd53	3^5	19	31.44	19	31.44	19	31.44
				21	30.23	21	30.23
con1	3^7	21	37.78	21	36.59	21	36.59
5xp1	3^7	37	61.39	37	61.39	37	61.39
				41	59.32	41	59.32
rd84	3^8	62	74.85	55	98.00	55	98.00
				59	96.50	58	96.99
				61	91.56	59	96.65
sao2	3^{10}	239	343.10	219	375.90	179	303.80
				230	313.25	180	295.95
br1	3^{12}	160	279.91	154	264.12	133	313.27
						140	266.31
						154	264.12
br2	3^{12}	107	189.34	107	255.19	78	145.18
		110	170.42	111	239.04	80	142.07
				150	196.23		
amd	3^{14}	402	614.26	385	600.57	373	729.32
						376	700.11
						385	598.62
alu4	3^{14}	483	788.33	479	805.52	479	805.52
				539	765.21	524	733.21
table5	3^{17}	71	128.90	71	128.90	50	98.62
				79	112.30		
bcc	3^{26}	97	145.18	57	153.03	57	153.03
		102	128.12	69	140.01	61	73.42
		113	110.26	85	132.12	63	66.55
				89	113.08	64	62.21
						67	60.02
bcd	3^{26}	125	247.09	72	298.87	70	199.69
				86	222.33	72	131.52
						113	101.16
e64	3^{65}	1526	2300.21	1484	1999.35	285	870.21
		1963	2010.15	1869	1898.32	321	599.30
						335	484.21

1. Area and power optimization of MPRM logic circuits is a high-dimensional combinatorial optimization problem. The previous approaches did not make full use of the effective information carried by individual dimensions, resulting in poor population diversity. MQWPA adopts the dual-population strategy and normalizes fitness functions to enhance the parallel search ability of the population, and maintains the diversity of the solution set in the independently converged decision subspace.

2. In WPA, all wolves gather at the end of iterations, which makes the algorithm premature. DRLS and the mutation strategy make the algorithm converge to the true Pareto optimal frontier by mapping the promising search area with base point and reverse point and by discovering the potential excellent solution and using the information carried by the head wolf.

5.3 Verification of NSGA-II, WSM, and EAPOA on execution time

To demonstrate the time consumption performance of EAPOA, the average time of 10 runs is taken as experimental data.

From Table 6, we can see that EAPOA is faster than the compared approaches in most test circuits. The saving rates of EAPOA are not significant for small-scale circuits, such as rd53 and con1, because of the need to solve problems such as initial population generation and memory allocation. However, for large-scale circuits with many input variables, the superiority of EAPOA is outstanding. Specifically, compared with WSM and NSGA-II, the maximum time-saving rates of EAPOA are 53.22% and 71.69%, respectively. The experimental results can be explained by the following reasons:

1. The dual-population strategy enhances the parallel search ability of the population and can easily jump out of the local optima in the early stage of evolution. The mutation strategy uses the dimensional information of the head wolf to mine potential excellent solutions.

2. DRLS can fully mine the information of the solution space to enable the wolves to explore a promising direction, avoiding unnecessary time waste. In addition, the effective combination of quantum theory and evolutionary algorithm speeds up the convergence of the algorithm, which also confirms that our prediction is correct.

5.4 Verification of NSGA-II, WSM, and EAPOA on convergence

To demonstrate the convergence performance of EAPOA more intuitively, we randomly select test circuits of different scales, and plot the convergence curves with area, power, and number of iterations. As shown in Fig. 7, EAPOA can exhibit excellent convergence performance early in the iterations. The main reasons for

Table 6 Experimental results on time

Approach	Time (s)												
	rd53	con1	5xp1	rd84	sao2	br1	br2	amd	alu4	table5	bcc	bcd	e64
WSM	5.86	4.32	5.68	10.51	10.71	8.23	4.75	7.78	16.90	20.12	54.83	16.48	500.14
NSGA-II	5.72	6.51	5.55	12.01	15.94	5.96	5.01	13.91	15.39	25.22	64.91	40.12	659.30
EAPOA	6.22	7.01	6.11	9.56	10.01	3.85	4.06	6.05	10.22	16.33	29.99	11.36	300.10
Save1	−6.14%	−62.27%	−7.57%	9.04%	6.54%	53.22%	14.53%	22.24%	39.53%	18.84%	45.30%	31.07%	40.00%
Save2	−8.74%	−7.68%	−10.09%	20.40%	37.20%	35.40%	18.96%	56.51%	33.59%	35.25%	53.80%	71.68%	54.48%

Save1 and Save2 are the time-saving rates of EAPOA relative to WSM and NSGA-II

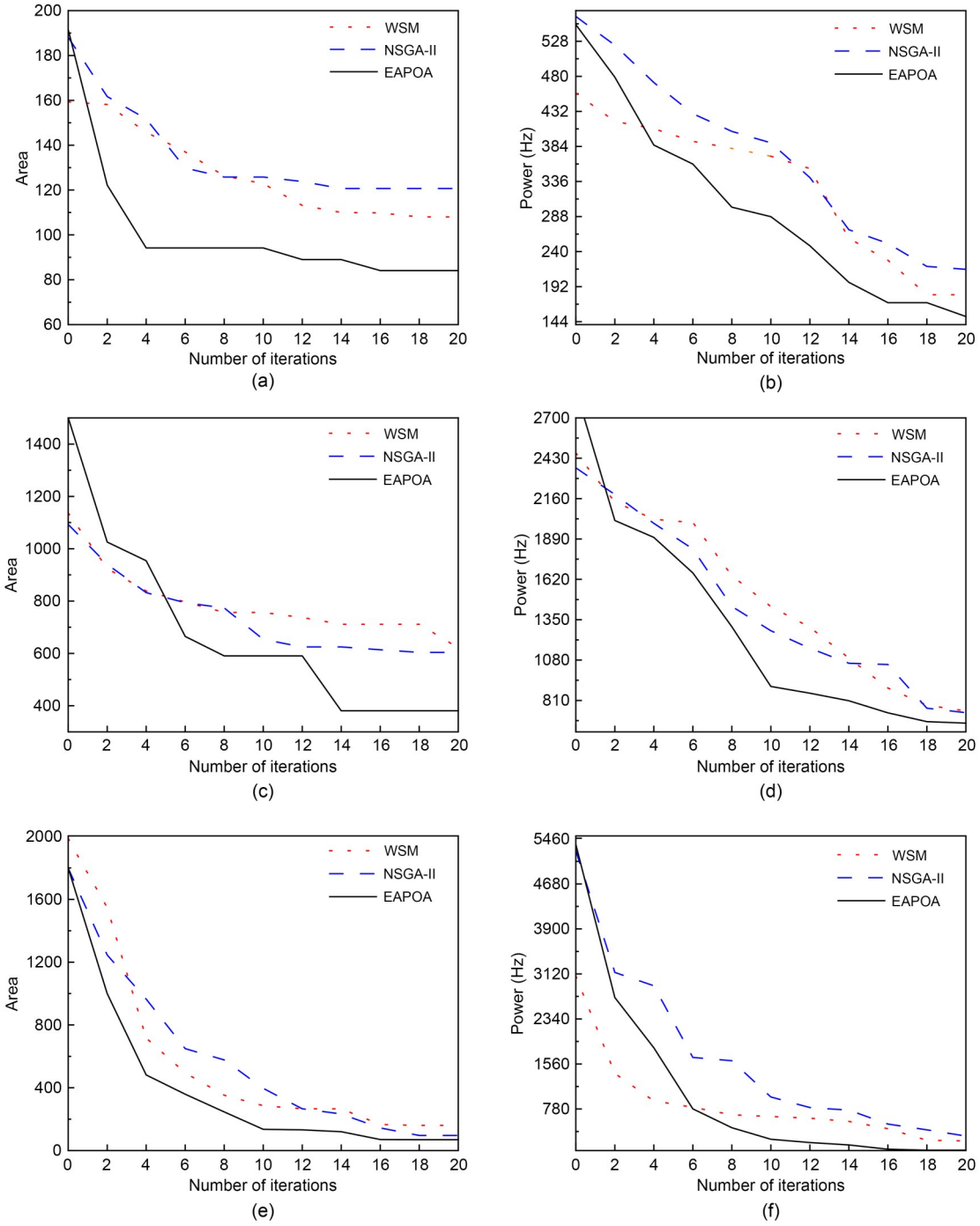


Fig. 7 Convergence curves of different circuits: (a) area of the br2 circuit; (b) power of the br2 circuit; (c) area of the amd circuit; (d) power of the amd circuit; (e) area of the bcc circuit; (f) power of the bcc circuit

these results are that the dual-population strategy enables individuals to converge independently in the target decision subspace and enhances the parallelism of the algorithm; DRLS makes full use of the information from each dimension to enable the algorithm to explore a promising region. In addition, the mutation strategy fully explores the potential excellent solution and uses the information carried by the head wolf.

5.5 Technology mapping

To demonstrate the performance of EAPOA compared with traditional Product-of-Sums (POS) forms in detail, we compare existing optimized POS based on binary decision diagram (BDD) (Zhang et al., 2019). The experimental results are shown in Table 7.

As shown in Table 7, area of 10 test circuits is improved in the benchmarks considered. Compared with BDD, EAPOA achieves an area optimization rate of 5% in the large-scale test circuit (the e64 benchmark). $E^{i}_{\text{sωd}}$ of EAPOA in some test circuits is higher due to the influence of signal probability. Furthermore, areas and delays of the circuits are reported after performing technology mapping by the logic synthesis tool ABC (Berkeley Logic Synthesis and Verification Group, 2018) using the MCNC generic standard cell library. The actual areas and delays after technology mapping are shown in Table 8.

For small-scale circuits, such as rd53, con1, and sao2, EAPOA's area mapping effect is not significant, but EAPOA has low delay. In large-scale circuits with many redundant nodes, such as the e64 benchmark, EAPOA has area and delay optimization rates of 15% and 19%, respectively. In other words, EAPOA achieves better results with greater computational effort and higher power consumption. After technology mapping, by selecting area as the optimal cost implementation, we obtain the average improvement in area of 2.27%.

5.6 LUT mapping

We map the optimized benchmark into a K-input lookup table (LUT). All LUTs are mapped by the "if-K" command. The experimental results are shown in Table 9. The table includes the optimized LUT results obtained using BDD and EAPOA. For each, it lists the size and depth for 4-LUT (S_4, D_4) and 6-LUT (S_6, D_6). In the case of 4-LUT, EAPOA demonstrates improvements or maintains performance in 9 out of 13 benchmarks, except rd53, con1, 5xp1, and br1, because the cost-aware algorithm will add some XNOR operators through structural reuse, but these XNOR operators are not reused. In the case of 6-LUT, EAPOA demonstrates improvements or maintains performance in 11 out of 13 benchmarks. Especially for the largest e64 circuit, the EAPOA-optimized FPGA implementation has improved both the LUT size and depth. After 6-LUT FPGA mapping, by selecting area as the optimal cost implementation, we obtain average improvements in the LUT size of 16.33%. It is worth mentioning that the mapping performed is a generic LUT mapping that does not take into account specific FPGA architecture. Therefore, in the LUT mapping and process library mapping, we do not consider the influence of power consumption. Since the power optimization is expressed based on the switching activity $E^i_{\text{sωd}}$, we focus on the logic level of the experiment result $E^i_{\text{sωd}}$.

6 Conclusions

Area and power optimization of MPRM logic circuits based on XNOR/OR is a typical combinatorial optimization problem. We propose an MQWPA, which includes quantum encoding, dual-population strategy, DRLS, mutation strategy, and termination criterion. The dual-population strategy ensures diversity in the independent convergence decision subspace; DRLS uses the base point and reverse point to ensure the diversity of the population; mutation strategy ensures the correctness of the convergence direction by making full use of the dimensional information carried by individuals and exchanging information with the head wolf in time. Moreover, based on the proposed MQWPA, we propose an EAPOA for MPRM logic circuits, which searches Pareto non-dominated solution sets corresponding to MPRM circuits. The effectiveness of the EAPOA on optimal solutions, time, and convergence is verified by comprehensive experiments, which can be used as a promising approach for developing electronic design automation tools.

We believe that running our proposed algorithm on a quantum machine is feasible for the following reasons: MQWPA uses quantum bits for chromosome encoding and uses single-phase quantum bit rotation gates to adjust the state of quantum bits; the polarity optimization problem is a high-dimensional combinatorial optimization

Table 7 Experimental results of BDD and EAPOA

Circuit	BDD		EAPOA	
	Area	$E^i_{\text{sωd}}$ (Hz)	Area	$E^i_{\text{sωd}}$ (Hz)
rd53	**19**	31.44	**19**	31.44
con1	**20**	30.24	21	36.59
5xp1	68	100.21	**37**	61.39
rd84	172	200.45	**55**	98.00
sao2	**138**	198.62	179	303.80
br1	137	300.12	**133**	313.27
br2	87	97.63	**78**	145.18
amd	621	700.21	**373**	729.32
alu4	487	804.00	**479**	805.52
table5	59	99.62	**50**	98.62
bcc	61	150.62	**57**	153.03
bcd	101	197.31	**70**	199.69
e64	300	880.21	**285**	870.21

The better results are in bold

Table 8 Technology mapping results

Circuit	BDD		EAPOA	
	Area (μm^2)	Delay (ns)	Area (μm^2)	Delay (ns)
rd53	**52**	15.3	57	12.3
con1	98	25.3	**97**	25.6
5xp1	119	24.3	**112**	21.8
rd84	240	58.6	**237**	57.5
sao2	**190**	46.2	214	38.9
br1	347	83.8	**313**	78.8
br2	138	31.5	**137**	27.5
amd	**658**	146.3	706	149.1
alu4	**735**	136.8	831	196.6
table5	284	60.9	**247**	56.0
bcc	275	42.8	**222**	44.6
bcd	301	57.1	**284**	57.9
e64	511	62.1	**432**	50.6

The better results are in bold

Table 9 LUT optimization results for the MCNC benchmarks

Circuit	BDD				EAPOA			
	S_4	D_4	S_6	D_6	S_4	D_4	S_6	D_6
rd53	**3**	3	**3**	3	5	2	**3**	3
con1	**8**	3	**6**	4	11	5	**6**	4
5xp1	**9**	5	**6**	4	11	5	**6**	4
rd84	18	14	6	3	**15**	15	**5**	5
sao2	25	6	20	12	**24**	11	**16**	11
br1	**33**	17	30	13	37	20	**21**	12
br2	19	9	**11**	7	**16**	9	**11**	6
amd	95	49	**48**	25	**86**	42	55	27
alu4	105	46	**49**	44	**98**	47	64	42
table5	29	13	17	4	**26**	12	**11**	5
bcc	37	14	36	9	**23**	11	**10**	5
bcd	58	15	47	10	**29**	13	**20**	8
e64	89	28	58	17	**55**	19	**43**	11

S_K: the size of K-LUT; D_K: the depth of K-LUT. LUT size and depth are generated by 4- and 6-LUT technology mapping using the ABC command "if-K." The best results are in bold

problem, and benchmark circuit reads are also represented by two logic values, 0 and 1.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos. 62102130 and 62502451), the Natural Science Foundation of Zhejiang Province of China (No. LMS26F020021), the China Postdoctoral Science Foundation (No. 2026M791695), the Natural Science Foundation of Hebei Province (Nos. F2024204001 and F2025204012), and the Science Research Project of Hebei Education Department (No. QN2024138).

Author contributions

Chengcheng CHEN designed the research. Yuhao ZHOU, Juncai HE, and Dongge QIN processed the data. Zhanhui SHI, Wei ZHANG, and Jicong DUAN drafted the paper. Zhenxue HE and Limin XIAO helped organize the paper. All the authors engaged in discussions, ensuring the comprehensiveness and cohesiveness of the final paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The source code and data that support the findings of this study are publicly available at https://gitee.com/Yuhao_Zhou.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used ChatGPT to improve language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Abu Lebdeh M, Abunahla H, Mohammad B, et al., 2017. An efficient heterogeneous memristive xnor for in-memory computing. *IEEE Trans Circ Syst I Reg Papers*, 64(9):2427-2437. https://doi.org/10.1109/TCSI.2017.2706299

Belot J, Cherkaoui A, Laurent R, et al., 2021. An area- and power-efficient stochastic number generator for Bayesian sensor fusion circuits. *IEEE Des Test*, 38(6):69-77. https://doi.org/10.1109/MDAT.2021.3050694

Berkeley Logic Synthesis and Verification Group, 2018. ABC: a System for Sequential Synthesis and Verification. https://github.com/berkeley-abc/abc [Accessed on May 10, 2025].

Chen HW, Xue XY, Liu CS, et al., 2021. Logic gates based on neuristors made from two-dimensional materials. *Nat Electron*, 4(6):399-404. https://doi.org/10.1038/s41928-021-00591-z

Chen J, Li XB, Zhao YX, et al., 2025. TPCMEA: a tri-population evolutionary algorithm with adaptive stage-switching for complex CMOPs. *Appl Soft Comput*, 184:113792. https://doi.org/10.1016/j.asoc.2025.113792

Chen YB, Yang D, Yu JQ, 2018. Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm. *IEEE Trans Aerosp Electron Syst*, 54(6):2853-2872. https://doi.org/10.1109/TAES.2018.2831138

Das A, Pradhan SN, 2018. Area-power-temperature aware AND-XOR network synthesis based on shared mixed polarity Reed-Muller expansion. *Int J Intell Syst Appl*, 10(12):35-46. https://doi.org/10.5815/ijisa.2018.12.04

Drechsler R, Theobald M, Becker B, 1996. Fast OFDD-based minimization of fixed polarity Reed-Muller expressions. *IEEE Trans Comput*, 45(11):1294-1299. https://doi.org/10.1109/12.544485

Drechsler R, Becker B, Drechsler N, 2000. Genetic algorithm for minimisation of fixed polarity Reed-Muller expressions. *IEE Proc Comput Digit Tech*, 147(5):349-353.

He ZX, Xiao LM, Ruan L, et al., 2017. A power and area optimization approach of mixed polarity Reed-Muller expression for incompletely specified Boolean functions. *J Comput Sci Technol*, 32(2):297-311. https://doi.org/10.1007/s11390-017-1723-1

He ZX, Wu XQ, Wang C, et al., 2021. Delay optimization for ternary fixed polarity Reed–Muller circuits based on multilevel adaptive quantum genetic algorithm. *Int J Intell Syst*, 36(10):5981-6006. https://doi.org/10.1002/int.22538

Kalay U, Hall DV, Perkowski MA, 2000. A minimal universal test set for self-test of EXOR-Sum-of-Products circuits. *IEEE Trans Comput*, 49(3):267-276. https://doi.org/10.1109/12.841130

Long W, Wu TB, Tang MZ, et al., 2020. Grey wolf optimizer algorithm based on lens imaging learning strategy. *Acta Autom Sin*, 46(10):2148-2164 (in Chinese). https://doi.org/10.16383/j.aas.c180695

Niu B, Wang H, Wang JW, et al., 2013. Multi-objective bacterial foraging optimization. *Neurocomputing*, 116:336-345. https://doi.org/10.1016/j.neucom.2012.01.044

Park SY, Na K, Vöröslakos M, et al., 2022. A miniaturized 256-channel neural recording interface with area-efficient hybrid integration of flexible probes and CMOS integrated circuits. *IEEE Trans Biomed Eng*, 69(1):334-346. https://doi.org/10.1109/TBME.2021.3093542

Qin DG, He ZX, Zhao XJ, et al., 2023. Area and power optimization for fixed polarity Reed-Muller logic circuits based on multi-strategy multi-objective artificial bee colony algorithm. *Eng Appl Artif Intell*, 121:105906. https://doi.org/10.1016/j.engappai.2023.105906

Rahnamayan S, Tizhoosh HR, Salama MMA, 2008. Opposition-based differential evolution. *IEEE Trans Evol Comput*, 12(1):64-79. https://doi.org/10.1109/TEVC.2007.894200

Tsvelikhovskiy B, Safro I, Alexeev Y, 2026. Equivariant quantum approximate optimization algorithm. *IEEE Trans Quantum Eng*, 7:3101013. https://doi.org/10.1109/TQE.2026.3654930

Wagle R, Sharma P, Sharma C, et al., 2021. Bio-inspired hybrid BFOA-PSO algorithm-based reactive power controller in a standalone wind-diesel power system. *Int Trans Electr Energy Syst*, 31(3):e12778.

Wang L, Xie GJ, 2020. A novel XOR/XNOR structure for modular design of QCA circuits. *IEEE Trans Circ Syst II Exp Briefs*, 67(12):3327-3331. https://doi.org/10.1109/TCSII.2020.2989496

Wang X, Lu Y, Zhang Y, et al., 2013. Probabilistic modeling during power estimation for mixed polarity Reed-Muller logic circuits. Proc IEEE Int Conf on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, p.1414-1418. https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.247

Wang X, Zhang R, Wang WK, et al., 2015. Polarity searching for MPRM logic circuit based on improved adaptive genetic algorithm. Proc 12th Int Conf on Ubiquitous Intelligence and Computing and IEEE 12th Int Conf on Autonomic and Trusted Computing and IEEE 15th Int Conf on Scalable Computing and Communications and Its Associated Workshops, p.1354-1358. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.244

Wang YC, Cui ZH, Li WC, 2019. A novel coupling algorithm based on glowworm swarm optimization and bacterial foraging algorithm for solving multi-objective optimization problems. *Algorithms*, 12(3):61. https://doi.org/10.3390/a12030061

Wu XQ, Wang P, Li SQ, et al., 2025. An area optimization approach for large-scale RM-TB dual logic circuits based on a multitasking optimization algorithm. *IEEE Trans Comput*, 74(7):2348-2363. https://doi.org/10.1109/TC.2025.3558077

Xia YS, Wu XW, Almaini AEA, 2003. Power minimization of FPRM functions based on polarity conversion. *J Comput Sci Technol*, 18(3):325-331. https://doi.org/10.1007/BF02948902

Xing HL, Zhou XY, Wang XH, et al., 2019. An integer encoding grey wolf optimizer for virtual network function placement. *Appl Soft Comput*, 76:575-594. https://doi.org/10.1016/j.asoc.2018.12.037

Xu J, Kang XY, Fan ZK, et al., 2021. Design of highly uniform magnetic field coils with wolf pack algorithm. *IEEE Sens J*, 21(4):4412-4424. https://doi.org/10.1109/JSEN.2020.3030622

Yan PP, Yu HZ, Shi XH, et al., 2020. Area and power optimization of MPRM circuit based on PDTDPSO. *Comput Appl Softw*, 37(5):72-76 (in Chinese). https://doi.org/10.3969/j.issn.1000-386x.2020.05.013

Yang CG, Tu XY, Chen J, 2007. Algorithm of marriage in honey bees optimization based on the wolf pack search. Proc Int Conf on Intelligent Pervasive Computing, p.462-467. https://doi.org/10.1109/IPC.2007.104

Yang M, Xu HY, Almaini AEA, 2011. Optimization of mixed polarity Reed-Muller functions using genetic algorithm. Proc 3rd Int Conf on Computer Research and Development, p.293-296. https://doi.org/10.1109/ICCRD.2011.5764198

Yang S, 1991. Logic synthesis and optimization benchmarks user guide version 3.0. Proc MCNC Int Workshop on Logic Synthesis, p.1-44.

Yi J, Bai JR, Zhou W, et al., 2018. Operating parameters optimization for the aluminum electrolysis process using an improved quantum-behaved particle swarm algorithm. *IEEE Trans Ind Inform*, 14(8):3405-3415. https://doi.org/10.1109/TII.2017.2780884

Yin P, Shu Z, Xia YJ, et al., 2021. A low-area and low-power comma detection and word alignment circuits for JESD204B/C controller. *IEEE Trans Circ Syst I Reg Papers*, 68(7):2925-2935. https://doi.org/10.1109/TCSI.2021.3072772

Yu F, Li YX, Wei B, et al., 2014. The application of a novel OBL based on lens imaging principle in PSO. *Acta Electron Sin*, 42(2):230-235 (in Chinese). https://doi.org/10.3969/j.issn.0372-2112.2014.02.004

Yu HZ, Wang PJ, Wang DS, et al., 2013. Discrete ternary particle swarm optimization for area optimization of MPRM circuits. *J Semicond*, 34(2):025011.

https://doi.org/10.1088/1674-4926/34/2/025011

Yu HZ, Wang PJ, Zhang HH, et al., 2017. Optimization of MPRM circuits based on ternary diversity particle swarm optimization. *Acta Electron Sin*, 45(7):1601-1607 (in Chinese). https://doi.org/10.3969/j.issn.0372-2112.2017.07.008

Zhang HH, Chen ZW, Wang PJ, 2019. Area and delay optimization of binary decision diagrams mapped circuit. *J Electron Inform Technol*, 41(3):725-731 (in Chinese).

Zhou YH, He ZX, Wang T, et al., 2022a. Area and power optimization approach for mixed polarity Reed–Muller logic circuits based on multi-strategy bacterial foraging algorithm. *Appl Soft Comput*, 130:109720. https://doi.org/10.1016/j.asoc.2022.109720

Zhou YH, He ZX, Chen C, et al., 2022b. An efficient power optimization approach for fixed polarity Reed-Muller logic circuits based on metaheuristic optimization algorithm. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 41(12):5380-5393. https://doi.org/10.1109/TCAD.2022.3149720

Zhou YH, He ZX, Jiang JH, et al., 2024. An efficient and fast area optimization approach for mixed polarity Reed-Muller logic circuits. *Chin J Electron*, 33(5):1165-1180. https://doi.org/10.23919/cje.2022.00.407

Zhou YH, Jiang JH, He ZX, et al., 2025. An efficient area and reliability optimization method for MPRM circuits based on high-dimensional genetic algorithm. *ACM Trans Des Autom Electron Syst*, 30(2):31. https://doi.org/10.1145/3712591

Zhu YW, Qin YH, Yang D, et al., 2023. An enhanced decomposition-based multi-objective evolutionary algorithm with a self-organizing collaborative scheme. *Expert Syst Appl*, 213:118915. https://doi.org/10.1016/j.eswa.2022.118915

Zitzler E, Deb K, Thiele L, 2000. Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput*, 8(2):173-195. https://doi.org/10.1162/106365600568202