

文章编号:1672-3961(2024)06-0147-09 DOI:10.6040/j.issn.1672-3961.0.2023.210

# 基于 AMSChOA 的 MPRM 电路面积优化

张梦雨<sup>1,2</sup>, 何振学<sup>1,2\*</sup>, 赵晓君<sup>1,2</sup>, 王浩然<sup>3</sup>, 肖利民<sup>4</sup>, 王翔<sup>5</sup>

(1.河北农业大学智能农业装备研究院, 河北 保定 071001; 2.河北农业大学河北省农业大数据重点实验室, 河北 保定 071001; 3.河北农业大学林学院, 河北 保定 071001; 4.北京航空航天大学计算机学院, 北京 100191; 5.北京航空航天大学电子信息工程学院, 北京 100191)

**摘要:**为解决现有基于同或/或(XNOR/OR)的混合极性 Reed-Muller(mixed polarity Reed-Muller, MPRM)电路面积优化方法中存在的收敛速度较慢、不容易跳出局部最优等问题,提出一种基于自适应多策略选择黑猩猩优化算法(adaptive multi-strategy selection chimp optimization algorithm, AMSChOA)的 MPRM 电路面积优化方法。AMSChOA 使用柯西变异、螺旋搜索、随机搜索和翻筋斗策略在4个最优黑猩猩附近进行搜索,扩大算法的搜索范围。针对其他黑猩猩个体加入动态学习因子策略,动态学习4个最优黑猩猩位置,加快算法跳出局部最优。利用提出的 AMSChOA 对基于 XNOR/OR 的 MPRM 电路进行面积优化,搜索电路面积最小时对应的 MPRM 电路。基于北卡罗来纳微电子中心(Microelectronics Center of North Carolina, MCNC)基准测试电路的试验结果表明,本研究提出的方法有效,与基于传统黑猩猩优化算法、粒子群算法、改进粒子群算法的 MPRM 电路面积优化方法相比,最高面积优化率为 68.09%,平均优化率为 41.24%。

**关键词:**MPRM 面积优化;自适应多策略选择黑猩猩优化算法;混合极性 Reed-Muller;动态学习因子;组合优化问题

**中图分类号:**V443;TP391.72 **文献标志码:**A

**引用格式:**张梦雨,何振学,赵晓君,等.基于 AMSChOA 的 MPRM 电路面积优化[J].山东大学学报(工学版),2024,54(6):147-155.

ZHANG Mengyu, HE Zhenxue, ZHAO Xiaojun, et al. Area optimization for MPRM circuits based on AMSChOA [J]. Journal of Shandong University (Engineering Science), 2024, 54(6):147-155.

## Area optimization for MPRM circuits based on AMSChOA

ZHANG Mengyu<sup>1,2</sup>, HE Zhenxue<sup>1,2\*</sup>, ZHAO Xiaojun<sup>1,2</sup>, WANG Haoran<sup>3</sup>, XIAO Limin<sup>4</sup>, WANG Xiang<sup>5</sup>

(1. Intelligent Agricultural Equipment Research Institute, Hebei Agricultural University, Baoding 071001, Hebei, China; 2. Key Laboratory of Agricultural Big Data of Hebei Province, Hebei Agricultural University, Baoding 071001, Hebei, China; 3. College of Forestry, Hebei Agricultural University, Baoding 071001, Hebei, China; 4. School of Computer Science and Engineering, Beihang University, Beijing 100191, China; 5. School of Electronic Information Engineering, Beihang University, Beijing 100191, China)

**Abstract:** To solve the problems of slow convergence speed and not easy to jump out of local optimum in the existing XNOR/OR based mixed polarity Reed-Muller (MPRM) circuit area optimization approaches, an approach based on adaptive multi-strategy selection chimp optimization algorithm (AMSChOA) was proposed. AMSChOA used the Cauchy variation, spiral search, random search, and somersault strategy to search in the vicinity of the four optimal chimps to expand the search range of the algorithm, and a dynamic learning factor strategy was added for other chimpanzee individuals to dynamically learn the four optimal chimpanzee locations to accelerate the algorithm to jump out of the local optimum. The proposed AMSChOA was used to optimize the area of the XNOR/OR-based MPRM circuits, and the MPRM circuits corresponding to the smallest circuit area were searched. The

**收稿日期:**2023-08-30

**基金项目:**国家自然科学基金资助项目(62102130);中央引导地方科技发展资金资助项目(226Z0201G);河北省自然科学基金资助项目(F2020204003);河北省青年拔尖人才计划资助项目(BJ2019008);河北省高等学校科学技术研究资助项目(QN2022138);河北省省属高等学校基本科研业务费研究资助项目(KY2022073)

**第一作者简介:**张梦雨(2000—),女,河南郑州人,硕士研究生,主要研究方向为 RM 电路智能优化。E-mail: 20222060110@pgs.hebau.edu.cn

**\*通信作者简介:**何振学(1987—),男,山东泰安人,副教授,博士生导师,博士,主要研究方向为智能优化算法、最优化理论、电子设计自动化。

E-mail: hezhenxue@buaa.edu.cn

experimental results based on Microelectronics Center of North Carolina(MCNC) benchmark test circuits showed that the proposed approach in this study was effective, with a maximum area optimization rate of 68.09% and an average optimization rate of 41.24%, compared to the area optimization methods for MPRM circuits based on the traditional chimp optimization algorithm, particle swarm optimization, and modified particle swarm optimization.

**Keywords:** area optimization; adaptive multi-strategy selection chimp optimization algorithm; mixed polarity Reed-Muller; dynamic learning factor; combinatorial optimization problem

## 0 引言

数字逻辑电路通常采用2种实现方式:基于与/或/非(AND/OR/NOT)的 Boolean 逻辑和基于同或/或(XNOR/OR)的 Reed-Muller(RM)逻辑<sup>[1]</sup>。研究表明,与 Boolean 逻辑相比, RM 逻辑在电路面积、功耗、可靠性等性能优化方面更具优势<sup>[2]</sup>。因此,基于 XNOR/OR 的 RM 逻辑电路优化已成为集成电路领域的研究热点。

RM 逻辑电路综合性能优化的一个重要技术手段是极性优化。与固定极性 RM 逻辑电路相比,混合极性 RM(mixed polarity Reed-Muller, MPRM)逻辑电路在优化面积、功耗等方面的空间大且更具优势。因此,MPRM 逻辑电路受到更多关注。从本质上看,MPRM 电路面积优化就是搜索最小电路面积时对应的最佳电路极性。任意一个有  $n$  个变量的逻辑函数都具有  $3^n$  个极性,对应  $3^n$  个不同的 MPRM 表达式<sup>[3]</sup>。因此,MPRM 电路面积优化属于组合优化问题。MPRM 表达式中所含或项数量决定 MPRM 电路面积。近年来,使用群体智能算法解决此类优化问题成为越来越多研究学者的选择。文献[4]针对三值变量组合优化问题,提出一种改进的多扰动烟花算法,利用该算法搜索具有最小面积的 MPRM 逻辑电路对应极性,兼顾全局和局部搜索,具有良好的效果,但所需参数较多;文献[5]提出一种全离散粒子群优化算法,用于解决较多输入位数的可编程阵列结构纳电子混合极性 RM 电路面积优化问题,具有较强的全局收敛能力和结果稳定性,但缺乏对收敛性的验证;文献[6]提出一种具有爆炸和重启机制的鲸鱼优化算法,结合改进的混合极性转换算法,搜索含与项数量最少的 MPRM 电路,提高算法的收敛速度,增加跳出局部最优的能力,但重启策略在一定程度上影响了算法的效率;文献[7]使用粒子群优化算法,通过离散的方式,将该算法应用在基于 XNOR/OR 的电路面积优化的三值优化问题上,但只处理了

中小规模的电路,缺少对中大规模电路的测试。使用群体智能优化算法求解 MPRM 面积优化还存在算法搜索最优解的速度慢、易陷入局部最优等问题。因此,研究 MPRM 电路面积优化尤为重要和必要。

文献[8]提出一种新型元启发式优化算法——黑猩猩优化算法(chimp optimization algorithm, ChOA),主要模拟黑猩猩群体狩猎行为。与其他算法相比,ChOA 具有收敛速度快、寻优精度高等特点。目前,黑猩猩优化算法已广泛用于各领域的研究;文献[9]利用黑猩猩优化算法提出一种新的彩色图像增强方法,提高参数选择效率;文献[10]将黑猩猩优化算法应用于生物医学数据分类,有效提高特征数据选择精度;文献[11]利用黑猩猩优化算法进行单词识别,能有效提高识别效率和准确率;文献[12]将黑猩猩优化算法用于太阳能网络的模型参数估计,准确建模和识别太阳能电池参数,证实了算法的鲁棒性和高性能。与其他群智能算法相比,ChOA 具有参数较少、结构简单、稳定性高等特点,且 MPRM 电路优化属于组合优化问题,使用 ChOA 对 MPRM 电路面积优化更加有效。本研究使用 ChOA 对 MPRM 电路进行最佳极性搜索,以提高面积优化效率。

## 1 MPRM 电路面积优化理论

目前,集成电路设计大多以 Boolean 逻辑为主,在基准电路中一般以 Boolean 逻辑表达式出现。随着 RM 逻辑电路的出现,在部分电路上 RM 逻辑电路的可测试性、面积等方面的性能均好于 Boolean 逻辑电路。为得到 MPRM 逻辑表达式,需要进行极性转换。现有的基于 XNOR/OR 形式的 MPRM 表达式需要通过基于 AND/OR/NOT 形式的 Boolean 表达式进行转化,这个过程称作极性转换<sup>[13]</sup>。

### 1.1 MPRM 电路表达式

在混合极性 XNOR/OR 电路中, $n$  个输入变量

的逻辑函数  $f(x_1, x_2, \dots, x_n)$  对应有  $3^n$  个不同的极性,对于任一极性  $P$ ,对应的 MPRM 电路表达式为:

$$f_P(x_1, x_2, \dots, x_n) = \odot \prod_{k=0}^{2^n-1} (d_k + S_k), \quad (1)$$

式中:  $\odot \prod$  为 XNOR 运算;  $S_k$  为 OR 项,  $S_k = \hat{x}_1 + \hat{x}_2 + \dots + \hat{x}_n$ , 其中  $\hat{x}_n$  为逻辑函数中的变量,下标  $k$  对应的三进制形式为  $k_1, k_2, \dots, k_n$ ;  $d_k$  为判断  $S_k$  是否出现的值,取值为  $\{0, 1\}$ , 当  $d_k = 1$  时,  $S_k$  不在电路表达式中出现,当  $d_k = 0$  时,  $S_k$  在电路表达式中出现。

由式(1)可以看出,变量的表现形式有 3 种:原变量形式、反变量形式、同时以原变量和反变量 2 种形式出现。因此,一个有  $n$  个变量的函数对应的极性有  $3^n$  个。

### 1.2 变量的极性

输入变量的一组取值称为极性。在混合极性 XNOR/OR 表达式中,每个变量  $\hat{x}_j$  有 3 种表现形式,变量的出现形式与极性  $P$  有关,极性  $P$  的三进制表示为  $P_1, P_2, \dots, P_n$ 。当  $P_i = 0$  时,变量  $\hat{x}_j$  以原变量  $x_j$  的形式出现;当  $P_i = 1$  时,变量  $\hat{x}_j$  以反变量  $\bar{x}_j$  的形式出现;当  $P_i = 2$  时,变量  $\hat{x}_j$  以原变量和反变量的形式同时出现。因此,变量  $\hat{x}_j$  与  $P_n$  和  $k_n$  的关系为:

$$\hat{x}_j = \begin{cases} x_j, & P_n = 0, k_n = 0 \\ 0, & P_n = 0, k_n = 1 \\ \bar{x}_j, & P_n = 1, k_n = 0 \\ 1, & P_n = 1, k_n = 1 \\ x_j, & P_n = 2, k_n = 0 \\ \bar{x}_j, & P_n = 2, k_n = 1 \end{cases} \quad (2)$$

### 1.3 极性转换

在基准电路中,为得到 MPRM 逻辑表达式,需要对 Boolean 逻辑表达式进行极性转换。将给定的最小项表达式转换到极性为  $P$  的 MPRM 表达式的极性转换方法具体步骤如下。

(1) 读取一个含有  $n$  个变量的最小项表达式,将其中所含的最小项以二进制形式表示。

(2) 将所需极性转换为对应的三进制形式  $P_1, P_2, \dots, P_n$ 。若  $P_i = 1$ ,将最小项对应位数与  $P_i$  进行异或运算,得到新的最小项;否则,对最小项不进行操作。

(3) 若  $P_i = 0$  或  $P_i = 1$ ,且二进制字符串的第  $i$  位为 1,则生成一个新项,将二进制字符串的第  $i$  位替换为 0,并复制其他所有位以生成新项;若  $P_i = 2$ ,

则保持其他所有位不变。

(4) 删除原始二进制字符串和新生成的二进制字符串之间的重复项。

(5) 对其他剩余变量重复执行步骤(3)(4),剩余项构成混合极性 XNOR/OR 表达式所需极性的和项。

### 1.4 MPRM 电路面积模型

由式(1)可知, XNOR 操作和 OR 操作组成了 MPRM 电路展开式。因此,基于 XNOR/OR 的 MPRM 电路面积表示为这两部分操作的项数之和。多输入逻辑运算需要分解为二输入运算,根据分解后的运算对 MPRM 电路面积进行计算。因此,可用二输入 XNOR 门的个数和二输入 OR 门的个数之和表示 MPRM 电路面积<sup>[14]</sup>。

由式(1)可知,多输入 OR 门  $S_k$  可由  $m_i$  个二输入 OR 操作组成,  $m_i$  的具体计算式为:

$$m_i = \sum_{k=0}^{n-1} \bar{i}_k - 1, \quad (3)$$

式中,  $n$  为变量个数,  $\bar{i}_k$  为对  $i_k$  进行取反。

式(1)可由二输入 XNOR 操作和二输入 OR 操作组成,其项数计算方式分别为:

$$M = \sum_{j=0}^{2^n-1} \bar{d}_j - 1, \quad (4)$$

$$N = \sum_{j=0}^{2^n-1} \left[ \bar{d}_j \left( \sum_{k=1}^n \bar{c}_k - 1 \right) \right], \quad (5)$$

式中,  $\bar{d}_j$  为乘积项数量,  $\bar{c}_k$  为输入变量数。因此,基于 XNOR/OR 的 MPRM 电路面积之和

$$S = M + N = \sum_{j=0}^{2^n-1} \bar{d}_j \sum_{k=0}^{n-1} \bar{i}_k - 1, \quad (6)$$

## 2 黑猩猩优化算法及其改进

### 2.1 黑猩猩优化算法

在黑猩猩群体中,个体的能力、智力并不相似。鉴于各自能力不同,将黑猩猩分为 4 种角色:攻击者、障碍者、追逐者和驱动者。每一类黑猩猩都有自己的特殊能力,依据不同策略更新自己的位置及预测猎物的位置。在获得猎物的同时,黑猩猩的狩猎行为还受到获取猎物之后所带来的性行为和社会效益的影响,以激励它们在狩猎的最后阶段出现混乱捕获猎物,实现黑猩猩对猎物的攻击、拦截、追逐和驱赶行为。

黑猩猩的捕猎过程主要分为 2 个阶段:全局探索阶段,即驱赶、阻拦、追逐猎物;局部勘探阶段,即

攻击猎物。假设黑猩猩种群大小为  $K$ , 第  $i$  只黑猩猩的位置为  $\mathbf{X}_i$ , 群体最优解为  $\mathbf{X}_{\text{Attacker}}$ , 次优解、第三最优解、第四最优解依次为  $\mathbf{X}_{\text{Barrier}}$ 、 $\mathbf{X}_{\text{Chaser}}$ 、 $\mathbf{X}_{\text{Driver}}$ 。其他黑猩猩被迫根据最优黑猩猩位置更新自己的位置。

### 2.1.1 驱赶、阻拦、追逐猎物

猎物是在探索和开发阶段被猎杀, 黑猩猩个体根据猎物的位置改变自身位置, 位置更新公式为:

$$\mathbf{X}_{\text{chimp}}(t+1) = \mathbf{X}_{\text{prey}}(t) - \mathbf{a}(c\mathbf{X}_{\text{prey}}(t) - m\mathbf{X}_{\text{chimp}}(t)), \quad (7)$$

式中:  $t$  为当前迭代次数;  $\mathbf{X}_{\text{chimp}}$  为黑猩猩位置;  $\mathbf{X}_{\text{prey}}$  为猎物位置;  $\mathbf{a}$  为系数矢量, 决定黑猩猩个体与猎物之间的距离,  $\mathbf{a} = 2\mathbf{f}r_1 - \mathbf{f}$ , 其中  $r_1$  为  $[0, 1]$  之间的随机数,  $\mathbf{f}$  是由收敛因子组成的系数矩阵, 通过迭代过程使收敛因子从 2.5 非线性减小到 0;  $c$  为控制黑猩猩驱逐和追赶猎物的系数, 为猎物提供随机权重,  $c = 2r_2$ , 其中  $r_2$  为  $[0, 1]$  之间的随机数;  $m$  为根据各种混沌映射计算出的混沌向量, 表示黑猩猩在狩猎过程中性动机的影响,  $m = \mathbf{C}$ , 其中  $\mathbf{C}$  为通过混沌模型计算出的值组成的混沌向量。假设迭代过程中, 在正常位置更新机制和混沌模型之间选择更新黑猩猩位置的概率为 50%。黑猩猩位置更新为:

$$\mathbf{X}_{\text{chimp}}(t+1) = \begin{cases} \mathbf{X}_{\text{prey}}(t) - \mathbf{a}\mathbf{d}, & \mu \leq 0.5 \\ \mathbf{C}, & \mu > 0.5 \end{cases}, \quad (8)$$

式中:  $\mathbf{d}$  为黑猩猩与猎物之间的间距;  $\mu$  为选择判断因子, 根据  $\mu$  选择不同的位置更新机制进行黑猩猩个体位置更新。

### 2.1.2 攻击猎物

为了在数学上模拟黑猩猩的攻击行为, 通过驱赶、阻拦和追逐 3 种行为, 黑猩猩能够探索到猎物当前位置并对猎物进行包围。黑猩猩攻击猎物的数学模型为:

$$\mathbf{d}_{\text{Attacker}} = |c_1\mathbf{x}_{\text{Attacker}} - m_1\mathbf{x}|, \quad (9)$$

$$\mathbf{d}_{\text{Barrier}} = |c_2\mathbf{x}_{\text{Barrier}} - m_2\mathbf{x}|, \quad (10)$$

$$\mathbf{d}_{\text{Chaser}} = |c_3\mathbf{x}_{\text{Chaser}} - m_3\mathbf{x}|, \quad (11)$$

$$\mathbf{d}_{\text{Driver}} = |c_4\mathbf{x}_{\text{Driver}} - m_4\mathbf{x}|, \quad (12)$$

式中,  $\mathbf{d}_{\text{Attacker}}$ 、 $\mathbf{d}_{\text{Barrier}}$ 、 $\mathbf{d}_{\text{Chaser}}$ 、 $\mathbf{d}_{\text{Driver}}$  为 4 类最优黑猩猩个体与猎物的间距,  $\mathbf{x}_{\text{Attacker}}$ 、 $\mathbf{x}_{\text{Barrier}}$ 、 $\mathbf{x}_{\text{Chaser}}$ 、 $\mathbf{x}_{\text{Driver}}$  为 4 类最优黑猩猩个体的位置,  $\mathbf{x}$  为当前黑猩猩个体的位置信息,  $c_1$ 、 $c_2$ 、 $c_3$ 、 $c_4$  均为  $[0, 1]$  之间的随机数,  $m_1$ 、 $m_2$ 、 $m_3$ 、 $m_4$  为混沌向量。通过每类黑猩猩与猎物的间距可计算 4 类最优黑猩猩的位置, 剩余黑猩猩个体位置依据 4 类最优个体位置即可得出。4 类最优黑猩猩个体的位置计算公式分别为:

$$\mathbf{x}_1 = \mathbf{X}_{\text{Attacker}} - \mathbf{a}_1\mathbf{d}_{\text{Attacker}}, \quad (13)$$

$$\mathbf{x}_2 = \mathbf{X}_{\text{Barrier}} - \mathbf{a}_2\mathbf{d}_{\text{Barrier}}, \quad (14)$$

$$\mathbf{x}_3 = \mathbf{X}_{\text{Chaser}} - \mathbf{a}_3\mathbf{d}_{\text{Chaser}}, \quad (15)$$

$$\mathbf{x}_4 = \mathbf{X}_{\text{Driver}} - \mathbf{a}_4\mathbf{d}_{\text{Driver}}, \quad (16)$$

式中,  $\mathbf{a}_1$ 、 $\mathbf{a}_2$ 、 $\mathbf{a}_3$ 、 $\mathbf{a}_4$  为不同的系数矢量。更新后黑猩猩个体的位置

$$\mathbf{X}_{(t+1)} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4}{4}. \quad (17)$$

## 2.2 自适应多策略选择黑猩猩优化算法

与其他智能优化算法相比, ChOA 具有参数少、易实现、稳定性高等优点, 但在迭代过程中也容易出现陷入局部最优和收敛速度慢等问题。首先, 算法将分类的攻击者黑猩猩、障碍者黑猩猩、追逐者黑猩猩、驱动器黑猩猩依次规定为群体最优解、次优解、第三最优解和第四最优解, 但在最优解周围可能还存在更好的解, 导致算法搜索精度低; 其次, 在迭代过程中, 其他黑猩猩被迫根据 4 个最佳黑猩猩的位置更新它们的位置, 如果 4 个最佳黑猩猩的位置没有发生改变, 其他黑猩猩的位置也会随之停止改变, 容易造成算法停止搜索。通过分析算法现存的问题, 本研究提出一种自适应多策略选择黑猩猩优化算法 (adaptive multi-strategy selection chimp optimization algorithm, AMSCOA), 利用最优解自适应选择搜索策略、动态学习因子更新当前黑猩猩位置, 以提高算法的搜索精度, 避免陷入局部最优。

### 2.2.1 自适应多策略选择

原始算法中将分类的 4 个黑猩猩依次规定为种群的最优解、次优解、第三最优解和第四最优解, 但在最优解周围可能还存在更好的解, 为了找到最优解周围潜在的更好的解, 受文献[15-18]启发, 本研究提出 4 种最优解探索策略: 柯西变异、随机搜索、螺旋搜索、翻筋斗策略。通过迭代过程找到 4 个最优解后, 每个最优解可自适应选择其中一种策略进行深度搜索或保持自身不变, 使用该策略只改变个体的部分维度, 重复执行  $w$  次改变个体维度的操作, 以扩大算法的搜索范围, 提高算法的搜索精度。维度和操作次数  $w$  通过正交试验确定。4 种最优解探索策略的具体内容如下。

#### (1) 柯西变异

柯西变异来源于柯西分布, 柯西分布是一个数学期望不存在的连续性概率分布, 在所有分布中, 柯西分布具有最大的散布特性, 即具有最大的拖尾概率, 一维标准柯西分布的概率密度函数

$$f(x) = \frac{m}{\pi(m^2 + x^2)}, \quad -\infty < x < +\infty, \quad (18)$$

式中,  $m$  为比例参数,  $m > 0$ 。当  $m = 1$  时, 为标准柯西分布, 如图 1 所示。

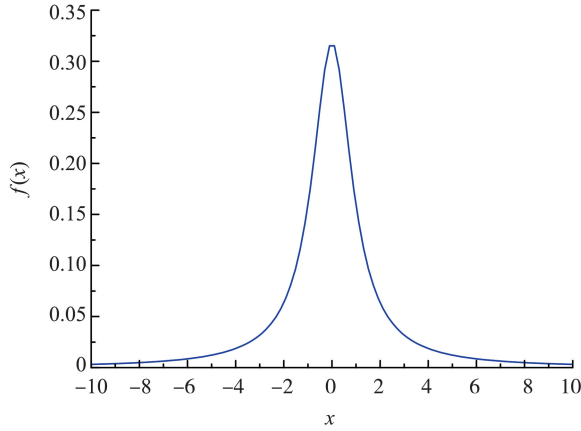


图1 标准柯西分布曲线

Fig.1 Standard Cauchy distribution curve

由图 1 可以看出, 标准柯西分布概率密度曲线是对称图形, 由峰值到两端缓慢降低, 无限接近于  $x$  轴, 但不与  $x$  轴相交, 产生的随机数具有更广的分布范围。因此, 将柯西变异引入最优解搜索阶段, 扩大最优个体搜索范围。该方法数学模型为:

$$y = \text{Cauchy}(0, 1), \quad (19)$$

$$\mathbf{X}_i^d(t+1) = \mathbf{X}_{\text{best}}^d(t) + \mathbf{X}_i^d(t)y, \quad (20)$$

式中,  $\text{Cauchy}()$  为标准柯西函数,  $\mathbf{X}_i^d(t)$  为第  $i$  个个体第  $d$  维的位置,  $\mathbf{X}_{\text{best}}^d(t)$  为当前最优个体攻击者黑猩猩的位置。

### (2) 随机搜索

通过迭代过程找到最优解后, 为了进一步对最优解周围进行探索, 在最优解周围利用随机搜索策略生成新的候选解, 以扩大搜索空间, 避免算法陷入局部最优解。该方法数学模型为:

$$\mathbf{X}_i^d(t+1) = (2r_3 - 1)\mathbf{X}_i^d(t), \quad (21)$$

式中  $r_3$  为  $[0, 1]$  之间的随机数。

### (3) 螺旋搜索

受太阳系的运行规律启发, 以当前迭代过程中产生的最优解  $\mathbf{X}_{\text{Attacker}}$  作为恒星的位置, 其他最优解选取该方法时, 将该点距最优点的距离作为运动半径, 围绕选定的恒星做螺旋运动, 在下次迭代过程中, 将新生成的最优解作为新的恒星位置, 最终生成稳定的“星系”, 从而找到最优解。该方法数学模型为:

$$\mathbf{L} = |\mathbf{X}_{\text{best}}^d(t) - \mathbf{X}_i^d(t)|, \quad (22)$$

$$\mathbf{X}_i^d(t+1) = \mathbf{L} \cos(2\pi r_4) + \mathbf{X}_i^d(t), \quad (23)$$

式中,  $r_4$  为  $[-1, 1]$  之间的随机数,  $\mathbf{L}$  为选取最优解与全局最优解  $\mathbf{X}_{\text{Attacker}}$  的距离。

### (4) 翻筋斗策略

翻筋斗策略来源于猴群算法中的跳跃行为, 黑猩猩与猴子都属于灵长类动物, 它们的觅食行为在某些方面有相似之处, 因此将猴群算法中猴子跳跃这一行为引入黑猩猩最优解搜索过程。当黑猩猩的 4 个最优解依次找到后, 为了找到更优解, 以当前最优个体所在位置为支点, 计算当前个体和最优个体的距离, 并将此距离作为翻筋斗的距离, 搜索周围区域。数学模型为:

$$\mathbf{X}_i^d(t+1) = \mathbf{X}_{\text{best}}^d(t) + \mathbf{L}(r_5\mathbf{X}_i^d(t) - r_6\mathbf{X}_{\text{best}}^d(t)), \quad (24)$$

式中,  $r_5, r_6$  为  $[0, 1]$  之间的随机数。

在初始算法中, 分成的 4 类黑猩猩依次为算法的 4 个最优解, 通过使用提出的自适应多策略选择, 4 个最优解通过扩大搜索找到 4 个新的个体, 将这 4 个个体与之前的 4 个个体组成一个新的种群  $D$ , 按照从小到大的顺序对个体的适应度进行排序, 选取前 4 个适应度依次赋值给  $\mathbf{X}_{\text{Attacker}}$ 、 $\mathbf{X}_{\text{Barrier}}$ 、 $\mathbf{X}_{\text{Chaser}}$ 、 $\mathbf{X}_{\text{Driver}}$ , 更新 4 个最优解的位置, 使算法在每次迭代过程中都能找到适应度最好的 4 个个体。

### 2.2.2 动态学习因子

在原始算法中, 其他黑猩猩被迫根据最佳黑猩猩的位置更新它们的位置。若在迭代过程中算法陷入局部最优, 其他黑猩猩的位置也会随着最优个体的位置陷入局部最优。为了解决这个问题, 在其他黑猩猩更新位置时结合原始算法提供的收敛因子, 加入动态学习因子<sup>[19]</sup>, 个体通过学习 4 个最优黑猩猩的位置在全局自由分布, 增加了个体位置的多样性, 避免算法过早陷入局部最优。该策略数学模型为:

$$n = 1 + \frac{t-1}{T}, \quad (25)$$

$$\mathbf{X}_i^d(t+1) = f\mathbf{X}_i^d(t) + \left( \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4}{4} - \mathbf{X}_i^d(t) \right) n \text{rand}(), \quad (26)$$

式中,  $T$  为最大迭代次数,  $n$  为动态学习因子,  $\text{rand}()$  为随机产生  $(0, 1)$  之间的随机数。

### 2.3 基于 XNOR/OR 的 MPRM 电路面积优化

MPRM 电路面积优化是典型的组合优化问题, 现有的面积优化方法存在搜索解质量比较差、不容易搜索到最优解等问题, 结合 1.4 节中提出的面积优化模型, 利用 AMSchOA 对 MPRM 电路面积进

行优化,搜索对应的最佳极性,以实现电路面积最小。对应面积优化流程如图2所示。

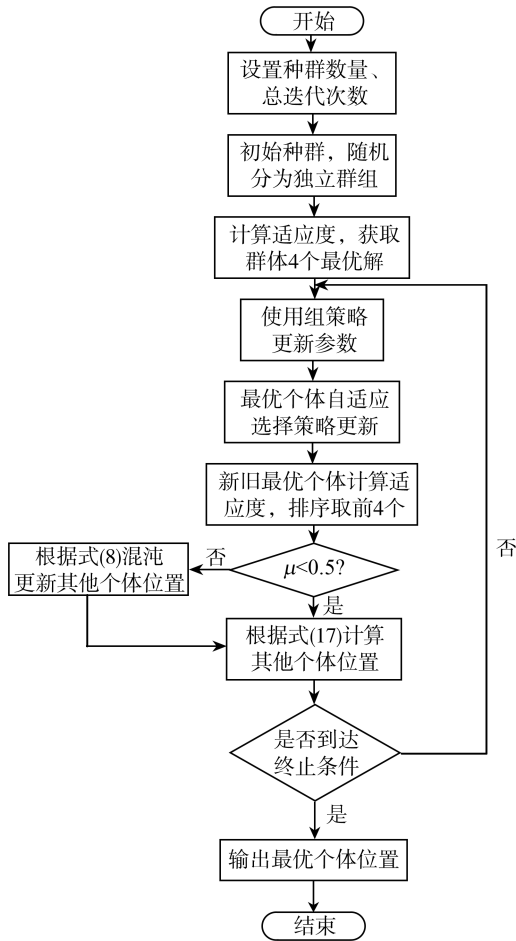


图2 电路面积优化流程

Fig.2 Circuit area optimization process

### 3 试验结果与分析

利用 MATLAB R2020b 实现对代码的编写,软

件运行环境为基于 Windows10 的操作系统、英特尔酷睿 i7-10700 CPU、32 GB RAM,测试电路采用北卡罗来纳微电子中心 (Microelectronics Center of North Carolina, MCNC) 基准测试电路。

为了验证所提 AMSChOA 在 MPRM 电路面积优化上的有效性,分别使用 ChOA、粒子群优化算法 (particle swarm optimization, PSO)<sup>[20]</sup>、改进粒子群优化算法 (modified particle swarm optimization, MPSO)<sup>[21]</sup> 3 种算法与 AMSChOA 进行对比。4 种算法的参数设置如表 2 所示,试验终止条件均为迭代次数达到 40 次。从 MCNC 基准测试电路中随机选取 12 个电路作为试验电路。同时,群智能算法具有随机性,为了充分说明所提算法的优越性,选择进行 10 次测试试验。

表 2 4 种算法的参数设置

Table 2 Parameterization of the four algorithms

算法	种群个数	迭代次数	个体变异位数	循环变异次数
ChOA	50	40		
PSO	50	40		
MPSO	50	40		
AMSChOA	50	40	2	10

注:空格为该项没有取值。

#### 3.1 电路面积对比

极性是指输入变量的一组取值。试验选取 3 种算法与 AMSChOA 进行对比,试验结果为在每个电路运行的平均面积,如表 3 所示。其中,  $S_{ave1}$  为 AMSChOA 与原始 ChOA 相比的面积节省率,  $S_{ave2}$  为 AMSChOA 与 PSO 相比的面积节省率,  $S_{ave3}$  为 AMSChOA 与 MPSO 相比的面积节省率。

表 3 不同电路中 4 种算法的平均面积

Table 3 Average area of four algorithms in different circuits

电路名称	输入位数	平均面积				$S_{ave1}/\%$	$S_{ave2}/\%$	$S_{ave3}/\%$
		ChOA	PSO	MPSO	AMSChOA			
Sqrt8	8	179.0	221.1	207.3	153.2	14.41	30.71	26.09
Prom2	9	54.8	87.1	69.6	40.5	26.09	53.50	41.81
Br1	12	112.5	82.2	83.2	75.3	33.06	8.39	9.49
T3	12	190.7	271.2	221.6	138.5	27.37	48.93	37.50
Newapla	12	51.3	65.4	67.3	33.1	35.47	49.83	50.81
Br2	12	71.0	46.9	50.2	35.2	50.42	24.94	29.88
Amd	14	717.2	1 000.5	922.3	505.4	29.53	49.48	45.20
Table3	15	3 319.6	2 180.4	2 302.5	1 465.7	55.84	32.77	36.34
Gary	15	1 248.5	1 783.8	1 809.6	986.3	21.00	44.70	45.49
Newtpla	15	111.1	174.1	175.2	55.9	49.68	67.89	68.09
Pdc	16	541.9	1 086.4	911.2	361.6	33.27	66.71	60.31
Table5	17	182.5	147.6	155.5	64.0	64.93	56.63	58.84

由表 3 可知,AMSChOA 在不同电路下得到的电路面积优于其他 3 种算法。通过分析试验数据可知:和 ChOA 相比,AMSChOA 的平均面积节省率为 36.76%;和 PSO 相比,AMSChOA 的平均面积节省率为 44.54%;和 MPSO 相比,AMSChOA 的平均面积节省率为 42.49%。对于不同输入位数及最小项的数量,AMSChOA 也具有更强的寻优效果。对于输入位数 10 位以下且最小项数量较少的电路,AMSChOA 在面积节省方面的效果较小;对于输入位数 10 位以上且最小项数量较多的电路,AMSChOA 在面积节省方面更加有效。这是由于随着输入位数的增加,AMSChOA 收敛速度较快且跳出局部最优解的能力较强。

为了更直观地显示所提算法的性能,AMSChOA 在每个电路运行 10 次的试验结果如表 4 所示。由表 4 可知:对于输入位数小于等于 12 位的电路,同一电路中 AMSChOA 的 10 次试验结果基本一致,相差不大,较为稳定;对于输入位数大于 12 位的电路,同一电路中 AMSChOA 的 10 次试验结果有所波动,但大体保持在平均值附近;对于不同的电路,随着输入位数增加,AMSChOA 搜索最优解的效果更好且波动较小。这是由于群智能算法在搜索最优解的过程中会出现每次搜索到的最优解不一样的现象,但改进后的 AMSChOA 通过增加自适应多策略选择方法及动态学习因子,加快搜索最优解,不易陷入局部最优。

表 4 AMSChOA 面积优化试验结果  
Table 4 Results of AMSChOA area optimization experiments

电路名称	输入位数	面积最优解									
		1 次循环	2 次循环	3 次循环	4 次循环	5 次循环	6 次循环	7 次循环	8 次循环	9 次循环	10 次循环
Sqrt8	8	155	152	152	152	155	152	155	152	152	155
Prom2	9	37	37	44	37	51	37	37	37	51	37
Br1	12	80	71	77	67	73	67	81	71	83	83
T3	12	119	157	133	150	144	158	140	124	133	127
Newapla	12	27	27	44	27	37	37	34	27	37	34
Br2	12	34	34	34	34	34	37	37	34	37	37
Amd	14	487	585	481	487	499	511	460	487	503	554
Table3	15	1 206	1 206	1 566	1 574	1 172	1 172	1 980	1 566	1 574	1 641
Gary	15	937	1 129	958	886	1 031	967	1 017	912	1 049	977
Newtpla	15	60	50	60	41	60	41	60	50	77	60
Pdc	16	372	384	331	481	322	310	435	317	279	385
Table5	17	65	85	68	77	64	44	64	65	64	44

### 3.2 电路运行时间对比

试验选取 3.1 节中相同的 3 个算法与 AMSChOA 进行运行时间对比,试验结果为每个算法在相同电路上运行 10 次的时间,如表 5 所示。其

中, $\Delta t_1$  为 AMSChOA 与 ChOA 相比的时间差, $\Delta t_2$  为 AMSChOA 与 PSO 相比的时间差, $\Delta t_3$  为 AMSChOA 与 MPSO 相比的时间差。

表 5 不同电路中 4 种算法的运行时间  
Table 5 Running time of four algorithms in different circuits

电路名称	输入位数	运行时间/s				$\Delta t_1$	$\Delta t_2$	$\Delta t_3$
		AMSChOA	PSO	MPSO	ChOA			
Clip	9	12.406	24.382	25.513	25.743	11.976	13.107	13.337
9sym	9	12.471	20.558	29.066	26.521	8.087	16.595	14.050
Ex1010	10	27.608	43.139	67.722	59.363	15.531	40.114	31.755
Alu3	10	21.951	49.480	55.586	52.004	27.529	33.635	30.053
Br1	12	16.804	12.673	77.579	40.022	-4.131	60.775	23.218
Br2	12	3.341	3.428	20.716	11.784	0.087	17.375	8.443
T3	12	28.558	22.488	13.037	17.399	-6.070	-15.521	-11.159
Newapla	12	131.789	76.999	149.217	150.932	-54.790	17.428	19.143
Table3	14	934.865	816.292	2 913.523	1 515.649	-118.573	1 978.658	580.784
Misex3	14	236.091	613.531	1 012.401	934.355	377.140	776.310	698.264
Newtpla	15	90.204	99.007	84.262	122.475	8.803	-5.942	32.271
Gary	15	1 721.858	2 690.063	2 505.906	2 541.431	968.205	968.205	784.048

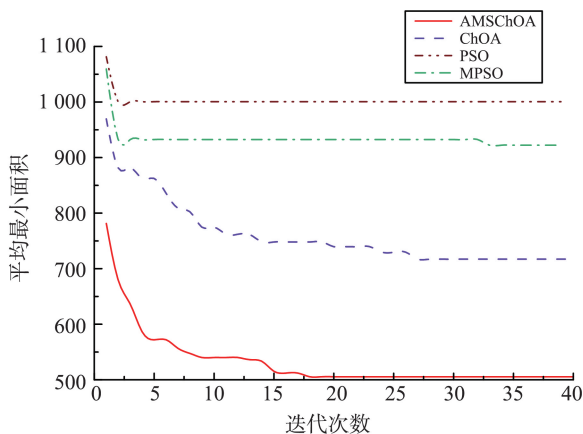
由表5可知:对于输入位数小于等于10的电路,AMSchOA在同一个电路上的运行时间均明显少于其他3个算法;对于输入位数大于10的电路,AMSchOA在同一个电路上的运行时间大于其他3种算法,但大多数运行时间较少;4种算法均采用迭代次数40次作为试验终止条件,在相同的迭代次数内,AMSchOA找到最优解所消耗时间最少,速度最快。若采用常规的根据误差结果判断是否终止试验,算法可能会因此陷入局部最优。

总体来看,与其他3种算法相比,AMSchOA在相同电路上的运行时间更少,这是由于针对种群中的最优解进行变异操作,扩大了算法搜索范围,提

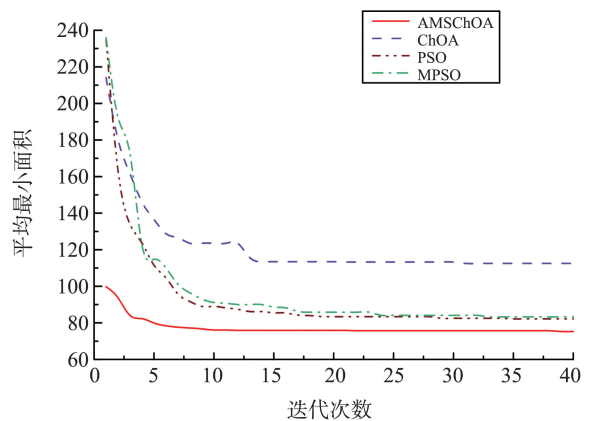
高了算法搜索精度;对于种群中的其他个体加入动态学习因子,避免算法在迭代过程中陷入局部最优解而停止搜索,且与PSO、MPSO相比,AMSchOA的搜索机制较为简单。

### 3.3 收敛性对比

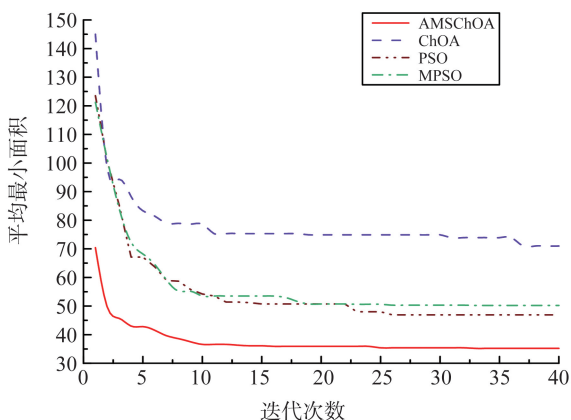
为了进一步说明所提算法的优越性,随机选取表3中的4个电路,计算算法在迭代过程中获得的面积最优解,得到每个算法在每个电路上运行10次的平均值,根据得到的数据绘制收敛曲线,如图3所示。从图3中可以看出,与其他3种算法相比,AMSchOA的收敛速度更快,找到的最优解结果最好。



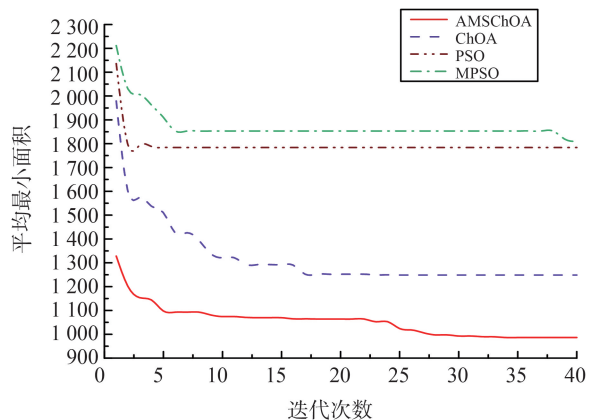
(a) Amd电路面积优化曲线



(b) Br1电路面积优化曲线



(c) Br2电路面积优化曲线



(d) Gary电路面积优化曲线

图3 面积优化曲线图

Fig.3 Area optimization graph

## 4 结论

基于XNOR/OR的MPRM电路面积优化属于三值组合优化问题,为解决现有的MPRM电路优化方法存在的搜索最小面积速度慢、不容易跳出局部最优等问题,本研究提出一种自适应多策略选择黑

猩猩优化算法AMSchOA。通过加入4种最优解搜索策略,在最优解附近进行搜索,提高算法的收敛速度;通过加入动态学习因子,对除最优解之外的黑猩猩个体位置进行更新,避免算法陷入局部最优。

基于AMSchOA提出一种MPRM电路面积优化方法,可以有效求解类似MPRM电路优化的三值组合优化问题,提高电路面积的搜索精度和

效率。

基于 MCNC 基准测试电路的试验结果表明,与基于 PSO、MPSO、ChOA 的 MPRM 电路面积优化方法相比,基于 AMChOA 的优化方法最高节省电路面积为 68.09%,平均为 41.24%,具有更高的搜索精度和更优的效果。

#### 参考文献:

- [1] 周宇豪,何振学,梁新艺,等. 基于 BABFA 的 XNOR/OR 电路面积优化[J]. 北京航空航天大学学报, 2022, 48(10): 2031-2039.  
ZHOU Yuhao, HE Zhenxue, LIANG Xinyi, et al. Optimization of XNOR/OR circuit area based on BABFA [J]. Journal of Beijing University of Aeronautics and Astronautics, 2022, 48(10): 2031-2039.
- [2] ZHOU Y H, HE Z X, JIANG J H, et al. Fast area optimization approach for XNOR/OR-based fixed polarity Reed-Muller logic circuits based on multi-strategy wolf pack algorithm[J]. ACM Transactions on Design Automation of Electronic Systems, 2023, 28(3): 1-16.
- [3] 刘庆鑫,齐琦,贾鹤鸣,等. 混合改进策略的阿奎拉优化算法[J]. 山东大学学报(工学版), 2023, 53(4): 93-103.  
LIU Qingxin, QI Qi, JIA Heming, et al. Aquila optimizer based on hybrid improved strategies[J]. Journal of Shandong University (Engineering Science), 2023, 53(4): 93-103.
- [4] HE Z X, PAN Y H, WANG K J, et al. Area optimization for MPRM logic circuits based on improved multiple disturbances fireworks algorithm [J]. Applied Mathematics & Computation, 2021, 399: 126008.
- [5] 黄华娟,程前,韦修喜,等. 融合 Jaya 高斯变异的自适应乌鸦搜索算法[J]. 山东大学学报(工学版), 2023, 53(2): 11-22.  
HUANG Huajuan, CHENG Qian, WEI Xiuxi, et al. Adaptive crow search algorithm with Jaya algorithm and Gaussian mutation [J]. Journal of Shandong University (Engineering Science), 2023, 53(2): 11-22.
- [6] 何俊才,何振学,王福顺,等. 基于 ERWOA 的多输出 MPRM 电路面积优化[J]. 北京航空航天大学学报, 2023, 49(5): 1193-1200.  
HE Juncai, HE Zhenxue, WANG Fushun, et al. Circuit area optimization of multi-output MPRM based on ERWOA algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics, 2023, 49(5): 1193-1200.
- [7] 俞海珍,蒋志迪,汪鹏君,等. GA-DTPSO 算法及其在混合极性 XNOR/OR 电路面积优化中应用[J]. 计算机辅助设计与图形学学报, 2015, 27(5): 946-952.  
YU Haizhen, JIANG Zhidi, WANG Pengjun, et al. GA-DTPSO algorithm and its application in area optimization of mixed polarity XNOR/OR circuits [J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(5): 946-952.
- [8] KHISHE M, MOSAVI M R. Chimp optimization algorithm [J]. Expert Systems with Applications, 2020, 149: 113338.
- [9] DU N T, LUO Q F, DU Y Q, et al. Colorimage enhancement: a metaheuristic chimp optimization algorithm [J]. Neural Processing Letters, 2022, 54(6): 4769-4808.
- [10] PASHAEI Elnaz, PASHAEI Elham. An efficient binary chimp optimization algorithm for feature selection in biomedical data classification [J]. Neural Computing and Applications, 2022, 34: 6427-6451.
- [11] SARA K, ABDOLAH C. Chimp optimization algorithm to optimize a convolutional neural network for recognizing Persian/Arabic handwritten words [J]. Mathematical Problems in Engineering, 2022, 2022: 4894922.
- [12] YANG C, SU C, HU H T, et al. Performance optimization of photovoltaic and solar cells via a hybrid and efficient chimp algorithm [J]. Solar Energy, 2023, 253: 343-359.
- [13] 季雨瑄,叶军,杨震宇,等. 结合分辨矩阵改进的邻域粗糙集属性约简算法 [J]. 山东大学学报(工学版), 2022, 52(4): 99-109.  
JI Yuxuan, YE Jun, YANG Zhenyu, et al. An improved neighborhood rough set attribute reduction algorithm combined with resolution matrix [J]. Journal of Shandong University (Engineering Science), 2022, 52(4): 99-109.
- [14] 宁春梅,孙博,肖敬先,等. 基于时间感知注意力机制的混合编码网络方法 [J]. 山东大学学报(工学版), 2022, 52(2): 23-30.  
NING Chunmei, SUN Bo, XIAO Jingxian, et al. The method of hybrid code networks based on time-aware attention mechanism [J]. Journal of Shandong University (Engineering Science), 2022, 52(2): 23-30.
- [15] VERDU S. The Cauchy distribution in information theory [J]. Entropy, 2023, 5(2): 346.
- [16] DOS SANTOS M A F, MENON JR L, ANTENEODO C. Efficiency of random search with space-dependent diffusivity [J]. Physical Review E, 2022, 106(4): 044113.