结合Ising模型的AEIPSO算法优化数字组合 逻辑电路

王瑞祥1,曾红丽2*

¹南京邮电大学电子与光学工程(柔性电子、未来技术)学院,江苏南京 ²南京邮电大学理学院,江苏南京

收稿日期: 2025年1月28日; 录用日期: 2025年2月21日; 发布日期: 2025年2月28日

摘要

粒子群优化算法(PSO)源于人工生命和复杂的自适应系统,近年来在数字电路的设计和优化中得到了应用。人们通过参数调节来改进PSO算法,虽然PSO算法具有参数更少、收敛速度更快等优点,但在迭代进 化过程中容易陷入局部最优解,从而导致计算资源的低效使用。针对这一局限性,我们引入了基于Ising 模型的改进自适应增强粒子群优化算法(AEIPSO)。提高了进化迭代的效率并且增强了解的多样性。同时, 它保留了从PSO算法的快速收敛和卓越的全局搜索能力。实验结果表明,AEIPSO算法在组合逻辑电路的 设计和优化方面优于其他PSO算法。

关键词

组合逻辑电路,优化设计,伊辛模型,混合算法

The AEIPSO Algorithm Based on Ising Model Optimizes the Digital Combinational Logic Circuit

Ruixiang Wang¹, Hongli Zeng^{2*}

¹School of Electronic and Optical Engineering (Flexible Electronics, Future Technology), Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

²College of Science, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu

Received: Jan. 28th, 2025; accepted: Feb. 21st, 2025; published: Feb. 28th, 2025

*通讯作者。

Abstract

Particle swarm optimization (PSO), derived from artificial life and complex adaptive systems, has been applied in the design and optimization of digital circuits in recent years. PSO algorithm is improved by parameter adjustment. Although PSO algorithm has the advantages of fewer parameters and faster convergence, it is easy to fall into local optimal solution during iterative evolution, which leads to inefficient use of computing resources. To address this limitation, we introduce an improved adaptive enhanced particle swarm optimization algorithm (AEIPSO) based on Ising model. It improves the efficiency of evolutionary iterations and increases the diversity of understanding. At the same time, it retains the fast convergence and excellent global search capability from the PSO algorithm. Experimental results show that AEIPSO algorithm is superior to other PSO algorithms in combinatorial logic circuit design and optimization.

Keywords

Combinatorial Logic Circuit, Optimal Design, Ising Model, Hybrid Algorithm

Copyright © 2025 by author(s) and Hans Publishers Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). http://creativecommons.org/licenses/by/4.0/

1. 引言

组合逻辑电路(CLCs)是一种数字电路,其输出只取决于当前输出[1]。传统的人工设计数字电路方法, 首先根据待实现电路的特定的逻辑功能需求而制定唯一的电路真值表。然后利用卡诺图或 Quinn-McCluskey 简化方法[2]进行电路设计。人工设计的效率不仅与电路规模有关,还与设计者的经验密切相关。 同时,具有同一逻辑功能的组合逻辑电路可以对应多种不同形式的电路实现方案。这使得电路设计成为一 个优化问题,即用最少的门来实现相同的功能[3]。为了减少人力资源,研究人员倾向于开发自动化方法。 一个较大的分支是演化电路,通常具有更高的效率和精确度[4]。粒子群算法(PSO)[5]、遗传算法(GA)[6]、 灰狼算法(ACO)[7]等算法已广泛应用于数字 CLCs [8]的设计与优化中。

PSO 算法最早是由 J. Kenney 与 R. Eberhart 通过研究鸟群使用信息共享来进行觅食、探寻栖息地等行为,而提出的一种基于群的优化算法,算法被广泛应用于多种优化领域[9]。考虑到数字 CLCs 的众多设计方案,寻找具有最小门级的最优解符合 PSO 算法的探索策略。Coello 将 PSO 算法扩展到 CLCs 的设计中,验证了其优于手工设计方法和遗传算法[10]。Sagar 进行了对比分析,评估了 PSO、遗传算法(GA)和蚁群算法(ACO)在设计 CLCs [11]时的性能差异。实验结果表明,PSO 算法能够在最短的时间内,以最少的电路门数高效地找到满足设计要求的可行解。

与遗传算法等启发式算法相比,经典的 PSO 算法具有收敛速度快、迭代次数少的特点。然而,这种 效率是以整个迭代过程中粒子解的多样性减少,即增加了陷入局部最优解的可能性为代价的。因此,这增 加了与电路演化设计相关的时间和计算资源成本。将物理模型中的伊辛模型(Ising model)与改进的自适应 增强粒子群优化算法(AEPSO)[12]-[14]集成是一种缓解 PSO 算法缺陷的策略。在 AEPSO 算法中,粒子的 新位置和速度的更新方向,依赖于粒子个体历史最优位置以及种群中多个适应度较优层次中的粒子引导。 这种进化策略使 AEPSO 算法对解空间有更广泛的探索,并增强了解的多样性。此外,AEPSO 算法[15]-[18]采用非线性方法自适应调整加速系数,提高了算法的收敛速度和精度。这种自适应调整加快了算法寻 找最优解的速度,从而提高了算法的整体效率。

与标准 PSO 算法在迭代完成之后抛弃粒子较差解的做法不同,我们在 AEPSO 算法的基础上进一步引入了伊辛模型和 Metropolis 采样算法,用于解决粒子如何以较少的迭代次数跳出局部最优解的问题以及粒子在迭代过程中如何处理迭代过程中较差解的问题。之前就曾有研究将伊辛模型引入到 PSO 算法中,最终的实验结果相较于直接使用 PSO 算法的实验结果更加出色[19]。综上所述,本文提出一种全新的算法,我们将其称为 AEIPSO 算法。

本文其余部分组织如下:第二节我们将详细介绍本文采用的 AEIPSO 算法的原理,第三节将详细介绍 AEIPSO 算法对组合逻辑电路的优化设计过程,第四节将介绍 CEC 基准优化函数集实验的过程,第五节 是实际组合逻辑电路优化设计过程。第六节是本文的结论和讨论。

2. AEIPSO 算法的原理

AEIPSO 算法首先初始化一组粒子,每个粒子都有对应的初始速度与位置。迭代过程中,粒子的速度 将会随之变化,粒子当前位置变化由粒子当前的速度和先前位置共同影响,算法会根据伊辛模型能量公式 计算并比较粒子当前位置和先前位置对应的能量值,得出个体最优解对应位置 Pheer。

根据种群中粒子的能量大小分布,将粒子的解分为四个梯度层。筛选前三个层次的最优解 $P_{g_{(1,2,3)}}$ 参与引导粒子在解空间中的搜索方向。随后,根据公式(1),粒子当前位置 $X_i(t)$,前三个能量分布层次的最优解 $P_{g_{(1,2)}}$ 以及公式(2)中的收敛因子A计算前三个层次对应的种群粒子参考位置 $X_{ref_{(1,2)}}$ 。

粒子更新位置和速度向量遵循公式(4)和公式(3),其中公式(3)中的 ω 代表惯性权重,其值随着迭代次数的增加而变化, r₁~r₃代表 0 到 1 之间的随机数。

$$X_{ref_{(1,2,3)}} = P_{g_{(1,2,3)}}(t) - A^* \left| C_{(1,2,3)}(t)^* P_{g_{(1,2,3)}}(t) - \omega^* X_i(t) \right|$$
(1)

$$A = (2 - Iter * (2/Iter_{max})) * (2 * r_1 - 1)$$
(2)

$$V_{i}(t+1) = \omega^{*} \left(V_{i}(t) + C_{1}(t)^{*} r_{1}^{*} \left(X_{ref_{1}}(t) - X_{i}(t) \right) + C_{2}(t)^{*} r_{2}^{*} \left(X_{ref_{2}}(t) - X_{i}(t) \right)$$
(3)

$$+C_{3}(t)*r_{3}*(X_{ref_{3}}(t)-X_{i}(t))+C_{4}(t)*r_{4}*(P_{best?}(t)-X_{i}(t)))$$

$$X_{i}(t+1) = X_{i}(t) + V_{i}(t+1)$$
(4)

 $X_i(t+1)$ 为粒子迭代后的位置, $V_i(t+1)$ 为粒子迭代后的速度值。不同的是,AEIPSO 算法在粒子在 进行位置更新时,同时考虑粒子个体认知部分和粒子社会认知部分,从而使种群中粒子之间的信息共享更 加充分。最后,我们在 AEIPSO 算法的速度更新和参考位置更新公式中引入了随粒子能量值变化而动态变 化的加速度系数 $C_{(1,2,3)}(t)$,从而根据粒子所处迭代更新阶段的不同,进行加速度系数的动态调整。在解空 间搜索的初始阶段,由于粒子探索解的情况较少,所以应该有较大的 $C_4(t)$ 和较小的 $C_{(1,2,3)}(t)$,有利于粒 子在整个解空间内搜索,在解空间搜索的后期阶段,为使粒子探索解加速收敛,则应该有较小的 $C_4(t)$,和 较大的 $C_{(1,2,3)}(t)$,具体公式如下。

$$C_{1,2,3}(t) = 1 - \exp\left(-a\left|E_a(t) - E_g(t)\right|\right)$$
(5)

$$C_4(t) = 1 - C_1(t) \tag{6}$$

 $E_a(t)$ 为种群中所有粒子能量值的平均值, $E_g(t)$ 为每次迭代更新后粒子群中最优解位置对应的粒子能量值, a 为控制系数。 ω_{\min} 为最小惯性权重, ω_{\max} 为最大惯性权重。Iter 为当前迭代次数, Iter_{max}为最大迭代次数。惯性权重 ω 的计算公式如下所示:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \frac{(\text{Iter}_{\max} - \text{Iter})}{\text{Iter}_{\max}}$$
(7)

为了解决 AEIPSO 算法中的粒子的较差解的处理问题,我们结合 Ising 模型和 Metropolis 采样算法, 让粒子以一定概率接受较差解,增加解的多样性,减少粒子跳出局部最优解需要的迭代次数。这一方法的 思路主要源于寻找伊辛模型最低能量构型过程和寻找组合优化问题的全局最优解过程之间的映射[18],将 种群中粒子对应的电路方案映射为伊辛模型,伊辛模型中的能量就相当于组合优化问题中的目标函数,而 伊辛模型中的最低能量构型就相当于组合优化问题中的全局最优解[19]。在伊辛模型内部粒子的之间存在 相互作用,模型中不同方向粒子之间的相互作用形成了伊辛模型的整体能量。本文中的每种电路输入组合 对应的电路输出映射为伊辛模型的粒子之间的相互作用,而粒子之间的相互作用形成了电路方案映射成的 伊辛模型的整体能量。在本文中,粒子的能量函数计算公式如下:

$$E_{a}(t) = \frac{1}{N} \sum_{i=1}^{N} E_{(S_{i})}(X_{i}(t))$$
(8)

$$E_g(t) = E_{(S_i)}\left(P_{g_1}(t)\right) \tag{9}$$

$$E_{(S_i)} = -J \sum_{\langle i,j \rangle} S_i S_j \tag{10}$$

$$F_{(S_i)} = \sum_{\langle i,j \rangle} S_i S_j \tag{11}$$

 $E_a(t)$ 是种群粒子能量的平均值, $E_g(t)$ 是每次迭代后种群最优位置对应的能量值, 两者的公式如公式(8)和公式(9)所示, $E_{(s_i)}$ 为第 *i* 个粒子对应的组合逻辑电路方案的能量值, *J* 为能量耦合常数, $F_{(s_i)}$ 为 PSO 算法实验中第 *i* 个粒子对应的适应度值。 $F_{(s_i)}$ 的计算规则类似于 $E_{(s_i)}$ 计算规则, 但不具有交换相互作用常数 *J*。*S_i*和*S_j*分别为进化电路真值表的输出值和目标真值表的输出状态(在数字电路中系统的输出状态只有两种分别为: 1 和 0), 相当于伊辛模型中自旋粒子的两种状态。如果进化电路的真值表与目标真值表完全对应, 且 J 为正,则粒子的能量将是数值范围内的最优值。

粒子在迭代完成后是否接受更新后的位置值要根据两条判断原则进行判断,如果粒子在位置更新后对 应的能量值减小,则接受更新后的位置。若更新后的能量值未减小但计算 Metropolis 采样算法中概率值 *P*_(si)大于 rand(0,1)的随机数则同样接受该位置变化。其中 Δ*E* 为粒子迭代前后的能量值差值,*K* 为常数, *T* 为调整概率变化的温度值,T 的值随着迭代次数的增加而发生改变,迭代更新次数增大,*T* 值变小,当 能量差值为正值时,代表能量增长为负方向。如果此时处在粒子迭代变化前期,因为种群粒子探索的解空 间为多峰解空间,峰值的不同对应的较优解不用。所以我们更希望提高前期接受较差解的概率,使粒子能 够充分探索解空间,探索不同峰值对应的解。判断原则以及概率计算公式如公式(12)所示。

$$x_{i}(t+1) = \begin{cases} x_{i}(t+1)', & E_{(sj)} < E_{(si)} \\ x_{i}(t+1)', & E_{(sj)} > E_{(si)} \text{ and } p_{(si)} > \text{random}(0,1) \\ x_{i}(t), & E_{(sj)} > E_{(si)} \text{ and } p_{(si)} <= \text{random}(0,1) \end{cases}$$
(12)
$$p(s_{i}) = \exp\left(-\frac{\Delta E}{KT}\right)$$
(13)

在每一次迭代过程中温度值 T 改变之后引起的粒子位置更新,就相当于伊辛模型中粒子构型随着温度 T 的变化而改变。其中温度值 T 的计算公式如下,初始值 T_0 取决于 $|E_{max} - E_{min}|$:

$$T = T_0 * \left[(\text{Iter}_{\text{max}} - \text{Iter}) / \text{Iter}_{\text{max}} \right]$$
(14)

3. AEIPSO 算法用于电路设计的过程

本文利用 MATLAB 对粒子的初始化、更新迭代、能量计算等过程进行程序编写和运行,最后根据电路矩阵编码规则(表 1 和表 2)以及迭代完成得到的电路矩阵直接获得满足设计要求的数字组合逻辑电路。

每次实验开始时,首先初始化一组数量为*N*的粒子,每个粒子有对应的初始速度与初始位置。初始位置按照一定的矩阵编码规则生成,初始速度为零矩阵,速度矩阵规模与位置矩阵相同。每一个粒子的位置矩阵都代表一种组合逻辑电路方案。电路矩阵的输入信号和逻辑运算编码规则如表1、表2所示:

矩阵第一列和第三列的矩阵元素值	输入信号
1	А
2	\overline{A}
3	В
4	\overline{B}
5	С
6	\overline{C}
7	L11
8	L12
9	L13
10	L21
11	L22
12	L23
13	L3

Table 1. Matrix input signal coding rules 表 1. 矩阵输入信号编码规则

 Table 2. Encoding rules for matrix logic operations

 表 2. 矩阵逻辑运算编码规则

矩阵第二列矩阵元素值	逻辑门类型
1	AND
2	OR
3	XOR
4	NOT
5	WIRE

将组合逻辑电路编码为矩阵,矩阵的每一行代表一种数字组合逻辑电路的逻辑操作,其中每一行的第一列元素代表第一位输入,第三列元素代表第二位输入,第二列元素代表逻辑运算。其中逻辑运算共有五种分别为:1代表逻辑与、2代表逻辑或、3代表逻辑异或、4代表逻辑非、5代表导线连接,而输入量由数字1~13代替。每次迭代更新时,程序将遍历矩阵每一行并进行元素更新。我们在程序编写过程中为每

一行的元素设定了对应的上下限,保证矩阵元素在更新迭代过程中不会超出取值范围。

电路矩阵中的每一行相当于组合逻辑电路结构中不同的层级,每个层级之间相互联系,上一个层级的 逻辑输出可能作为下一个层级的逻辑输入。具体组合逻辑电路结构图如图1所示,其中 INPUT 层代表电 路的输入端,OUTPUT 层代表电路的输出端。图1中每一个网格代表组合逻辑电路的一个逻辑单元,每个 逻辑单元有两个输入,一个输出,其中L1层的输入来自电路的输入端输入,而L2层、L3层的输入来自 上一级的输入。例如:L1层包含L11、L12、L13逻辑单元,其它层同理。逻辑单元的数目、层级的数目 以及电路输入端输入量的数目不固定,可以根据自己的需求添加输入量数目,因为电路的层级之间相互联 系,因此当输入量发生变化时,其它层级也要随之发生变化。



Figure 1. Schematic diagram of a combinational circuit connection. The INPUT and OUTPUT layers represent the input and output terminals of the circuit, and L1, L2, and L3 represent three layers with three logic units in each layer 图 1. 组合电路连接形式图。INPUT、OUTPUT 层分别代表电路的输入端和输出端, L1、L2、L3 分别代表三个层级, 每个层级分别有三个逻辑单元

粒子初始化完成之后,根据公式(10)计算每个粒子位置对应的能量值并将其设定为初始个体最优位置 P_{best}。

并将粒子按照能量分为四个层次,挑选前三个层次中个体最优位置设为初始种群最优位置 Pg(12)。

随后根据位置更新迭代公式(4)和速度迭代更新公式(3)对每个粒子相应的速度矩阵和位置矩阵进行更 新迭代,并根据粒子能量计算公式(10)计算迭代后种群中每个粒子的能量值大小,直到得到理想输出值或 迭代达到最大迭代次数为止。

本文参考以往研究人员的有关粒子群优化算法参数选择的论文[19]。按照本文的应用环境对应的参数 选择以及本文具体的研究目的,在实验的初始化部分对更新迭代公式中的一些固定参数和变化参数的初始 值进行的设置,其中种群中每个粒子的初始速度*V_i*为0,惯性权重最大值 *ω_{max}*为0.9,惯性权重最小值 *ω_{min}*为 0.4,最大迭代次数 Iter_{max}为 20,控制系数 *a* 设置为1,加速度系数 *C*1和加速度系数 *C*2的初始值都设 为 2, *r*₁-*r*₃ 设置为 0~1 的随机数,其数值在每次迭代过程随机产生。实验中采用的对比算法 PSO, IPSO 算法同属于粒子群优化算法,为了公平体现算法之间的差异,因此采用相同的参数设置,上述参数取值采 用了论文[19]中经过大量实验所得出推荐参数范围内。

4. CEC 基准优化函数集实验

为了进一步比较本文提出的 AEIPSO 算法与其他粒子群优化算法的性能差异。使用 MATLAB 2020a 对 cec2020 中的基准优化函数测试集以及针对实际问题的优化函数测试集中的电子电路优化问题进行测试,并对比了 AEIPSO、IPSO 和 PSO 算法在上述优化函数上运行的结果。用于测试的计算机配置包括以

下规格:软件环境是 Windows 10 (64 位),硬件环境是 Intel(R) Core (TM) i7-8750H CPU @ 2.20 GHz, 8.0 GB RAM。

4.1. CEC 基准问题

为了充分反映 AEIPSO 算法与其他算法的在 CEC2020 基准优化函数测试集上的性能表现差异,我们 选取 CEC2020 测试函数集中的单峰、多峰、混合和组合类型函数,从收敛速度、解的多样性以及对复杂 问题的求解性能等方面对算法的性能进行测试,测试函数的详细信息如表 3 所示。

Table 3. Baseline optimization functions 表 3. 基准优化函数

函数编号	搜索范围	维度	优化目标值
F2	[-100, 100]	10	1100
F5	[-100, 100]	10	1700
F6	[-100, 100]	10	1600
F9	[-100, 100]	10	2400
F10	[-100, 100]	10	2500

4.2. 实验过程

为了尽量减少实验偶然性对实验结果的影响,我们将每个算法的程序运行 100 次,取 100 次运行后, 平均最优函数值随迭代次数变化的收敛曲线用来与其他算法作对比。此外,迭代次数的设置与算法结果在 每个基准优化测试函数中的收敛程度有关,我们在实验开始前,为每个算法设置足够的迭代次数,使每种 算法都能收敛到各自的稳定且最优函数值。

4.3. 实验结果和讨论

从附录 A 的结果图中可以看出, AEIPSO 在收敛速度上比其他算法有很大的优势, 并且在除 F10 之外的所有测试函数上都取得了更好的最终优化结果。在 F10 测试函数中, 虽然 AEIPSO 取得的最终优化结果 与其他算法相差不大, 但仍有优势。总体而言, 与其他算法相比 AEIPSO 算法表现出最好的性能。

5. 组合逻辑电路优化设计实验

为了评估 AEIPSO 算法在组合逻辑电路优化设计方面的性能,进行了三个实际组合逻辑电路优化设计 实验。这些实验包括一个数据选择器、一个全加法器和一个四输入一输出电路。

实验过程中,通过统计 IPSO、PSO 和 AEIPSO 三种算法的适应度值以及能量值(AEIPSO、IPSO)随着 迭代次数的变化,来检验和比较它们在数字组合逻辑电路设计中的性能。

5.1. 实验建立过程

 Table 4. Data selector circuit truth table

 表 4.
 数据选择器电路真值表

Inputs			Outputs
A	В	С	Out
0	0	0	0

4	续表				
	0	1	0	1	
	1	0	0	0	
	1	1	0	1	
	0	0	1	0	
	0	1	1	0	
	1	0	1	1	
	1	1	1	1	

以数据选择器电路为例,我们在仅给出表 4 真值表的前提下分别通过执行 AEIPSO、IPSO、PSO 算法 程序得出具体的数据选择器电路。

在本文实验中,我们首先根据编码规则初始化 50 个粒子。其中六个粒子 P1 到 P6 的位置矩阵具体形 式如图 2 所示:

()									
10	3	1)	(1	0	4`)	(4	0	1)
6	5	1	1	0	0		3	2	6
6	4	3	5	1	4		4	4	1
0	5	7	9	2	0		7	4	8
9	0	8	7	4	9		9	0	8
9	4	8	0	2	7		0	5	7
(11	3	8)	(10	1	11)	7	1	11)
	P1			Р2				Р3	
(6	1	2	(3	2	0)		(0	0	3)
					I				
1	2	4	1	1	5		6	4	0
1 2	2 3	4 5	1 3	1 3	5 1		6 5	4 0	0 6
1 2 8	2 3 1	4 5 9	1 3 7	1 3 4	5 1 8		6 5 7	4 0 2	0 6 9
1 2 8 9	2 3 1 3	4 5 9 7	1 3 7 8	1 3 4 0	5 1 8 7		6 5 7 9	4 0 2 1	0 6 9 0
1 2 8 9 0	2 3 1 3 4	4 5 9 7 8	1 3 7 8 0	1 3 4 0 0	5 1 8 7 9		6 5 7 9 0	4 0 2 1 3	0 6 9 0 9
1 2 8 9 0 12	2 3 1 3 4 3	4 5 9 7 8 0	1 3 7 8 0 11	1 3 4 0 0 5	5 1 8 7 9 9)		6 5 7 9 0 9	4 0 2 1 3 5	0 6 9 0 9 7

Figure 2. AEIPSO algorithm initializes the circuit matrix 图 2. AEIPSO 算法初始化电路矩阵

如表 5 所示,每个粒子使用公式 10 计算粒子初始能量值。在实际过程中,种群的每个粒子在初始化 过程都需要经过位置矩阵初始化和计算初始能量值的过程。

表 5. AEIPSO 算法初始粒子能量值							
	P1	P2	Р3	P4	Р5	P6	
Energy	8	8	8	8	0	8	

Table 5. Initial particle energy value of AEIPSO algorithm 表 5. AEIPSO 算法初始粒子能量值

剩余两种组合逻辑电路的实验建立过程与数据选择器电路实验建立过程一致,不同的地方在于期望电路的真值表不同,具体期望电路真值表分别如表 6、表 7 所示:

王瑞祥,曾红丽

	Inputs			Outputs			
А	В	С	D	Y			
0	0	0	0	1			
0	0	0	1	1			
0	0	1	0	0			
0	0	1	1	0			
0	1	0	0	1			
0	1	0	1	1			
0	1	1	0	1			
0	1	1	1	1			
1	0	0	0	0			
1	0	0	1	0			
1	0	1	0	1			
1	0	1	1	0			
1	1	0	0	0			
1	1	0	1	0			
1	1	1	0	0			
1	1	1	1	1			

Table 6. Four-input-one-output circuit truth table 表 6. 四输入一输出电路真值表

Table 7. Full adder circuit truth table 表 7. 全加器电路真值表

Inputs			Outputs		
A	В	С	Sum	Carry	
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	

以数据选择器电路为例,在本实验中, IPSO 算法和 PSO 算法对应的初始粒子位置矩阵所对应的初始 能量值分别如表 8、表 9 所示:

Table 8. Initial particle energy value of IPSO algorithm 表 8. IPSO 算法初始粒子能量值

	P1	P2	Р3	P4	Р5	P6
Energy	8	0	8	0	8	2

TILOTI

表 9. PSO 算法初始粒子适应度值表						
	P1	P2	Р3	P4	P5	P6
Energy	-8	-8	-8	0	-8	0

• .1

CDCO 1

5.2. 组合逻辑电路设计实验过程

a

. 1 . .

如果粒子初始化过程完成后,没有粒子达到最小能量值和最大适应度值。然后,根据算法的速度和 位置更新规则,在每次迭代过程中更新粒子的位置和速度,并计算每个粒子的能量或适应度值;直到达 到最小能量值和最大适应度值,或达到实验中设置的最大迭代次数 Itermax 时,算法程序的执行过程才会 终止。

以下是 AEIPSO、IPSO 和 PSO 算法在不同实验中通过迭代更新得到的实验结果。为降低实验结果的 偶然性,我们也重复进行了多次实验,得出多次运行下的,平均适应度或能量值随迭代次数的变化图。

本节将先给出单次运行下得出的粒子能量值以及对应的满足最优适应度值和能量值的电路方案。随 后在本文的附录 B 给出多次运行下的平均适应度值和能量值随迭代次数变化的曲线图。

5.2.1. 数据选择器电路

1. AEIPSO 算法:

Table 10. Energy values of particles after the 4th iteration

在 AEIPSO 算法的实验中, 粒子 P2 经过 4 次迭代后达到最小能量。其相应的电路图如图 3 所示。同时, P1~P6 粒子的能量值如表 10 所示。



Figure 3. Using AEIPSO algorithm, the circuit optimization results are updated after 4 iterations, and the left side of the circuit shows the updated circuit matrix iteration results

图 3. 使用 AEIPSO 算法, 电路优化结果在 4 次迭代后更新, 电路左侧为更新后的电路矩阵迭代结果

2. IPSO 算法:

在 IPSO 算法的实验中, 粒子 P4 经过 11 次迭代后达到最小能量。P4 相应的电路图如图 4 所示。同时, P1~P6 粒子的能量如表 11 所示。



Figure 4. IPSO algorithm is used to update the circuit optimization results after 11 iterations, and the left side of the obtained circuit is the updated circuit matrix iteration results

图 4. 采用 IPSO 算法,经过 11 次迭代更新电路优化结果,得到电路的左侧是更新后的电路矩阵迭代结果

3. PSO 算法:

在 PSO 算法的实验中, 粒子 P4 经过 18 次迭代后达到最大适应度值, 满足最大适应度值的粒子 P4 的电路图及电路矩阵如图 5 所示, 粒子 P1~P6 的适应度值如表 12 所示。



Table	12.	Fitness of particles after the 18th iteration	
表 12.	第	18 次迭代后粒子的适应度	

Figure 5. PSO algorithm is used to update the circuit optimization results after 18 iterations, and the left side of the obtained circuit is the updated circuit matrix iteration results 图 5. 采用 PSO 算法,经过 18 次迭代更新电路优化结果,得到电路的左侧是更新后的电路矩阵迭代结果

3. 木田 [30 异本, 红色 10 八达代史制电磁优化结末, 特到电磁的左侧定更制度的电磁定阵达代结末

实验中使用的三种算法的适应度值或能量值随种群迭代次数变化的整体过程,如图6所示。



Figure 6. Schematic diagram of IPSO, PSO, and AEIPSO algorithm iterative process. Where, the abscissa is the number of iterations of the algorithm, and the left and right axes of the ordinate are the energy value and fitness value of the particles corresponding to the number of iterations

图 6. IPSO、PSO 和 AEIPSO 算法迭代过程示意图。其中,横坐标为算法的迭代次数,纵坐标的左右轴分别为迭代次 数所对应的粒子的能量值和适应度值

5.2.2. 四输入一输出电路

1. AEIPSO 算法:

在 AEIPSO 算法的实验中, 粒子 P6 经过 28 次迭代后达到最小能量。其相应的电路图如图 7 所示。 同时, P1~P6 粒子的能量值如表 13 所示。

	P1		P2	P3	P4	Р5	P6
Energy	0		-4	2	-12	-11	-16
	$ \begin{pmatrix} 3 & 3 \\ 18 & 1 \\ 1 & 0 \\ 6 & 2 \\ 7 & 1 \\ 1 & 2 \\ 10 & 3 \end{pmatrix} $	18 5 2 7 6 9				Out	

 Table 13. Energy values of particles after iteration 28

 表 13. 第 28 次迭代后粒子的能量值

Figure 7. Using AEIPSO algorithm, the circuit optimization results are updated after 28 iterations, and the left side of the circuit shows the update iteration results of the circuit matrix

图 7. 使用 AEIPSO 算法, 电路优化结果经过 28 次迭代后更新, 电路左侧为电路矩阵的更新迭代结果

2. IPSO 算法:

在 IPSO 算法的实验中, 粒子 P5 经过 39 次迭代后达到最小能量。其相应的电路图如图 8 所示。同时, P1~P6 粒子的能量值如表 14 所示。

	P1		P2	Р3	P4	P5	P6
Energy	0	0		-14	-2	-16	-10
	$ \begin{pmatrix} 18 \\ 5 \\ 1 \\ 8 \\ 10 \\ 7 \\ 11 \end{pmatrix} $	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$				- Out	

 Table 14. Energy values of particles after the 39th iteration

 表 14. 第 39 次迭代后粒子的能量值

Figure 8. The circuit optimization results are updated after 39 iterations using the IPSO algorithm, and the left side of the circuit shows the update iteration results of the circuit matrix 图 8. 使用 IPSO 算法,电路优化结果经过 39 次迭代后更新,电路左侧为电路矩阵的更新迭代结果

3. PSO 算法:

在 PSO 算法的实验中, 粒子 P3 经过 48 次迭代后达到最大适应度。其相应的电路图如图 9 所示。同时, P1~P6 粒子的适应度如表 15 所示

Table	15. Fitness values of particles after the 48th iteration
表 15.	第48次迭代后粒子的适应度值

	P1	P2	Р3	P4	P5	P6
Energy	2	10	16	2	13	0
	$ \begin{pmatrix} 3 & 1 \\ 15 & 1 \\ 15 & 1 \\ 8 & 2 \\ 2 & 2 \\ 1 & 2 \\ 11 & 3 \end{pmatrix} $	$ \begin{array}{c} 19\\ 4\\ 5\\ 7\\ 10\\ 9\\ 12 \end{array} $			Dut	

Figure 9. The PSO algorithm is used to update the circuit optimization results after 48 iterations, and the left side of the circuit is the update iteration results of the circuit matrix

图 9. 使用 PSO 算法,经过 48 次迭代更新电路优化结果,电路左侧为电路矩阵的更新迭代结果

最后,我们统计分析了实验中使用的三种算法的适应度值或能量值随种群迭代次数变化的整体过程, 如图 10 所示。



Figure 10. Schematic diagram of IPSO, PSO, and AEIPSO iterative process. Where the abscissa is the number of iterations of the algorithm, and the left and right axes of the ordinate are the energy value and fitness value of the particles corresponding to the number of iterations

图 10. IPSO、PSO 和 AEIPSO 迭代过程示意图。其中横坐标为算法的迭代次数,纵坐标的左右轴分别为迭代次数对 应的粒子的能量值和适应度值

5.2.3. 全加器电路

1. AEIPSO 算法:

在 AEIPSO 算法的实验中, 粒子 P2 经过 33 次迭代后达到最小能量。其相应的电路图如图 11 所示。 同时, P1~P6 粒子的能量值如表 16 所示。

Table	16.	Energy of particles after iteration 33	
表 16.	第	33 次迭代后粒子的能量	

	P1	P2	Р3	P4	P5	P6
Energy	0	-16	3	-12	-10	-4



Figure 11. Using AEIPSO algorithm, the circuit optimization result is updated after 33 iterations, and the left side of the circuit is the update iteration result of the circuit matrix

图 11. 使用 AEIPSO 算法, 在 33 次迭代后更新电路优化结果, 电路左侧为电路矩阵的更新迭代结果

2. IPSO 算法:

在 IPSO 算法的实验中, 粒子 P1 经过 42 次迭代后达到最小能量。其相应的电路图如图 12 所示。同时, P1~P6 粒子的能量值如表 17 所示。

	P1	P2	P3	P4	P5	P6
Energy	-16	-3	3	-13	0	-4
	$ \begin{pmatrix} 1 & 1 \\ 3 & 2 \\ 3 & 3 \\ 7 & 5 \\ 8 & 1 \\ 5 & 3 \\ 12 & 5 \\ 11 & 2 \end{pmatrix} $	3 1 1 8 5 9 10 7)— Carry — Sum	

Table 17. Energy of particles after iteration 42 表 17. 第 42 迭代后粒子的能量

Figure 12. Using IPSO algorithm, the circuit optimization results are updated after 42 iterations, and the left side of the circuit shows the updated iteration results of the circuit matrix

图 12. 使用 IPSO 算法,在 42 次迭代后更新电路优化结果,电路左侧为电路矩阵的更新迭代结果

3. PSO 算法:

在 PSO 算法的实验中, 粒子 P4 经过 57 次迭代后达到最小能量。其相应的电路图如图 13 所示。同时, P1~P6 粒子的能量值如表 18 所示。

 Table 18. Fitness of particles after iteration 57

 表 18. 第 57 次迭代后粒子的适应度

	P1	P2	P3	P4	P5	P6
Energy	0	12	3	16	8	4



Figure 13. The circuit optimization results are updated after 57 iterations using the PSO algorithm, and the left side of the circuit shows the update iteration results of the circuit matrix

图 13. 使用 PSO 算法, 电路优化结果经过 57 次迭代后更新, 电路左侧为电路矩阵的更新迭代结果

我们统计分析了实验中使用的三种算法的适应度值或能量值随种群迭代次数变化的整体过程,如图14 所示。



Figure 14. Schematic diagram of the iterative process of the three algorithms IPSO, PSO, and AEIPSO. Where the abscissa is the number of iterations of the algorithm, and the left and right axes of the ordinate are the energy value and fitness value of the particles corresponding to the number of iterations

图 14. IPSO、PSO 和 AEIPSO 三种算法的迭代过程示意图;其中横坐标为算法的迭代次数,纵坐标的左右轴分别为 迭代次数对应的粒子的能量值和适应度值

6. 结论与讨论

本文分别使用 PSO、IPSO 和 AEIPSO 算法在前述实验中成功生成了满足要求的电路方案。然而,与 IPSO 和 AEIPSO 算法相比,PSO 算法容易陷入局部最优解场景,需要更多的迭代次数才能跳出局部最优 解场景,导致最终解的多样性较差。

在 Ising 模型和 Metropolis 采样算法的作用下,AEIPSO 算法需要更少的迭代就能使粒子跳出进化迭 代过程中的局部最优解,从而增加了最终解的多样性。此外,AEIPSO 算法还在粒子速度更新规则中增加 了除粒子先前位置以外的两种加权因子,第一种加权因子是粒子历史最优位置和粒子当前位置之间的差异, 另一种加权因子是种群粒子能量分布的前三层次中的粒子位置和粒子当前位置之间的差异。这种改进增加 了粒子迭代进化的引导因素种类,粒子的进化方向不再只由种群中唯一的最优粒子引导,而是能够在迭代 的不同时期进行合理的解空间探索,并且考虑多个层次最优粒子的影响。

迄今为止,智能优化算法已经产生了许多应用于不同场景的变体算法,继续寻找性能更好的优化算法 并将其应用于数字组合逻辑电路的设计与优化方面,将是我们未来要继续开展的工作。

参考文献

- [1] Tocci, R., Widmer, N. and Moss, G. (2006) Digital Systems: Principles and Applications. 10th Edition, Prentice-Hall, Inc.
- [2] Karnaugh, M. (1953) The Map Method for Synthesis of Combinational Logic Circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72, 593-599. https://doi.org/10.1109/tce.1953.6371932
- [3] Karakatic, S., Podgorelec, V. and Hericko, M. (2013) Optimization of Combinational Logic Circuits with Genetic

Programming. Electronics and Electrical Engineering, 19, 86-89. https://doi.org/10.5755/j01.eee.19.7.5169

- [4] Miller, J.F., Job, D. and Vassilev, V.K. (2000) Principles in the Evolutionary Design of Digital Circuits-Part I. Genetic Programming and Evolvable Machines, 1, 7-35. <u>https://doi.org/10.1023/a:1010016313373</u>
- [5] Moore, P.W. and Venayagamoorthy, G.K. (2006) Evolving Digital Circuits Using Hybrid Particle Swarm Optimization and Differential Evolution. *International Journal of Neural Systems*, 16, 163-177. https://doi.org/10.1142/s0129065706000585
- [6] Soliman, A.T. and Abbas, H.M. (2003) Combinational Circuit Design Using Evolutionary Algorithms. CCECE 2003, Vol. 1, 251-254.
- [7] Abd-El-Barr, M., Sait, S.M., Sarif, B.A.B. and Al-Saiari, U. (2003) A Modified Ant Colony Algorithm for Evolutionary Design of Digital Circuits. *The* 2003 *Congress on Evolutionary Computation*, Vol. 1, 708-715.
- [8] Pavitra, Y., Arun, E., Jamuna, S. and Manikandan, J. (2018) Study and Evaluation of Digital Circuit Design Using Evolutionary Algorithm. 2018 15th IEEE India Council International Conference (INDICON), Coimbatore, 16-18 December 2018, 1-5. <u>https://doi.org/10.1109/indicon45594.2018.8987190</u>
- [9] Eberhart, R. and Kennedy, J. (1995) A New Optimizer Using Particle Swarm Theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nayoga, 4-6 October 1995, 39-43. https://doi.org/10.1109/mhs.1995.494215
- [10] Coello Coello, C.A., Luna, E.H. and Aguirre, A.H. (2003) Use of Particle Swarm Optimization to Design Combinational Logic Circuits. In: Tyrrell, A.M., Haddow, P.C. and Torresen, J., Eds., *Lecture Notes in Computer Science*, Springer, 398-409. <u>https://doi.org/10.1007/3-540-36553-2_36</u>
- [11] Sagar, K. and Vathsal, S. (2013) Design of Combinational Circuits Using Evolutionary Techniques. International Journal of Science and Modem Engineering, 1, 47-51.
- [12] He, R., Wang, J.Y., Wang, Q., Zhou, H.J. and Hu, Y.C. (2005) An Improved Particle Swarm Optimization Based on Self-Adaptive Escape Velocity. *Journal of Software*, 16, 2036-2044. <u>https://doi.org/10.1360/jos162036</u>
- [13] Sabat, S.L. and Ali, L. (2008) Accelerated Exploration Particle Swarm Optimizer-AEPSO. 2008 IEEE Region 10 Conference, Hyderabad, 19-21 November 2008, 1-6. <u>https://doi.org/10.1109/tencon.2008.4766568</u>
- [14] Meng, X., Lin, Y. and Qui, D. (2017) Hybrid Algorithm of Adaptive Inertia Weight Particle Swarm and Simulated Annealing. *International Journal of Computer Techniques*, **4**, 105-110.
- [15] Atyabi, A., Phon-Amnuaisuk, S. and Ho, C.K. (2008) Cooperative Learning of Homogeneous and Heterogeneous Particles in Area Extension PSO. 2008 *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Hong Kong, 1-6 June 2008, 3889-3896. <u>https://doi.org/10.1109/cec.2008.4631326</u>
- [16] Atyabi, A., Phon-Amnuaisuk, S. and Ho, C.K. (2009) Applying Area Extension PSO in Robotic Swarm. Journal of Intelligent and Robotic Systems, 58, 253-285. <u>https://doi.org/10.1007/s10846-009-9374-2</u>
- [17] Yang, Q., Gao, H., Zhang, W. and Li, H. (2016) Simultaneous Hybrid Modeling of a Nosiheptide Fermentation Process Using Particle Swarm Optimization. *Chinese Journal of Chemical Engineering*, 24, 1631-1639. https://doi.org/10.1016/j.cjche.2016.08.013
- [18] Li, Y., Zhao, P., Guo, B., Zhao, C., Liu, X., He, S., et al. (2021) Design of Combinational Digital Circuits Optimized with Ising Model and PSO Algorithm. 2021 IEEE 15th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, 29-31 October 2021, 31-35. <u>https://doi.org/10.1109/asid52932.2021.9651723</u>
- [19] Shi, Y. and Eberhart, R.C. (1998) Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., Saravanan, N., Waagen, D. and Eiben, A.E., Eds., *Evolutionary Programming VII*, Springer, 591-600. https://doi.org/10.1007/bfb0040810

附录

附录 A:



Figure (a). Iterative change curve of F2 optimization function value 图(a). F2 优化函数值迭代变化曲线



Figure (b). Iteration change curve of F5 optimization function value **图(b).** F5 优化函数值迭代变化曲线



Figure (c). Graph of iterative change of F6 optimization function value 图(c). F6 优化函数值迭代变化曲线图



Figure (d). Iterative change curve of F9 optimization function value 图(d). F9 优化函数值迭代变化曲线



Figure (e). Iterative change curve of F10 optimization function value **图(e).** F10 优化函数值迭代变化曲线



Figure (a). Data selector circuit average energy (fitness value) iteration change curve **图(a).** 数据选择器电路平均能量(适应度值)迭代变化曲线





Figure (b). Iterative change curve of average energy (fitness value) of four-input-one-output circuit **图(b).** 四输入一输出电路平均能量(适应度值)迭代变化曲线



Figure (c). Iterative variation curve of the average energy (fitness value) of a one-bit full adder circuit 图(c). 一比特位全加器电路的平均能量(适应度值)迭代变化曲线