

考虑阀门切换均衡的连续微流控生物芯片 控制模式分配算法

刘耿耿^{1),2),3)} 马祖钊^{1),2),3)} 蔡华洋^{1),2),3)}
周茹平^{1),2),3)} 潘友林^{1),2),3)} 傅仰耿¹⁾

¹⁾(福州大学计算机与大数据学院 福州 350116)

²⁾(大数据智能教育部工程研究中心 福州 350116)

³⁾(福建省网络计算与智能信息处理重点实验室 福州 350116)

摘 要 连续微流控生物芯片是一种利用微米级通道精确操控流体的技术平台,相比于传统检测方法,具有消耗试剂少、分析速度快、自动化程度高等优势,在生物医学和疾病诊断等领域具有广泛应用。为了确保生化测定反应各步骤的顺序性和可控性,连续微流控生物芯片内部通常采用基于多路复用器的控制逻辑对流阀进行精确控制。控制逻辑通过控制通道将核心输入信号传递至控制阀门,从而实现流体路径的动态切换与精准操控。由于多路复用器中控制阀门的加工和集成成本较高,因此减少其数量可以降低芯片的制造成本以及设计复杂性,从而提高量产的可能性。此外,每一个控制阀门都是一个潜在的故障点,频繁的切换控制阀门会导致多路复用器的稳定性下降,导致芯片的可靠性降低。针对上述问题,本文提出了一种考虑阀门切换均衡的连续微流控生物芯片控制模式分配算法,以最小化控制阀门数量及控制阀门的最大切换次数。具体来说,该方法基于离散粒子群优化框架,并引入分角色学习策略与可行性约束下的交叉与变异学习机制,构建了一个多子种群的协同进化框架,有效引导粒子搜索在切换均衡性与控制资源消耗之间实现最优均衡的控制模式分配方案,提升了控制模式分配方案的质量与算法收敛性能,从而在降低芯片制造成本的同时提高生物芯片的可靠性。实验结果表明,与现有方法相比,所提出的算法在多路复用器中控制阀门的使用数量上平均优化了 16.40%,在控制阀门的最大切换次数上平均优化了 12.15%,且在实际的生化应用中有良好的表现。

关键词 连续微流控生物芯片;控制模式分配;阀门切换均衡;离散粒子群优化;分角色学习策略

中图法分类号 TP391

DOI号 10.11897/SP.J.1016.2026.01500

Valve-Switching-Balanced Control Pattern Assignment Algorithm for Continuous-Flow Microfluidic Biochips

LIU Geng-Geng^{1),2),3)} MA Zu-Zhao^{1),2),3)} CAI Hua-Yang^{1),2),3)}
ZHOU Ru-Ping^{1),2),3)} PAN You-Lin^{1),2),3)} FU Yang-Geng¹⁾

¹⁾(College of Computer and Data Science, Fuzhou University, Fuzhou 350116)

²⁾(Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou 350116)

³⁾(Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350116)

Abstract Continuous-flow microfluidic biochips (CFMBs) have emerged as a promising technological platform that enables complex biochemical experiments by precisely manipulating fluids within micrometer-scale channels. Compared with traditional laboratory methods, CFMBs offer

收稿日期:2025-12-18;在线发布日期:2026-03-31。本课题得到国家自然科学基金面上项目(62372109)、福建省杰出青年科学基金项目(2023J06017)资助。刘耿耿,博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为集成电路和微流控生物芯片的设计自动化。E-mail:liugenggeng@fzu.edu.cn。马祖钊,硕士研究生,主要研究领域为微流控生物芯片设计自动化。蔡华洋,博士研究生,主要研究领域为微流控生物芯片设计自动化。周茹平,博士研究生,主要研究领域为集成电路设计自动化。潘友林,博士研究生,主要研究领域为微流控生物芯片设计自动化。傅仰耿(通信作者),博士,教授,中国计算机学会(CCF)杰出会员,主要研究领域为数据挖掘与机器学习、智能决策支持系统。E-mail:fu@fzu.edu.cn。

several significant advantages, including low reagent consumption, rapid analysis speed, high integration density, and strong automation capability. These advantages enable CFMBs to integrate various laboratory operations, such as mixing, transportation, reaction, and detection on a single chip, thereby providing a compact and highly efficient experimental platform, which makes them widely used in biomedical analysis, disease diagnosis, and drug discovery. With the increasing complexity of biochemical assays implemented on CFMBs, the design of efficient and reliable control logic has become an important issue in ensuring the correct execution of experimental procedures. To ensure the sequentiality and controllability of each step in biochemical assay reactions, CFMBs typically employ multiplexer-based control logic to precisely actuate flow valves within the chip. In this architecture, core input signals are transmitted to control valves through control channels, enabling dynamic switching of fluid paths and precise manipulation of microscale fluids, thereby ensuring accurate execution of complex biochemical protocols on the chip. However, the fabrication and integration of control valves are costly, and the number of valves directly affects the design complexity and manufacturing cost of biochips. Therefore, the number of control valves in the multiplexer architecture should be reduced to lower manufacturing costs and simplify the control logic structure, thereby improving the feasibility of large-scale production. In addition to cost considerations, the reliability of control valves is another critical factor that influences the performance of CFMBs. In a multiplexer-based architecture, each control valve can be regarded as a potential failure point. During biochemical operations, certain valves may be switched frequently, which can lead to elastic degradation and structural instability, thereby reducing the overall reliability and lifetime of the biochip. However, existing control pattern assignment methods mainly focus on minimizing the number of control valves, while the switching frequency distribution among valves is often neglected. As a result, certain valves may be excessively switched, which increases the risk of valve failure and limits the robustness of the control system. To address the above challenges, this paper proposes a control pattern assignment algorithm for CFMBs that considers valve-switching balance. The proposed method simultaneously minimizes the number of control valves and the maximum switching times of control valves. Specifically, the algorithm is developed based on a discrete particle swarm optimization (DPSO) framework, in which a role-based learning strategy together with a feasibility-constrained crossover-mutation learning mechanism is introduced. These strategies form a multi-subpopulation cooperative evolutionary framework. Within this framework, particles with different roles exchange information during the optimization process, thereby improving the diversity of the population and enhancing the search capability in the discrete solution space. Experimental results show that the proposed algorithm achieves an average reduction of 16.40% in the number of control valves used in multiplexers, and an average reduction of 12.15% in the maximum switching times of control valves, while demonstrating promising performance in practical biochemical applications.

Keywords continuous-flow microfluidic biochips; control pattern assignment; valve-switching balance; discrete particle swarm optimization; role-based learning strategy

1 引言

在传统生化实验中,繁琐的人为操作不仅使流

程复杂,还难以保证实验准确性并导致试剂大量浪费^[1-2]。为提高效率和降低成本,连续微流控生物芯片(Continuous-Flow Microfluidic Biochips, CFMBs)应运而生,作为一项突破性技术,其利用微米级通道实

现流体精确操控,从而在提高 CFMBs 的自动化和微型化程度的同时^[3-4],取得了低试剂消耗和快速分析的优势^[5]。基于上述优点,CFMBs 已经在许多领域广泛应用,如生物医学和疾病诊断等领域^[6-8]。

CFMBs 由两层逻辑层组成,分别为控制层和流层,其中控制层包含用于压力传导的控制通道,流层包含用于运输试剂和流体样品的流通道^[9]。控制通道和流通道的交叉处存在一个阀门作为开关。图 1(a)呈现了阀门及其相关联通道的 3D 示意图。阀门是由聚二甲基硅氧烷(Polydimethylsiloxane, PDMS)材料制成的柔性膜^[10]。为了操控阀门的开闭,外部压力首先传导至控制通道,再由控制通道传递到阀门,从而实现控制阀门开启/关闭状态的切换。具体来说,当外部压力高时,阀门受压力驱使向下挤压以阻止流通道中的流体运动。而当外部压力低时,阀门将恢复原样,此时流体能够在流通道中自由流动。图 1(b)呈现了 CFMBs 的横截面示意图。

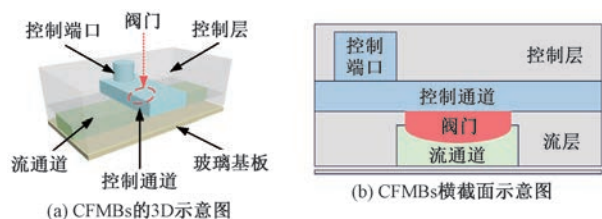


图 1 CFMBs 结构示意图

随着生物芯片制造技术的迅猛发展,成千上万个阀门已经能被集成到一个硬币大小的芯片中^[11]。然而,由于受到芯片面积和组件尺寸等物理条件的约束,为每个阀门配置单独的控制端口是不现实的。这不仅会显著增加芯片的制造成本和复杂性,还会加大对外部气动控制器的依赖^[12]。同时,随着微流控生物芯片集成度和实验复杂性提高,阀门数量不断增长,传统的独立端口控制方式已难以满足大规模阀门控制需求。因此,采用多路复用器(Multiplexer, MUX)技术来控制阀门已成为一种高效的解决方案^[13-14]。

基于多路复用器的控制逻辑架构最初由 Thorsen 等人提出^[15],该架构采用二进制编码方式,通过 $2N$ 个控制通道独立寻址并驱动 2^N 个流通道,其中流通道的状态由二进制阀门的组合阵列控制。基于该多路复用器的架构,Zhu 等人进一步提出了一种多通道控制方法来同时寻址多个控制通道,并且利用原始资源来构建备份通道,从而提高控制逻辑的容错能力^[16]。Liang 等人提出一种基于组合编码的 MUX 编码策略,利用 Sperner 定理实现理论上最大

编码容量,以使用更少的流通道和阀门来控制更多的独立控制通道,进一步提高了控制逻辑的资源利用率和可靠性^[17]。Wang 等人提出了一种基于汉明距离的微阀切换顺序优化方法,用于提高微流控生物芯片中控制层 MUX 的可靠性^[18]。Zeng 等人提出了一种基于混合离散粒子群优化的控制模式分配算法,进一步减少了 MUX 中的阀门数量^[19]。Cai 等人提出了一种基于深度强化学习的控制逻辑布线算法,有效减少了 MUX 中的控制路径长度,从而能够同时降低信号传输时延和控制设计成本^[20]。Huang 等人提出了一种基于深度强化学习的完全可编程阀门阵列生物芯片的控制逻辑综合方法,进一步优化了控制逻辑架构以提高芯片的效率和可靠性^[21]。此外,MUX 上的阀门和通道可能会遇到制造缺陷,包括阀门缺陷、堵塞和泄露^[22-24]。为此,Liu 等人提出了一种 MUX 的自动化故障检测方法,能够快速准确地检测 MUX 中出现的各类故障^[25]。

此外,在上述具有多通道切换机制的 MUX 中,合理分配控制模式不仅能够降低生物芯片的制造成本,还能提升控制逻辑的整体可靠性。例如,文献[16]采用带有交叉与变异操作的离散粒子群优化算法,优化控制模式分配方案,以减少所需的控制阀门数量。在此基础上,文献[19]通过改进粒子更新策略,实现了更高效的控制模式分配,进一步降低了控制阀门的使用数量。同时,文献[21]引入深度强化学习算法,探索更优的控制模式分配,也有效地减少了所需的控制阀门数量。

然而,到目前为止,现有的控制模式分配算法主要聚焦于 MUX 中控制阀门数量的最小化,并未对控制过程中阀门的切换均衡性给予足够关注。例如,某些控制阀门在执行生化测定功能的过程中可能被频繁切换,而其他控制阀门的切换次数则相对较少,导致控制阀门切换次数分布不均。需要注意的是,阀门的频繁切换易引发弹性退化以导致阀门失效^[18],进而影响生化测定的正常进行,甚至在严重情况下造成芯片结构性损坏。此外,阀门切换频次的失衡可能导致局部负载过载,继而降低控制逻辑的整体可靠性。为此,在确保生化测定功能正常执行的前提下,控制逻辑的设计不仅应优化控制模式以减少阀门数量,还需均衡阀门切换次数,以最大程度提高生物芯片的可靠性。

为了解决上述问题,本文基于粒子群优化(Particle Swarm Optimization, PSO)框架,并针对控制模式分配的离散特性,融合分角色学习策略与可行

性约束下的交叉与变异机制,提出了一种同时考虑减少控制阀门数量与控制阀门的最大切换次数的连续微流控生物芯片控制模式分配算法。本文的主要贡献如下:

(1)首次构建了同时考虑控制阀门数量与控制阀门的最大切换次数的控制模式分配优化模型,这对降低控制逻辑的设计复杂度和提高生物芯片可靠性具有重要意义。

(2)提出了一种引入分角色学习策略的离散粒子群优化算法,称为 DPSO-RL(Discrete Particle Swarm Optimization with Role-based Learning)。该算法以控制阀门数量与控制阀门的最大切换次数的加权和为适应度函数,旨在同时降低控制逻辑的复杂度和切换操作的不均衡性。在优化过程中,根据适应度值将粒子划分为多个子群体,并设置全局最优粒子(Gbest)、子群体最优粒子(Sbest)和普通粒子(Pbest)等不同角色,构建了一个多子群协同进化框架。通过不同角色间的分配方案信息传递与对比学习,构建出层次化的搜索机制,有效引导粒子搜索在切换均衡性与控制资源消耗之间实现最优权衡的控制模式分配方案。

(3)设计了一种基于交叉与变异的离散学习机制,实现了连续性粒子群优化算法在控制模式分配问题上的有效离散化。该机制在可行性约束条件下,引入针对控制模式结构的交叉与变异操作,使粒子在满足多路复用逻辑正确性的前提下实现对控制模式分配方案的有效扰动和重组。不仅提升了粒子群在离散控制模式分配空间中的全局搜索能力,同时有效改善了分配的精度,保证了所求控制模式分配方案的质量。

(4)使用多个实际测试用例来评估所提出的控制模式分配算法。实验结果表明,与现有方法相比,所提出的算法能够有效地最小化控制阀门数量和控制阀门的最大切换次数,从而获得资源利用率高、切换频次均衡的控制模式分配方案。

2 问题描述

2.1 阀门的逻辑简化

在控制模式分配问题中,假设某多路复用器含有 N 条控制通道,若已确定用于并行切换这些通道的组合,即多路复用组合,则需配置 $2 \times \log_2 N$ 个控制端口。并由此可构造出 $2^{\lceil \log_2 N \rceil}$ 种不同的控制模式。为更直观地描述控制关系,本文采用矩阵的形式

来表示上述多路复用组合。

如图 2 所示,本文以一个生化实验应用中复用矩阵的一部分为例。在该矩阵中,每行代表一个多路复用组合,每列代表一个控制通道,“0”和“1”分别表示阀门的关闭和开启状态。5 个流阀门(f_1-f_5)通过控制通道连接至核心输入,并采用 4 种多路复用组合实现多通道控制。此外,每个组合对应的控制模式在矩阵的旁侧列出,共有 A、B 和 C 三种控制模式分配方案,以直观呈现控制逻辑。例如,在分配方案 A 中,控制模式 $\bar{x}_1 \bar{x}_2 x_3$ 被分配给了多路复用组合 01100。当控制模式 $\bar{x}_1 \bar{x}_2 x_3$ 被激活时,与流阀门 f_2 和 f_3 相连接的控制通道将同步改变其控制状态,并在外部压力的作用下共同驱动流阀门 f_2 和 f_3 完成开启操作。

分配方案					A	B	C
0	1	1	0	0	$\bar{x}_1 \bar{x}_2 x_3$	$\bar{x}_1 \bar{x}_2 \bar{x}_3$	$\bar{x}_1 \bar{x}_2 \bar{x}_3$
1	0	0	1	0	$\bar{x}_1 x_2 \bar{x}_3$	$\bar{x}_1 x_2 x_3$	$x_1 \bar{x}_2 \bar{x}_3$
0	1	0	0	1	$x_1 \bar{x}_2 \bar{x}_3$	$\bar{x}_1 x_2 x_3$	$x_1 \bar{x}_2 x_3$
0	1	0	0	0	$x_1 x_2 x_3$	$\bar{x}_1 x_2 x_3$	$x_1 x_2 x_3$
f_1	f_2	f_3	f_4	f_5			

图 2 多路复用组合的不同分配方案

为了比较不同控制模式分配方案在控制逻辑上的资源使用差异,本文首先依据多路复用组合与分配方案之间的映射关系构建逻辑树表示,再结合逻辑化简及逻辑森林构造来分析各方案的资源消耗情况^[16]。例如,图 3 展示了分配方案 A 对应的此过程。在其逻辑结构中,根节点、叶子节点以及中间节点分别表示核心输入、流阀门和控制阀门,而节点间的连线表示用于传递控制信号的控制路径。最终分配方案 A 的逻辑森林构造的结果包含 18 个控制阀门和 25 条控制路径。相反,如图 4 所示,分配方案 B 和 C 的逻辑森林仅包含 11 个控制阀门和 17 条控制路径。

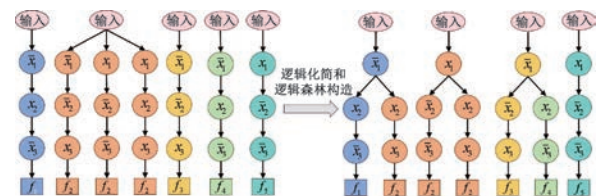


图 3 分配方案 A 的阀门逻辑化简和逻辑森林构造过程

图 5 和图 6 展示了分配方案 A 与方案 C 的逻辑森林布线结果,其中 6 个控制端口 $x_1, \bar{x}_1, x_2, \bar{x}_2, x_3$ 和 $\bar{x}_3$ 均与压力源相连,用于驱动 5 个流阀门的状态切换。由图 5 和图 6 可知,控制模式在各多路复用组合之间的分配差异会直接影响控制阀门的共

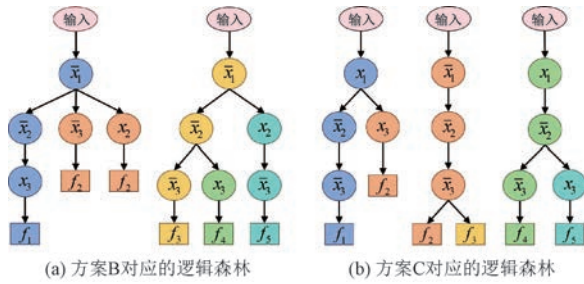


图4 分配方案B和C的阀门逻辑森林构造结果

享程度,从而导致所需阀门数量出现明显变化。因此,减少控制阀门数量不仅有助于缩短控制路径长度,同时也能够有效降低控制逻辑设计的复杂度,从而降低芯片的制造成本。

2.2 阀门切换次数优化

在控制逻辑的设计中,控制端口的压力值通常是互补的,同一时刻只能有一个端口具有高压^[26-27]。如图5所示, x_1 和 $\bar{x}_1$ 为一对互补的控制端口,分配方案A为多路复用组合01001分配的控制模式为 $x_1 \bar{x}_2 \bar{x}_3$,即状态100。而为组合01000分配的控制模式则为 $x_1 x_2 x_3$,即状态111。假设阀门状态需

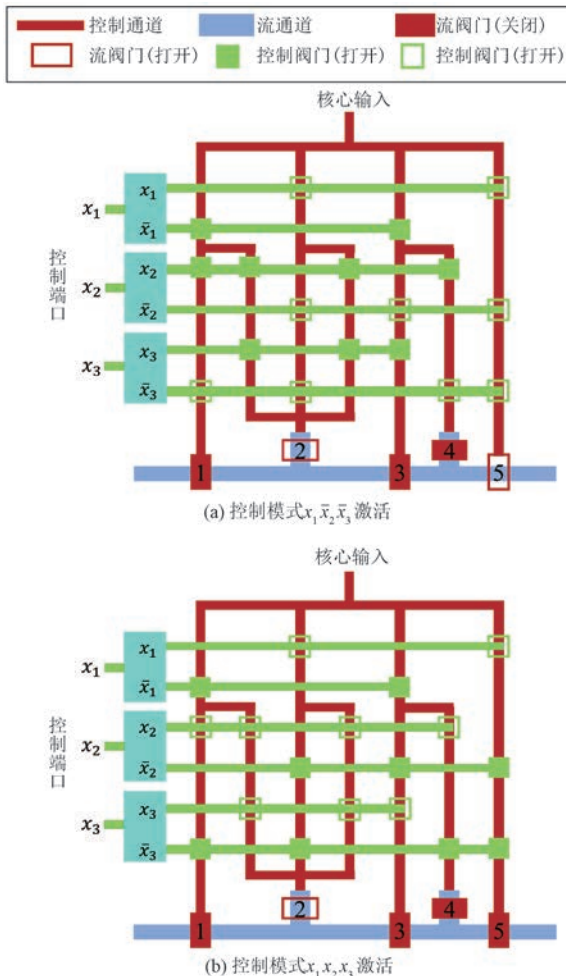


图5 分配方案A对应的逻辑森林布线图

要从如图5(a)所示时刻的“01001”切换至如图5(b)所示时刻的“01000”,多路复用器的控制阀门需完成从100到111的状态转换。此时 x_2 、 $\bar{x}_2$ 、 x_3 和 $\bar{x}_3$ 对应的控制阀门均需执行一次切换操作,因此多路复用器共需执行4次切换操作。基于上述切换情况,当采用如图6所示的分配方案C时,由于该方案为多路复用组合01001分配的控制模式为 $x_1 \bar{x}_2 x_3$,此时仅需切换 $\bar{x}_2$ 对应的控制阀门状态即可实现相同的功能。可以看出方案C仅使得多路复用器执行1次切换操作。相比于方案A,方案C在实现相同控制功能的前提下,减少了3次控制阀门切换操作,并显著降低了资源使用,提高了控制效率。

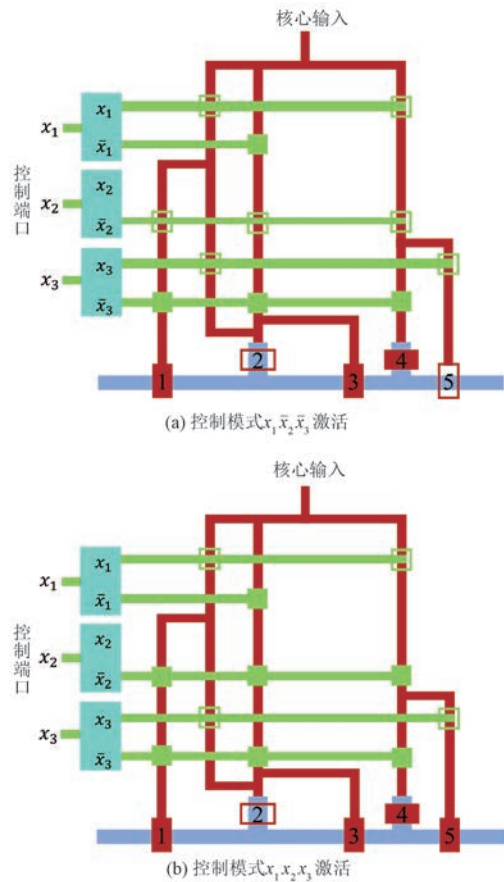


图6 分配方案C对应的逻辑森林布线图

此外,上述内容同样展示了在相邻的两个时刻之间如何计算控制阀门切换次数的方法。为了更完整地体现该过程,以不同时刻的流阀门切换动作矩阵 $\mathbf{SY}$ 为例。如公式(1)所示,矩阵中每一行表示某一时刻的流阀门状态,矩阵自上而下的排列顺序反映了流阀门状态随时间的变化过程,通过对矩阵中相邻行之间的状态差异进行分析,即可统计在各时刻之间发生的控制阀门切换次数,进而量化控制模式分配方案的切换均衡性。

$$SY = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

在此过程中,不同分配方案所对应的阀门切换次数如表 1 所示,其中 T-A 代表分配方案 A 的控制阀门切换次数,T-B 代表分配方案 B 的控制阀门切换次数,T-C 代表分配方案 C 的控制阀门切换次数。

表 1 SY 切换后不同分配方案对应的控制阀门的切换次数

控制阀门	x_1	$\bar{x}_1$	x_2	$\bar{x}_2$	x_3	$\bar{x}_3$
T-A	3	3	7	7	4	4
T-B	0	0	3	3	7	7
T-C	3	3	1	1	3	3

从 2.1 节所述的逻辑森林构建的结果可以看出,分配方案 B 和 C 均比方案 A 使用了更少的资源。此外,如表 1 所示,分配方案 A 和 B 控制阀门的最大切换次数均为 7 次,但最小控制阀门切换次数分别为 3 次和 0 次,各控制阀门之间的切换次数有较大的差异。这是因为现有的控制模式分配算法仅针对控制阀门数量进行优化,并未考虑控制阀门切换次数之间的均衡,导致部分控制阀门承担了过多的切换开销。而相比之下,分配方案 C 不仅使资源消耗更低,且最大的控制阀门切换次数仅为 3 次,最小控制阀门切换次数仅为 1 次。在减少控制阀门切换次数的同时,还均衡了各个阀门之间的切换次数。此外,由于每个控制阀门都是一个潜在的故障点,减少控制阀门数量同样意味着降低了潜在故障点的数量,从而在降低成本的同时提高了可靠性。同时,均衡阀门切换次数可避免某些阀门因过度使用而加速弹性退化,从而进一步延长阀门的工作耐久性并增强芯片的整体可靠性。

2.3 问题模型

基于上述分析,针对阀门切换均衡的 CFMBs 控制模式分配优化的问题模型如下:

输入: 不同时刻的流阀门切换动作矩阵,多路复用矩阵。

输出: 考虑阀门切换均衡的控制模式分配方案。

目标: 最小化控制阀门的数量,最小化控制阀门的最大切换次数。

3 算法设计

PSO 是一种基于种群的智能优化算法,粒子群中的每个粒子代表一个候选解^[28]。粒子通过结合自身经验(个体最优)与群体经验(全局最优),不断调整自身的位置与速度,从而逐步逼近最优解^[29]。原始的 PSO 算法主要面向连续优化问题,而本文研究的是考虑阀门切换均衡性的高质量控制模式分配问题,该问题属于离散优化范畴,因此需对 PSO 操作算子进行离散化处理。为解决上述问题,本文基于一种连续性粒子群优化算法^[30],在可行性约束下引入交叉与变异操作以实现 PSO 操作的离散化,提出了一种考虑阀门切换均衡的连续微流控生物芯片控制模式分配算法。该算法在减少控制阀门数量的基础上,进一步结合了最小化控制阀门的最大切换次数的优化目标,从而有效降低了逻辑架构的资源消耗并提升了芯片的可靠性。

3.1 算法流程

图 7 展示了算法的整体流程,首先,以不同时刻的

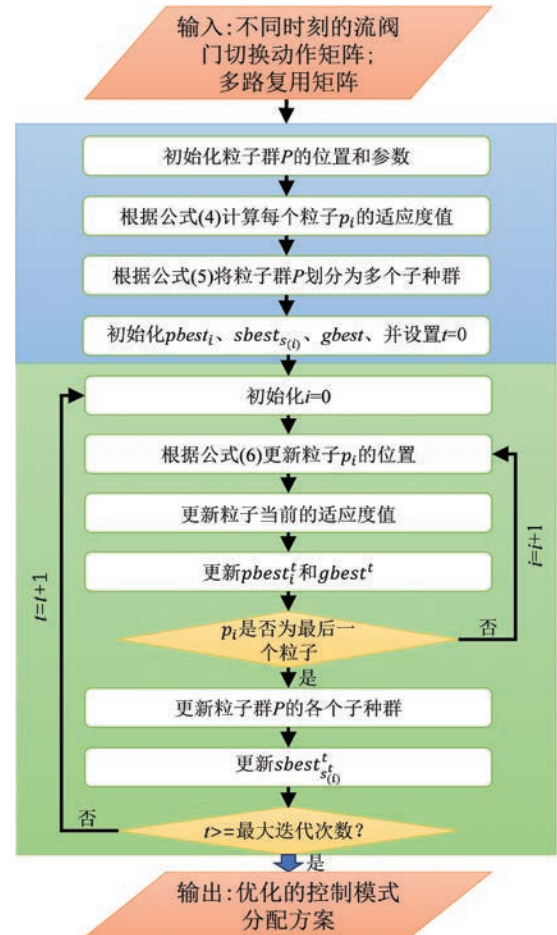


图 7 DPSO-RL 算法流程图

流阀门切换动作矩阵和多路复用矩阵作为输入,初始化粒子群的位置与速度,并设置算法参数。随后计算每个粒子的适应度值,根据适应度水平对粒子群进行动态分组,并相应调整各子群的规模。接着,为粒子分配不同的角色,包括全局最优粒子、子群最优粒子以及普通粒子,以此引导不同层次的学习策略:

(1)全局最优粒子整合各子群最优粒子的经验进行更新。

(2)子群最优粒子借鉴上层子群的最优粒子的经验进行更新。

(3)普通粒子则在子群内部学习当前子群最优粒子的经验进行更新。

在每一轮迭代中,粒子整体的学习网络如图 8 所示,根据上述学习策略更新粒子位置,不断探索最优的控制模式配置,直至满足最大迭代次数时输出优化的控制模式分配方案。

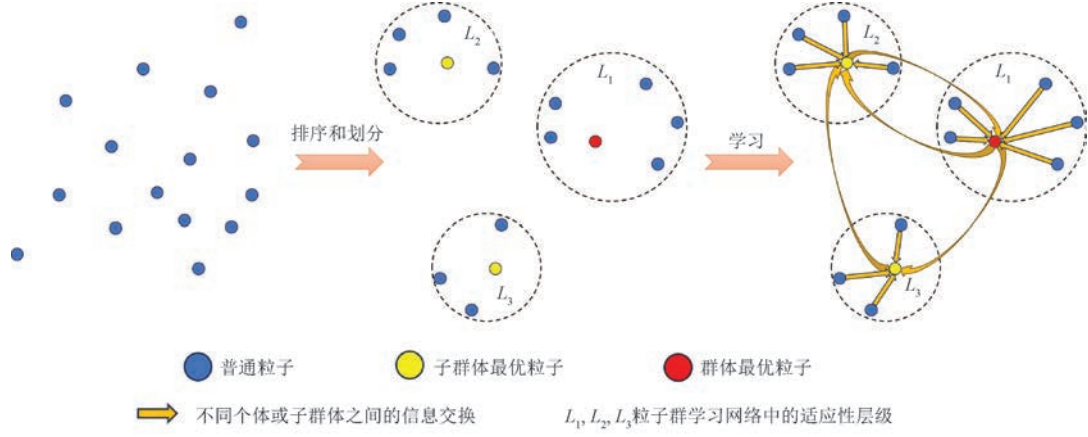


图 8 粒子间的学习网络示意图

3.2 本文算法设计

3.2.1 粒子编码方式

在本文中,每个粒子代表一种控制模式分配方案,粒子 p_i 的位置编码为

$$\text{cp}(0), \text{cp}(1), \dots, \text{cp}(i), \text{cp}(j), 0 \leq i, j \leq L-1 \quad (2)$$

$$\text{s.t. } \forall i, j \quad i \neq j, \text{cp}(i) \neq \text{cp}(j) \quad (3)$$

其中, $\text{cp}(i) = x (0 \leq x < 2^{\lceil \log_2 N \rceil})$, 表示分配给第 i 个多路复用组合的控制模式编号为 x , L 为多路复用组合的总数, N 为控制通道的数量。以图 2 中所示的矩阵为例,共有 5 条控制通道与核心输入端相连,因此需要配置 3 对互补的控制端口,共 6 个控制端口。基于这些端口,可以生成如下的控制模式(左侧数字代表编号)

$$\begin{aligned} 0: & \bar{x}_1 \bar{x}_2 \bar{x}_3 & 1: & \bar{x}_1 \bar{x}_2 x_3 & 2: & \bar{x}_1 x_2 \bar{x}_3 \\ 3: & \bar{x}_1 x_2 x_3 & 4: & x_1 \bar{x}_2 \bar{x}_3 & 5: & x_1 \bar{x}_2 x_3 \\ 6: & x_1 x_2 \bar{x}_3 & 7: & x_1 x_2 x_3 \end{aligned}$$

图 2 中的控制模式分配方案 C 对应的粒子编码表示为

$$p_i: \text{cp}(0)=0, \text{cp}(1)=4, \text{cp}(2)=5, \text{cp}(3)=7。$$

3.2.2 粒子适应度

为了评估不同控制模式分配方案的质量,粒子 p_i 的适应度函数如下:

$$\text{fitness}(p_i) = -(\alpha \cdot C_v + \beta \cdot T_m) \quad (4)$$

其中, α 和 β 是两个加权系数, C_v 为控制逻辑中的控制阀门数量, T_m 是控制阀门的最大切换次数。本文旨在在尽可能地减少控制阀门的数量与最大切换次数,因此适应度越大的粒子对应的控制模式分配方案质量越高。

3.2.3 粒子群的子群划分方式

在粒子群的更新过程中,群体根据粒子的适应度值被划分为 6 个子群,而非单一的整体种群,这种分群方式能够更好地保持群体多样性,同时各个子群的规模大小随着适应度水平动态的调整,能够有效增强对高质量控制模式分配方案的探索能力,避免陷入局部最优。子群划分的公式如下:

$$\begin{aligned} f_1 &= f_{\max} \\ f_2 &= f_{\min} + \alpha_1 (f_{\max} - f_{\min}) \\ f_3 &= f_{\min} + \alpha_2 (f_{\max} - f_{\min}) \\ f_4 &= f_{\min} + \alpha_3 (f_{\max} - f_{\min}) \\ f_5 &= f_{\min} + \alpha_4 (f_{\max} - f_{\min}) \\ f_6 &= f_{\min} + \alpha_5 (f_{\max} - f_{\min}) \\ f_7 &= f_{\min} \end{aligned}$$

$$\Omega_j^t = \{p_i^t \mid f_j \leq \text{fitness}(p_i^t) < f_{j+1}, i=1, 2, \dots, N\}, j=1, 2, \dots, 6 \quad (5)$$

其中, $f_{\max}$ 和 $f_{\min}$ 分别表示适应度函数的历史最大

值和最小值, $\{f_k, k=1, 2, \dots, 7\}$ 表示各个子群的边界。 Ω_j^t 表示第 t 次迭代中第 j 个子种群。 p_i^t 表示第 t 次迭代中第 i 个粒子的位置, $\text{fitness}(p_i^t)$ 表示粒子 p_i^t 的适应度值。 $\{\alpha_k, k=1, 2, \dots, 5\}$, 用于确定每个子种群适应度的上下界, 以避免各个子种群规模极端不平衡而影响算法的效果。

3.2.4 粒子状态更新

粒子状态的更新公式如下:

$$p_i^{t+1} = S_3(S_2(S_1(p_i^t, \omega), c_{11}, c_{12}, c_{13}), c_{21}, c_{22}, c_{23}) \quad (6)$$

其中, ω 是惯性权重, 用于调控粒子发生变异的概率, c_{1m} 和 c_{2n} 是加速度系数, 用于衡量粒子受自身经验和群体经验影响而进行交叉操作的概率。 S_1 代表惯性分量, S_2 代表个体经验感知, S_3 代表群体经验感知。

在 PSO 算法中, 惯性权重是一个非常关键的参数, 它对粒子的调整能力产生直接影响^[31]。为了协调算法在全局探索与局部开发之间的平衡, 本文采用一种自适应惯性权重, 公式如下:

$$\omega = 0.4 + 0.4 \times (\text{fitness}(p_i^t) - f_{\min}) / (f_{\max} - f_{\min}) \quad (7)$$

该式可将所有的惯性系数映射到区间 $[0.4, 0.8]$ 上, 这是 PSO 惯性权重的常见范围^[32]。对于适应度较高的粒子, 分配较小的惯性权重, 有助于其在当前优质解附近进行更精细的局部搜索, 而适应度较低或位于中间区间的粒子, 则被赋予较大的惯性权重, 从而鼓励其进行更广泛的全局探索。

(1) 惯性分量

算法中粒子的惯性分量更新通过变异操作实现, 更新公式表示如下:

$$W_i^{t+1} = S_1(p_i^t, \omega) = \begin{cases} \text{swap}(x_1, x_2), & r_1 < \omega \\ \text{rep}(x_1), & r_1 \geq \omega \wedge C_u \neq \emptyset \\ \text{mut}(x_1), & \text{else} \end{cases} \quad (8)$$

其中, W_i^{t+1} 是变异之后的粒子, r_1 是位于 $[0, 1)$ 区间中的随机数, C_u 是尚未被分配的控制模式编号集合。对于第 t 次迭代中的粒子 p_i^t , $\text{swap}(x_1, x_2)$ 通过将 x_1 和 x_2 对应的控制模式互换以更新粒子; $\text{rep}(x_1)$ 则是从 C_u 中随机选取一个控制模式来替换 x_1 所代表的控制模式; $\text{mut}(x_1)$ 则是从所有控制模式中(包括已经分配的控制模式)任选一个控制模式来替换 x_1 所代表的控制模式, 若替换之后出现控制模式重复, 即违反可行性约束, 则取消该操作。

图 9 展示了粒子 p_i^t 在迭代过程中通过变异操作进行位置更新的 3 种方式。图 9(a) 采用 $\text{swap}(0, 3)$ 操作, 将第 1 个和第 4 个已分配的控制模式进行交换。图 9(b) 采用 $\text{rep}(1)$ 操作, 在从未分配过的控制模式的集合 C_u 中取出编号为 6 的控制模式, 替换原本分配给第 2 个多路复用组合的控制模式。图 9(c) 采用 $\text{mut}(2)$ 操作, 试图用编号为 4 的控制模式替换第 3 个多路复用组合当前的控制模式。但由于替换后, 第 2 个和第 3 个多路复用组合分配的控制模式编号相同, 出现重复, 因此该替换操作被取消。

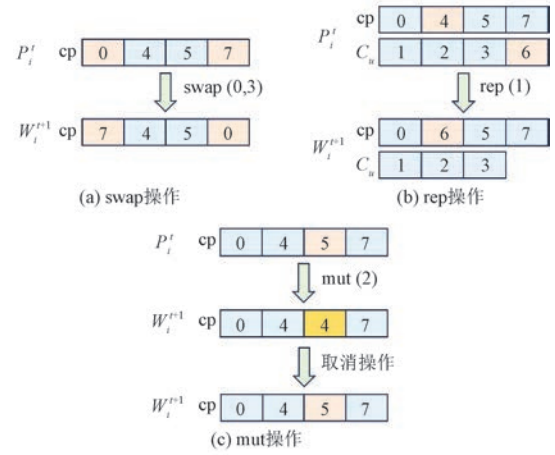


图 9 变异操作

(2) 个体经验感知

粒子的个体经验感知通过交叉操作进行更新, 公式表示如下:

$$I_i^{t+1} = S_2(W_i^{t+1}, c_{11}, c_{12}, c_{13}) = \begin{cases} X_r(W_i^{t+1}, p_{\text{best } i}^t), & r_2 < c_{1m} \\ W_i^{t+1}, & \text{else} \end{cases} \quad (9)$$

其中, I_i^{t+1} 是吸收个体经验之后的粒子, r_2 是位于 $[0, 1)$ 区间中的随机数, $p_{\text{best } i}^t$ 代表第 t 次迭代粒子个体的历史最佳位置, X_r 则表示两个粒子间的交叉操作, c_{1m} 为个体学习因子, 定义如下:

$$c_{1m} = \begin{cases} c_{11}, & p_i^t = g_{\text{best}}^t \\ c_{12}, & p_i^t = s_{\text{best } s(i)}^t \wedge p_i^t \neq g_{\text{best}}^t \\ c_{13}, & \text{else} \end{cases} \quad (10)$$

其中, c_{11} 为群体最优粒子的个体学习因子, c_{12} 为子群体最优粒子的个体学习因子, c_{13} 为普通粒子的个体学习因子, g_{best}^t 代表的是第 t 次迭代中种群的最优粒子, $s_{\text{best } j}^t$ 代表第 t 次迭代中第 j 个子种群的最优粒子, $s_{(i)}^t$ 代表第 t 次迭代时第 i 个粒子所在的子种群。

图 10 展示了粒子在进行个体经验感知时的交叉学习过程。此时,粒子当前的位置 W_i^{t+1} 与其个体历史最优位置 $p_{best i}^t$ 进行交叉操作,以生成新位置。对于编码位置上两者值相同的部分,直接保留该值至新粒子,对于值不相同的位置,从 W_i^{t+1} 和 $p_{best i}^t$ 的对应值中随机选择一个赋给新粒子。若交叉操作完成后,编码位置出现冲突,即新粒子中出现了控制模式编号重复的情况,违反了可行性约束,则取消本次交叉结果,重新执行交叉操作,直到满足无重复约束为止。通过上述交叉操作,有助于粒子在进化过程中保留部分优良信息。

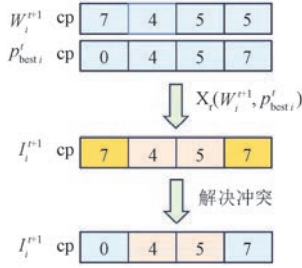


图 10 交叉操作

(3) 群体经验感知

粒子在进行群体经验感知时,全局最优粒子通过交叉学习的方式,从各个子群体的最优粒子中汲取经验,子群体最优粒子(非全局最优)向更优等级的相邻子群体最优粒子学习,以获得更高质量的搜索方向,普通粒子则向其所在子群体的最优粒子学习,不同角色的更新公式如下:

$$G_i^{t+1} = S_3(I_i^{t+1}, c_{21}, c_{22}, c_{23}) = \begin{cases} R_e(N(t) - 1), & r_3 < c_{21} \wedge p_i^t = gbest^t \\ X_r(I_i^{t+1}, sbest_{s(i)}^{t+1}), & r_3 < c_{22} \wedge p_i^t = sbest_{s(i)}^t \wedge p_i^t \neq gbest^t \\ X_r(I_i^{t+1}, sbest_{s(i)}^t), & r_3 < c_{23} \wedge p_i^t \neq sbest_{s(i)}^t \\ I_i^{t+1}, & \text{else} \end{cases} \quad (11)$$

其中, $N(t)$ 为子种群的数量, c_{21} 为群体最优粒子的群体学习因子, c_{22} 为子群体最优粒子的群体学习因子, c_{23} 为普通粒子的群体学习因子, G_i^{t+1} 代表交叉之后的粒子, r_3 是 $[0, 1)$ 中的随机数, R_e 表示迭代交叉操作,用于引导全局最优粒子与各个子群最优粒子之间进行交叉学习,具体来说,在每一轮迭代中,全局最优粒子会依次与每个子群的最优粒子进行 X_r 交叉操作,从而使粒子的更新过程不仅受到个体历史经验的影响,还能够吸收其他子群体的最佳信息。 $R_e(j)$ 的计算公式为

$$R_e(j) = \begin{cases} I_i^{t+1}, & j = 0 \\ X_r(R_e(j), s_{best j}^t), & j > 0 \wedge r_{rand}^j < c_j^t \\ R_e(j - 1), & \text{else} \end{cases} \quad (12)$$

其中, r_{rand}^j 为 $[0, 1)$ 中的随机数, c_j^t 表示第 j 个子群体中最优粒子(非全局最优)的适应度在所有非全局最优子群体的最优粒子适应度总和中所占的比例,根据定义 c_j^t 的计算公式为

$$c_j^t = \left(1 - \frac{\text{fitness}(s_{best j}^t)}{\sum_{k=1}^{N(t)-1} \text{fitness}(s_{best k}^t)} \right) / (N(t) - 2) \quad (13)$$

并且 c_j^t 满足

$$\sum_{j=1}^{N(t)-1} c_j^t = 1 \quad (14)$$

此外,群体经验感知阶段所采用的交叉操作与个体经验感知阶段相同。该操作使得粒子能够持续吸收群体中更优的遗传信息,在迭代过程中不断优化自身状态,从而提高全局搜索能力并收敛至全局最优解。

4 实验结果及分析

本文所提出的方法在一台配备 1.8GHz CPU 和 8GB 内存的 PC 上使用 C/C++ 实现,并采用 6 个测试用例验证所提出方法的有效性,其中包括在 RA30 芯片上执行的比色蛋白测定(CPA)、CPA 芯片上执行的体外诊断(IVD)测定、应用于 mRNA 芯片上的 IVD 检测,以及三种合成生物检测 R0、R1、R2^[21]。上述测试用例的详细信息在表 2 中详细列出,其中 N_{cm} 和 N_{cl} 分别为控制逻辑中使用的控制模式的数量和控制通道的数量, N_{tm} 为不同时刻阀门切换状态的数量。此外,所提出算法的使用参数在表 3 中列出,其中 M_s 为构造逻辑森林时可连续合并的逻辑树的最大数量。

表 2 测试用例详细信息

测试用例	RA30	CPA	mRNA	R0	R1	R2
N_{cm}	17	22	40	26	28	51
N_{cl}	19	25	37	22	27	48
N_{tm}	14336	1693	8030	6650	9295	12316

表 3 算法参数设置值

参数	值	参数	值
M_s	5	α_5	0.9
α	0.7	c_{11}	0.5
β	0.3	c_{12}	0.5
α_1	0.1	c_{13}	0.6
α_2	0.3	c_{21}	0.5
α_3	0.5	c_{22}	0.5
α_4	0.7	c_{23}	0.6

4.1 分角色学习策略有效性验证

为了验证分角色学习策略的有效性,本节将取消分角色学习策略后的 DPSO 算法与引入分角色学习策略的 DPSO-RL(Ours)算法进行对比。具体来说,取消分角色学习策略后的 DPSO 算法使得所有粒子统一向个体历史最优和群体历史最优进行学习,对比结果如表 4 和表 5 所示。从表 4 和表 5 中可以看出,DPSO-RL 算法在最佳控制阀门数量、最差控制阀门

数量、平均控制阀门数量、标准差上分别优化了 15.96%、20.08%、18.37%、20.54%。在最佳控制阀门的最大切换次数、最差控制阀门的最大切换次数、平均控制阀门的最大切换次数、标准差上分别优化了 5.97%、10.49%、9.28%、14.03%。上述结果表明,这一分角色、多方向的学习机制在有效引导粒子向优质解搜索的同时,保留了群体的多样性与探索能力,从而提升了算法在收敛精度与收敛稳定性上的表现。

表 4 DPSO 与 DPSO-RL 控制阀门数量的比较

用例	最佳 C_v			最差 C_v			平均 C_v			标准差		
	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)
RA30	90	59	34.44	127	73	42.52	113.53	66.73	41.22	8.43	4.14	50.89
mRNA	136	127	6.62	176	154	12.50	151.33	144	4.84	11.34	9.03	20.37
CPA	70	67	4.29	99	90	9.09	84.4	75.07	11.05	8.12	5.53	31.90
R0	122	101	17.21	143	131	8.39	132	116.13	12.02	5.76	8.52	-47.92
R1	113	88	22.12	153	114	25.49	126.33	104.45	17.32	10.49	7.84	25.26
R2	217	193	11.06	320	248	22.50	283.8	216.4	23.75	31.01	17.76	42.73
平均			15.96			20.08			18.37			20.54

表 5 DPSO 与 DPSO-RL 控制阀门的最大切换次数的比较

用例	最佳 T_m			最差 T_m			平均 T_m			标准差		
	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)	DPSO	Ours	优化率(%)
RA30	7290	7290	0.00	8778	7470	14.90	7980.53	7302	8.50	376.5	44.9	88.07
mRNA	3446	3332	3.31	3828	3500	8.57	3624.8	3425.2	5.51	140.51	48.15	65.73
CPA	750	720	4.00	772	764	1.04	760.13	739.33	2.74	7.10	14.02	-97.46
R0	3882	3862	0.52	4144	3890	6.13	3961.47	3872.4	2.25	71.61	6.94	90.31
R1	2954	2938	0.54	4396	3938	10.42	3953.07	3270.53	17.27	320.93	458.75	-42.94
R2	4786	3472	27.46	6158	4810	21.89	5390	4342.67	19.43	511.71	611.59	-19.52
平均			5.97			10.49			9.28			14.03

这主要是因为 DPSO 的粒子在进行群体经验感知时,总是朝着群体最优粒子进行学习,学习对象过于单一,导致群体粒子趋于同质化,进而容易在搜索控制模式分配方案的早期阶段陷入局部最优,且一旦陷入局部最优就难以再跳出。相比之下,采用分角色学习策略的 DPSO-RL 通过角色划分增强了粒子间的信息交互与多样性,使得群体最优粒子整合多个子群体最优粒子的经验进行更新,从而有效融合不同区域的搜索信息以提升寻求多样化控制模式分配方案的能力。另一方面,子群体最优粒子通过向上层子群体的最优粒子学习来借鉴更优子群的控制模式分配方案,从而引导本子群向更优解区域移动。此外,普通粒子则在子群内部向当前子群的最优粒子学习,有效增强了子群内对特定控制模式分配方案的协同优化能力。

4.2 本文算法有效性验证

为验证本文所提算法(DPSO-RL)在解决同时优化控制阀门数量和控制阀门的最大切换次数问题

上的有效性,本节将其与文献[16]和文献[21]中所提出的算法进行对比。由于目前尚不存在同时以最小化控制阀门数量和最小化最大阀门切换次数为目标的控制模式分配优化模型,为确保公平对比,本文对文献[16]和文献[21]中的算法进行了适配修改,将其优化目标调整为与本文一致的优化问题(即同时最小化控制阀门数量和控制阀门的最大切换次数)。

如表 6 和表 7 所示,相比于文献[16]的算法,本文算法在最佳控制阀门数量、最差控制阀门数量、平均控制阀门数量及标准差方面分别降低了 18.87%、19.00%、16.40%和 12.33%。在最佳控制阀门的最大切换次数、最差控制阀门的最大切换次数、平均控制阀门的最大切换次数及标准差方面,分别降低了 6.80%、12.05%、12.15%和 9.40%。此外,如表 8 所示,相比于文献[21]的算法,本文算法同样表现优异,在控制阀门数量上平均降低了 9.88%,在控制阀门的最大切换次数上平均降低了

11.60%。实验结果表明,在与上述两种方法的直接对比中,本文算法在最小化阀门数量和最小化最大

阀门切换次数这两个核心优化目标上均取得了显著的提升。

表 6 DPSO-RL 与文献[16]算法的控制阀门数量的比较

用例	最佳 C_v			最差 C_v			平均 C_v			标准差		
	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)
RA30	76	59	22.37	90	73	18.89	82.33	66.73	18.95	4.19	4.14	1.19
mRNA	137	127	7.30	179	154	13.97	156.93	144	8.24	11.71	9.03	22.89
CPA	79	67	15.19	95	90	5.26	85.40	75.07	12.10	4.24	5.53	-30.42
R0	126	101	19.84	167	131	21.56	136.53	116.13	14.94	9.82	8.52	13.24
R1	102	88	13.73	143	114	20.28	115.6	104.45	9.65	10.81	7.84	27.47
R2	296	193	34.80	376	248	34.04	330.4	216.4	34.50	29.40	17.76	39.59
平均			18.87			19.00			16.40			12.33

表 7 DPSO-RL 与文献[16]算法的控制阀门的最大切换次数的比较

用例	最佳 T_m			最差 T_m			平均 T_m			标准差		
	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)	文献[16]	Ours	优化率(%)
RA30	7664	7290	4.88	9060	7470	17.55	8659.73	7302	15.68	500.94	44.9	91.04
mRNA	3480	3332	4.25	4024	3500	13.02	3696.67	3425.2	7.34	150.87	48.15	68.09
CPA	738	720	2.44	766	764	0.26	755.87	739.33	2.19	7.25	14.02	-93.38
R0	3874	3862	0.31	4258	3890	8.64	4062.67	3872.4	4.68	133.37	6.94	94.80
R1	2954	2938	0.54	4404	3938	10.58	3923.2	3270.53	16.64	305.51	458.75	-50.16
R2	4846	3472	28.35	6188	4810	22.27	5898.93	4342.67	26.38	397.1	611.59	-54.01
平均			6.80			12.05			12.15			9.40

表 8 DPSO-RL 与文献[21]算法的比较

用例	文献[21]		Ours		优化率(%)	
	C_v	T_m	C_v	T_m	C_v	T_m
RA30	69	7290	59	7290	14.49	0.00
mRNA	130	3728	127	3332	2.31	10.62
CPA	74	762	67	720	9.46	5.51
R0	116	3870	101	3862	12.93	0.21
R1	92	3932	88	2938	4.35	25.28
R2	229	4822	193	3472	15.72	28.00
平均					9.88	11.60

特别是与文献[16]的算法相比,本文算法在控制阀门数量和最大控制切换次数的标准差指标上均有明显下降(分别降低 12.33%和 9.40%),这表明本文算法具有更高的求解稳定性,能够在多次运行中更可靠地获得控制阀门数量更少、控制阀门的最大切换次数更低且波动更小的高质量控制模式分配方案。

综上,在与文献[16]和文献[21]的全面对比中,本文提出的 DPSO-RL 算法在最小化控制阀门数量与最小化最大阀门切换次数这两个关键指标上均取得了更优的结果,充分验证了本文算法的有效性与优越性。

此外,本节对比了本文算法 DPSO-RL 与文献[16]算法在 6 个测试用例上的收敛速度和收敛精度。如图 11 所示,本文提出的 DPSO-RL 算法在所有基准测试中均取得了更高精度的控制模式分配

方案。此外,从收敛曲线的变化趋势来看,DPSO-RL 在早期阶段具有更快的收敛速度,并在后期保持了更高的优化潜力,最终收敛至质量更优的全局解。

上述实验结果表明,本文提出的分角色学习策略有效维持了粒子群在离散解空间中的多样性。这使得算法在搜索过程中能够持续探索不同的控制模式分配方案,有效避免了因粒子群过早同质化而陷入局部收敛,从而让算法在整个优化过程中更容易向全局最优方向收敛。相比之下,文献[16]中的算法缺乏有效的多样性维护机制,导致收敛曲线在中后期呈现明显的平台化趋势。这种过早收敛现象,使其难以跳出局部次优的控制模式分配方案,最终难以获得控制阀门数量更少且控制阀门的最大切换次数显著降低的高质量控制模式分配方案。

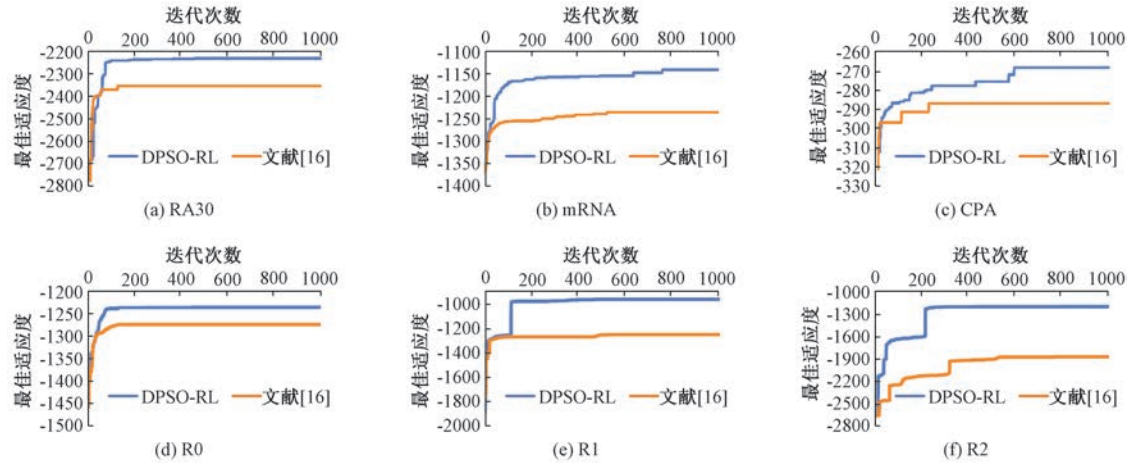


图 11 DPSO-RL 与文献[16]算法的收敛曲线对比

5 结 论

本文首次研究了 CFMBs 控制逻辑设计中同时考虑控制阀门数量与控制阀门的最大切换次数的控制模式分配问题,并提出了一种考虑阀门切换均衡的连续微流控生物芯片控制模式分配算法。该算法融合了分角色学习策略与可行性约束下的交叉与变异学习机制,将粒子划分为全局最优粒子、子种群最优粒子以及普通粒子三类角色,构建了一个多子种群协同进化框架。通过不同角色间的分配方案信息传递与对比学习,有效引导粒子搜索在切换均衡性与控制资源消耗之间实现最优权衡的控制模式分配方案。基于多个 CFMBs 测试用例的实验表明,本文提出的 DPSO-RL 算法能够在满足可行性约束条件的前提下,获得更高质量的控制模式分配方案,同时在算法稳定性和求解方案的精度方面均优于现有方法,验证了所提算法的有效性。

参 考 文 献

- [1] Mark D, Haeberle S, Roth G, et al. Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications. *Microfluidics Based Microsystems: Fundamentals and Applications*, 2010:305-376
- [2] Hu K, Chakrabarty K, Ho T Y. *Computer-Aided Design of Microfluidic Very Large Scale Integration Biochips*. Cham, Switzerland: Springer, 2017
- [3] Yao H, Wang Q, Ru Y, et al. Integrated flow-control code-sign methodology for flow-based microfluidic biochips. *IEEE Design & Test*, 2015,32(6):60-68
- [4] Tseng T M, Li B, Ho T Y, Schlichtmann U. Reliability-aware synthesis for flow-based microfluidic biochips by dynamic device mapping//*Proceedings of the ACM/IEEE Design Automation Conference*. New York, USA, 2015:1-6
- [5] Huang X, Ho T Y, Guo W, et al. Computer-aided design techniques for flow-based microfluidic lab-on-a-chip systems. *ACM Computing Surveys*, 2021,54(5):1-29
- [6] Zheng B, Roach L S, Ismagilov R F. Screening of protein crystallization conditions on a microfluidic chip using nanoliter-size droplets. *Journal of the American Chemical Society*, 2003,125(37):11170-11171
- [7] Heyman Y, Buxboim A, Wolf S G, Daube S S, Bar-Ziv R H. Cell-free protein synthesis and assembly on a biochip. *Nature Nanotechnology*, 2012,7(6):374-378
- [8] Chin C D, Laksanasopin T, Cheung Y K, Steinmiller D, Linder V, et al. Microfluidics-based diagnostics of infectious diseases in the developing world. *Nature Medicine*, 2011,17(8):1015-1019
- [9] Huang X, Pan Y, Zhang G L, Li B, Guo W, Ho T Y, Schlichtmann U. Pathdriver+: Enhanced path-driven architecture design for flow-based microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021,41(7):2185-2198
- [10] Unger M A, Chou H P, Thorsen T, et al. Monolithic micro-fabricated valves and pumps by multilayer soft lithography. *Science*, 2000,288(5463):113-116
- [11] Araci I E, Quake S R. Microfluidic very large scale integration with integrated micromechanical valves. *Lab on a Chip*, 2012,12(16):2803-2806
- [12] Liang S, Li M, Tseng T M, et al. Comux: Combinatorial-coding-based high-performance microfluidic control multiplexer design//*Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. New York, USA, 2022:1-9
- [13] Tseng T M, Li B, Li M, Ho T Y, Schlichtmann U. Relia-

- bility-aware synthesis with dynamic device mapping and fluid routing for flow-based microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016, 35(12):1981-1994
- [14] Shayan M, Bhattacharjee S, Song Y A, Chakrabarty K, Karri R. Toward secure microfluidic fully programmable valve array biochips. *IEEE Transactions on Very Large Scale Integration Systems*, 2019, 27(12):2755-2766
- [15] Thorsen T, Maerkl S J, Quake S R. Microfluidic large-scale integration. *Science*, 2002, 298(5593):580-584
- [16] Zhu Y, Huang X, Li B, et al. Multicontrol: Advanced control-logic synthesis for flow-based microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019, 39(10):2489-2502
- [17] Liang S, Li M, Tseng T M, et al. Combinatorial-coding-based high-performance microfluidic control multiplexer: design, synthesis, and adaptation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025, 44(2):655-668
- [18] Wang Q, Zuo S, Yao H, et al. Hamming-distance-based valve-switching optimization for control-layer multiplexing in flow-based microfluidic biochips//*Proceedings of the Asia and South Pacific Design Automation Conference*. Chiba, Japan, 2017:524-529
- [19] Zeng Yu-qin, Cai Hua-yang, Zhou Ru-ping, et al. Hybrid discrete particle swarm optimization algorithm for control pattern assignment. *Acta Electronica Sinica*, 2024, 52(8):2836-2849 (in Chinese)
(曾裕钦, 蔡华洋, 周茹平等. 基于混合离散粒子群优化的控制模式分配算法. *电子学报*, 2024, 52(8):2836-2849)
- [20] Cai Hua-yang, Huang Xing, Liu Geng-geng. Control logic routing for continuous-flow microfluidic biochips based on deep reinforcement learning. *Journal of Computer Research and Development*, 2025, 62(4):950-962 (in Chinese)
(蔡华洋, 黄兴, 刘耿耿. 基于深度强化学习的连续微流控生物芯片控制逻辑布线. *计算机研究与发展*, 2025, 62(4):950-962)
- [21] Huang X, Cai H, Guo W, et al. Control-logic synthesis of fully programmable valve array using reinforcement learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024, 43(1):277-290
- [22] Hu K, Yu F, Ho T Y, Chakrabarty K. Testing of flow-based microfluidic biochips: Fault modeling, test generation, and experimental demonstration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2014, 33(10):1463-1475
- [23] Hu K, Ho T Y, Chakrabarty K. Test generation and design-for-testability for flow-based mVLSI microfluidic biochips//*Proceedings of the IEEE VLSI Test Symposium*. Napa, USA, 2014:1-6
- [24] Zhu Y, Liu G, Guo W, et al. FTCD: Fault-tolerant co-design of flow and control layers for fully programmable valve array biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025, 44(7):2669-2682
- [25] Liu G, Zeng Y, Zhu Y, et al. Towards automated testing of multiplexers in fully programmable valve array biochips//*Proceedings of the Asia and South Pacific Design Automation Conference*. Incheon, Republic of Korea, 2024:570-575
- [26] Fidalgo L M, Maerkl S J. A software-programmable microfluidic device for automated biology. *Lab on a Chip*, 2011, 11(9):1612-1619
- [27] Melin J, Quake S R. Microfluidic large-scale integration: The evolution of design rules for biological automation. *Annual Review of Biophysics and Biomolecular Structure*, 2007, 36(1):213-231
- [28] Wang Feng, Zhang Heng, Han Meng-chen, Xing Li-ning, Chen Guo-long. A hybrid variable multi-objective particle swarm optimization algorithm based on co-evolution for solving the UAV collaborative multi-task allocation problem. *Chinese Journal of Computers*, 2021, 44(10):1967-1983 (in Chinese)
(王峰, 张衡, 韩孟臣等. 基于协同进化的混合变量多目标粒子群优化算法求解无人机协同多任务分配问题. *计算机学报*, 2021, 44(10):1967-1983)
- [29] Liu Geng-geng, Huang Yi-fei, Wang Xin, Guo Wen-zhong, Chen Guo-long. A hybrid discrete particle swarm optimization algorithm for X-architecture Steiner minimum tree under slew constraints. *Chinese Journal of Computers*, 2021, 44(12):2542-2559 (in Chinese)
(刘耿耿, 黄逸飞, 王鑫, 等. 基于混合离散粒子群优化的 Slew 约束下 X 结构 Steiner 最小树算法. *计算机学报*, 2021, 44(12):2542-2559)
- [30] J. E, Liu J, Wan Z. A novel adaptive algorithm of particle swarm optimization based on the human social learning intelligence. *Swarm and Evolutionary Computation*, 2023, 80:101336
- [31] Liu G, Zhu Y, Zhuang Z, Pei Z, Gan M, Huang X, Guo W. A robust multilayer X-architecture global routing system based on particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2024, 54(9):5627-5640
- [32] Lynn N, Suganthan P N. Ensemble particle swarm optimizer. *Applied Soft Computing*, 2017, 55:533-548



CAI Hua-Yang, Ph. D. candidate. His main research

LIU Geng-Geng, Ph. D., professor. His main research interests include design automation for integrated circuits and microfluidic biochips.

MA Zu-Zhao, M. S. candidate. His main research interests include design automation of microfluidic biochips.

interests include design automation of microfluidic biochips.

ZHOU Ru-Ping, Ph. D. candidate. Her main research interests include design automation of integrated circuits.

PAN You-Lin, Ph. D. candidate. His main research interests include design automation of microfluidic biochips.

FU Yang-Geng, Ph. D., professor. His main research interests include data mining and machine learning, intelligent decision support systems.

Background

The issue addressed in this paper concerns the control pattern allocation problem in the automated design of control logic for continuous-flow microfluidic biochips (CFMBs). Currently, numerous studies have been conducted on the control logic of CFMBs, focusing on various metrics, such as reducing manufacturing costs, improving control efficiency, and increasing chip fault tolerance. However, existing research on control pattern allocation in control logic mainly focuses on optimizing the number of control valves, while the balance of valve switching during operation has been insufficiently addressed. This neglect may lead to increased valve fatigue and reduced reliability. To address this problem, this paper introduces a pattern that simultaneously optimizes the number of control valves and the maximum switching frequency. Additionally, a discrete particle swarm optimization

with role-based learning (DPSO-RL) algorithm based on role-based learning and discrete crossover and mutation mechanisms is proposed, achieving a high-quality control pattern allocation that balances resource minimization and switching equilibrium. This approach has significant implications for reducing the manufacturing cost and enhancing the reliability of microfluidic biochips.

Compared with existing methods, the algorithm proposed in this paper ensures stability while obtaining a control pattern allocation scheme that requires fewer control valves and fewer maximum switching cycles.

In addition, this work was supported in part by the National Natural Science Foundation of China under Grant 62372109 and in part by the Fujian Natural Science Funds under Grant 2023J06017.