目录

[文献一 2](#_Toc145514094)

[文献二 3](#_Toc145514095)

[文献三 5](#_Toc145514096)

[文献四 7](#_Toc145514097)

[文献五 9](#_Toc145514098)

# 文献一

**文献名**：Wash Optimization for Cross-Contamination Removal in Flow-Based Microfluidic Biochips

**作者**：Kai Hu, Tsung-Yi Ho and Krishnendu Chakrabarty

**会议**：2014 19th Asia and South Pacific Design Automation Conference

**年份**：2014

**针对问题**：流式微流控芯片的清洗优化，并最小化清洗时间

**问题模型及约束**：

输入：芯片的布局；一组清洗目标；

目标：生成一种清洗液滴的布线方案并最小化清洗时间；

约束：

1、路径从缓冲流源端口开始，到废液槽结束；

2、每个顶点仅允许通过一次；

3、每个交叉点最多允许打开两个阀门；

4、清洗操作与其他操作尽量并行；

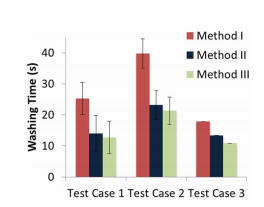
**解决方案**：

首先，需要构造一个路径字典，该字典存储基于芯片布局的物理合法路径。对于如何构造该路径字典，文献提出两种方法，第一种是基于DFS的算法，但该方法时间复杂度较高，无法应用于大规模的芯片；第二种方法是随机路径搜索，该方法主要思路是随机地从顶点列表中为当前顶点选择一个邻接顶点用于路径的扩展，它可以以很少的CPU时间覆盖全部顶点，然而可能忽略一些有效的候选路径。得到的路径字典很大，然而需要清洗的目标只是小部分，为了减少搜索字典的时间，按文献中提出的规则进行格式化。

清洗路径优化的问题可以表述为加权击球设置问题，用基于贪心算法的启发式算法来解决，在每次迭代过程中，总是选择包含目标顶点最多的路径进行清洗，然后从清洗目标集中去掉清洗过的顶点。由于流式微流控芯片的生成图的顶点度较少且不均匀，所以在进行优化之前需要给顶点赋予优先级，在字典中被更少路径包含的顶点优先级高，此外，文献还定义了一个函数用于赋优先级，该函数是非线性的以强调被覆盖少的顶点的高优先级。

**实验结果**：文献使用了两种芯片实例来评估该方案的优势。

第一个案例的结果如下：



其中Method 1，2，3分别表示上述的三种方法：构造路径字典、基于贪心策略的算法、清洗路径优化的算法，显然方法三能够最大限度地减少清洗时间。

第二个案例的结果也表明方法三具有最短的清洗时间。

# 文献二

**文献名**：Wash Optimization and Analysis for Cross-Contamination Removal Under

Physical Constraints in Flow-Based Microfluidic Biochips

**作者**：Kai Hu, Tsung-Yi Ho, and Krishnendu Chakrabarty

**期刊**：IEEE Transactions on Computer-Aided Design of Integrated

**年份**：2016

//引用：Hu K, Ho T Y, Chakrabarty K. Wash optimization for cross-contamination removal in flow-based microfluidic biochips[C]//2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2014: 244-249.

**针对问题**：流式微流控芯片的清洗优化，并最小化清洗时间。

**创新点：**考虑了污染物在缓冲流中的扩散系数对清洗时间的影响。

**痛点：**没有考虑多个清洗缓冲流入口的情况。

**问题模型及约束**：

输入：芯片的布局；一组清洗目标；污染物的种类；

目标：生成一种清洗液滴的布线方案并最小化清洗时间；

约束：

1、路径从缓冲流源端口开始，到废液槽结束；

2、每个顶点仅允许通过一次；

3、每个交叉点最多允许打开两个阀门；

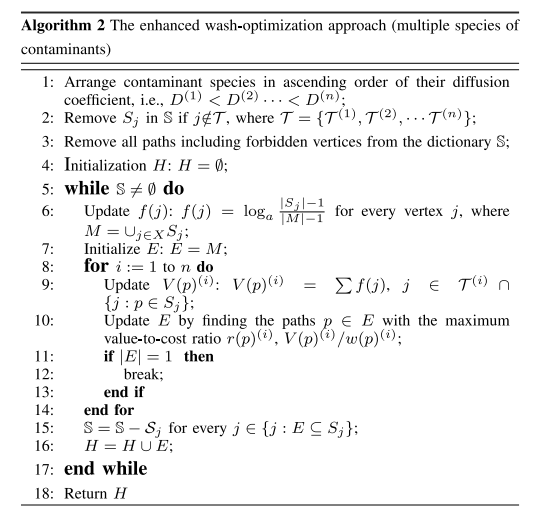
4、清洗操作与其他操作尽量并行；

**解决方案**：

首先，需要构造一个路径字典，该字典存储基于芯片布局的物理合法路径。对于如何构造该路径字典，文献提出两种方法，第一种是基于DFS的算法，但该方法时间复杂度较高，无法应用于大规模的芯片；第二种方法是随机路径搜索，该方法主要思路是随机地从顶点列表中为当前顶点选择一个邻接顶点用于路径的扩展，它可以以很少的CPU时间覆盖全部顶点，然而可能忽略一些有效的候选路径。得到的路径字典很大，然而需要清洗的目标只是小部分，为了减少搜索字典的时间，按文献中提出的规则进行格式化。

清洗路径优化的问题可以表述为加权击球设置问题，用基于贪心算法的启发式算法来解决，在每次迭代过程中，总是选择包含目标顶点最多的路径进行清洗，然后从清洗目标集中去掉清洗过的顶点。由于流式微流控芯片的生成图的顶点度较少且不均匀，所以在进行优化之前需要给顶点赋予优先级，在字典中被更少路径包含的顶点优先级高，此外，文献还定义了一个函数用于赋优先级，该函数是非线性的以强调被覆盖少的顶点的高优先级。//到这里都与2014那篇会议的内容一致，下面是该文相较于2014的工作的改进之处

该文定义了一种评价不同污染物的成本函数，然后根据下述算法进行迭代，最终返回优化后的路径集。



**实验结果**：文中对四种布线方案进行对比，分别是：缓冲液沿覆盖最多清洗目标的路径流动、基于路径字典的贪心算法、未考虑污染物种类的清洗优化算法（2014的工作）、考虑污染物种类的清洗优化算法。（下面分别称为方法一、二、三、四）

结果表明，清洗时间方面，方法一>方法二>方法三>方法四。CPU时间方面，方法二需要最少的CPU时间，方法四比方法二和方法三需要更多的计算，但它仍然只需要方法一的清洗执行时间的三分之一

# 文献三

**文献名**：Physical Synthesis of Flow-Based Microfluidic Biochips Considering Distributed Channel Storage

**作者**：Zhisheng Chen, Xing Huang, Wenzhong Guo, Bing Li, Tsung-Yi Ho

, and Ulf Schlichtmann

**会议**：Design, Automation and Test in Europe

**年份**：2019

**针对问题**：具有DCSA架构的FBMB的物理设计问题

**创新点：**考虑了运输冲突和清洗优化而提出的一种自顶向下的高级综合和物理设计方案

**痛点：**

**问题模型及约束**：

输入：建模为测序图的生化应用；元件库；扩散系数；

输出：一种优化的资源绑定与调度方案；优化的布局布线解决方案；

目标：最小化生化应用的完成时间；尽量减少微通道总长度；最大化资源利用率；

约束：

1、路径从缓冲流源端口开始，到废液槽结束；

2、每个交叉点最多允许打开两个阀门；

3、清洗操作与其他操作尽量并行；

**解决方案**：

为了设计一种优化的资源绑定和调度方案，以最小化生物测定的完成时间并最大化资源的利用率，该文通过扩展列表调度算法，提出了一种资源利用感知方法。首先，根据测序图中操作到水槽的最长路径长度赋予该操作优先级（测序图中顶点为操作，边为操作之间的关联），然后对于一个操作Oi，根据它的父操作提出了两种可能的情况来为该操作绑定组件，首先需要假设父操作中存在与当前操作相同类型的操作集，将该操作集称为Os，两种情况分别为：

·情况一：Os 中至少存在一个操作，其输出流体仍在其所绑定的组件内部

·情况二：Os中所有操作的输出流体已从它们所绑定的组件中移除，或者Os 是一个空集。

针对情况一，从其描述的操作集中用于执行操作的组件中选择输出流体具有最低扩散系数的组件，并将Oi绑定到该组件。

针对情况二，选择准备时间最早的组件（与 Oi 类型相同）与Oi绑定。

对于布局方案，该文提出了一种基于经典模拟退火的布局方法。针对布线，该文提出了一种运输冲突感知布线算法，以找到总路径长度最小化的可行路径，同时避免运输冲突。根据开始时间对所有运输任务进行非降序排序，并使用改进的 A\* 算法顺序为每个运输任务找到路径。

**实验结果**：该文将所提出的算法与基线算法（BA）进行比较，该算法在资源利用率和总信道长度方面分别比 BA 平均提高 12.5% 和 5.7%，CPU时间接近，生化应用的平均执行时间减少了6.4%。

**学习笔记：**

·传统的通过存储单元来存储中间结果的劣势在于存储单元的带宽太低了，而且占据很大的芯片面积。

·DSCA架构的FBMB的一大痛点是，暂存流体的微通道可能在接下来的时钟周期中还要用到，需要及时清洗，所以同步是大问题。

·清洗时间受到四个因素的影响：通道长度、通道宽度、施加给缓冲流的压力和污染物的扩散系数

·基于 DCSA 的生物芯片的设计自动化框架包含两个步骤：

1）资源绑定和调度；//绑定操作所需的资源和设计一个安排操作顺序的方案

2）布局和布线；

# 文献四

**文献名**：Computer-aided Design Techniques for Flow-based Microfluidic Lab-on-a-chip Systems

**作者**：Xing Huang, Tsung-Yi Ho, WenZhong Guo, Bing Li, Krishnendu Chakrabarty, and Ulf Schlichtmann

**期刊**：ACM Computing Surveys

**年份**：2021

**针对问题**：具有DCSA架构的FBMB的物理设计问题

**创新点：**考虑了运输冲突和清洗优化而提出的一种自顶向下的高级综合和物理设计方案

**痛点：**

**问题模型及约束**：

输入：建模为测序图的生化应用；元件库；扩散系数；

输出：一种优化的资源绑定与调度方案；优化的布局布线解决方案；

目标：最小化生化应用的完成时间；尽量减少微通道总长度；最大化资源利用率；

约束：

1、路径从缓冲流源端口开始，到废液槽结束；

2、每个交叉点最多允许打开两个阀门；

3、清洗操作与其他操作尽量并行；

**解决方案**：

为了设计一种优化的资源绑定和调度方案，以最小化生物测定的完成时间并最大化资源的利用率，该文通过扩展列表调度算法，提出了一种资源利用感知方法。首先，根据测序图中操作到水槽的最长路径长度赋予该操作优先级（测序图中顶点为操作，边为操作之间的关联），然后对于一个操作Oi，根据它的父操作提出了两种可能的情况来为该操作绑定组件，首先需要假设父操作中存在与当前操作相同类型的操作集，将该操作集称为Os，两种情况分别为：

·情况一：Os 中至少存在一个操作，其输出流体仍在其所绑定的组件内部

·情况二：Os中所有操作的输出流体已从它们所绑定的组件中移除，或者Os 是一个空集。

针对情况一，从其描述的操作集中用于执行操作的组件中选择输出流体具有最低扩散系数的组件，并将Oi绑定到该组件。

针对情况二，选择准备时间最早的组件（与 Oi 类型相同）与Oi绑定。

对于布局方案，该文提出了一种基于经典模拟退火的布局方法。针对布线，该文提出了一种运输冲突感知布线算法，以找到总路径长度最小化的可行路径，同时避免运输冲突。根据开始时间对所有运输任务进行非降序排序，并使用改进的 A\* 算法顺序为每个运输任务找到路径。

**实验结果**：该文将所提出的算法与基线算法（BA）进行比较，该算法在资源利用率和总信道长度方面分别比 BA 平均提高 12.5% 和 5.7%，CPU时间接近，生化应用的平均执行时间减少了6.4%。

**学习笔记：**

·传统的通过存储单元来存储中间结果的劣势在于存储单元的带宽太低了，而且占据很大的芯片面积。

·DSCA架构的FBMB的一大痛点是，暂存流体的微通道可能在接下来的时钟周期中还要用到，需要及时清洗，所以同步是大问题。

·清洗时间受到四个因素的影响：通道长度、通道宽度、施加给缓冲流的压力和污染物的扩散系数

·基于 DCSA 的生物芯片的设计自动化框架包含两个步骤：

1）资源绑定和调度；//绑定操作所需的资源和设计一个安排操作顺序的方案

2）布局和布线；

# 文献五

**文献名**：Flow-Based Microfluidic Biochips with Distributed Channel Storage: Synthesis,Physical Design, and Wash Optimization

**作者**：Xing Huang, Wenzhong Guo, Zhisheng Chen, Bing Li, Tsung-Yi Ho

**期刊**：IEEE Transactions on Computers

**年份**：2021

**针对问题**：具有DCSA架构的FBMB的物理设计问题

**创新点：**考虑了运输冲突和清洗优化而提出的一种自顶向下的高级综合和物理设计方案

**痛点：**

**问题模型及约束**：

输入：建模为测序图的生化应用；元件库；扩散系数；

输出：一种优化的资源绑定与调度方案；优化的布局布线解决方案；

目标：最小化生化应用的完成时间；尽量减少微通道总长度；最大化资源利用率；

约束：

1、路径从缓冲流源端口开始，到废液槽结束；

2、每个交叉点最多允许打开两个阀门；

3、清洗操作与其他操作尽量并行；

**解决方案**：

为了设计一种优化的资源绑定和调度方案，以最小化生物测定的完成时间并最大化资源的利用率，该文通过扩展列表调度算法，提出了一种资源利用感知方法。首先，根据测序图中操作到水槽的最长路径长度赋予该操作优先级（测序图中顶点为操作，边为操作之间的关联），然后对于一个操作Oi，根据它的父操作提出了两种可能的情况来为该操作绑定组件，首先需要假设父操作中存在与当前操作相同类型的操作集，将该操作集称为Os，两种情况分别为：

·情况一：Os 中至少存在一个操作，其输出流体仍在其所绑定的组件内部

·情况二：Os中所有操作的输出流体已从它们所绑定的组件中移除，或者Os 是一个空集。

针对情况一，从其描述的操作集中用于执行操作的组件中选择输出流体具有最低扩散系数的组件，并将Oi绑定到该组件。

针对情况二，选择准备时间最早的组件（与 Oi 类型相同）与Oi绑定。

对于布局方案，该文提出了一种基于经典模拟退火的布局方法。针对布线，该文提出了一种运输冲突感知布线算法，以找到总路径长度最小化的可行路径，同时避免运输冲突。根据开始时间对所有运输任务进行非降序排序，并使用改进的 A\* 算法顺序为每个运输任务找到路径。

相较于2019发表于DATE的工作，本文还提出了一种新的清洗优化方案，其思路主要是先建立清洗路径，准确计算每个清洗操作的开始时间和持续时间，以便清洗、运输与缓存可以适当协调而不会发生任何冲突。提出了调度计算流程，通过将生成的芯片架构反馈回高级综合阶段（上述绑定方案的算法）来重新计算完整生物测定的精确执行过程。文中把清洗任务分为两种，一种是从组件之间的传输或组件到微通道进行缓存而造成污染，另一种是取液操作导致的污染，即无组件参与。然后把污染路径映射到图中，通过文中所述的策略可以构建出清洗路径。计算出优化时间窗口，再加入调度方案中进行优化。

**实验结果**：

该文将所提出的系统级设计与SLBA进行比较，该算法在资源利用率和总信道长度方面分别比 SLBA 平均提高5.6% 和减少 12.8%，CPU时间接近，生化应用的平均执行时间减少了5.2%，清洗时间方面优化了34%，平均缓存时间减少了36%。

所提出的清洗优化方法与DDWO相比，平均清洗时间减少了15.9%。