本周工作：阅读并整理了两篇数字微流控生物芯片清洗的文献，有另外一篇快把解决方案部分看完。

目录

[文献一 2](#_Toc144805256)

[文献二 5](#_Toc144805257)

# 文献一

**文献名**：Integrated Functional and Washing Routing Optimization for Cross-Contamination Removal in Digital Microfluidic Biochips

**作者**：Hailong Yao, Qin Wang, Yiren Shen, Tsung-Yi Ho and Yici Cai

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

**年份**：2015

**针对问题**：考虑容量浪费的清洗液滴布线；避免功能液滴违反时间约束的同步问题

**创新点：**考虑了实际情况下液滴的清洗能力

**贡献点：** 提出了一个不会发生错误的清洗液滴和功能液滴布线方案

**痛点：**对于多个交叉污染点建模为有向有环图时的处理没有讨论；

**问题模型及约束**：

输入：网络列表；一组清洗液滴；一组路径堵塞；一组储液槽；

输出：一种考虑容量浪费的调度方案；

目标：在不违反约束的情况下计算所有网络的可行路径和调度解决方案；同时最小化执行时间的加权和、交叉污染点的数量以及用于布线的单元数量。

约束：

1、污染约束：防止功能液滴的交叉污染；

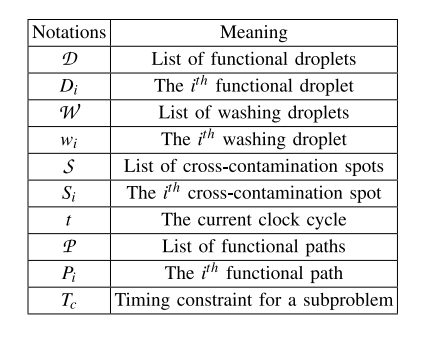
2、流体约束：每条路径之间的距离应大于规定的最小间距；

3、时间约束；

4、清洗液滴清洗能力约束：清洗液滴清洗非目标污染点和清洗目标污染点的清洗能力之和应小于最大清洗能力；

**解决方案**：

该方法的总体流程，包括五个主要步骤：1）功能液滴布线； 2）功能液滴路径压缩； 3）交叉污染点分析； 4) 清洗液滴布线； 5) 功能液滴和清洗液滴路径压缩。



1）功能液滴布线：基于经典的 A\* 搜索算法（即具有 A\* 成本函数的 Lee 式迷宫布线）允许多个液滴同时运动，从而能够获得全局最优解。为了得到满足流体约束的布线方案，将路径相邻的单元都视为该液滴的路径（即一个液滴的路径可能占用多个单元格。造成资源利用率不高？），这样即可避免违法流体约束，为了避免违反时间约束，修改后的A\*算法会删去违反时间约束的路径。为了尽量避免交叉污染，对使用过的单元设置较高的成本，但需绕行超过四个单元时则会选择已使用过的单元。布线时，需对路径进行排序，否则有发生死锁的可能（清洗操作的安排可能会抑制一些功能液滴停留在源点）。

为了避免功能液滴在源点或目标点的流体约束导致功能液滴不能正常流动，应按路径排序规则进行调度：如果满足以下任何一个条件，则液滴A需要早于液滴B被调度：1）A的源位置阻挡B的路由路径，2）B的目标位置阻挡A的路由路径。将路径建模成有向图，例如，当A顶点需先于B顶点被调度，则顶点A到B有一条边。这样可以构建出有环图，用以下策略消去环：

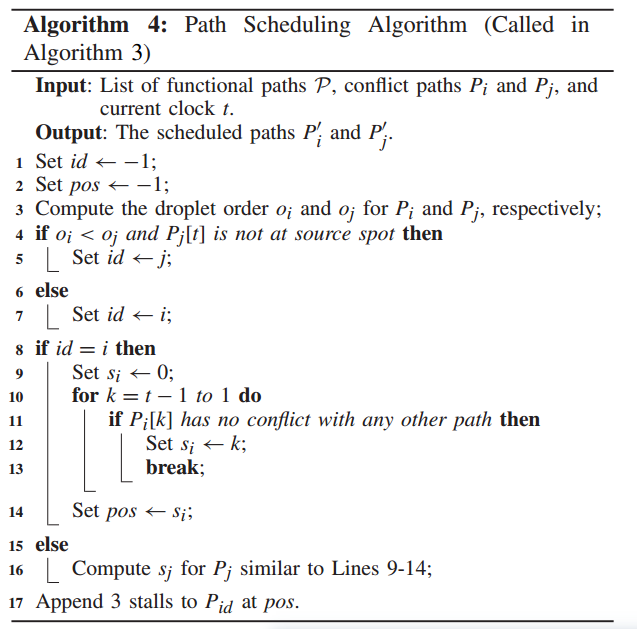
1、基于协商策略的撕裂和重布线；

2、布线让步方法；

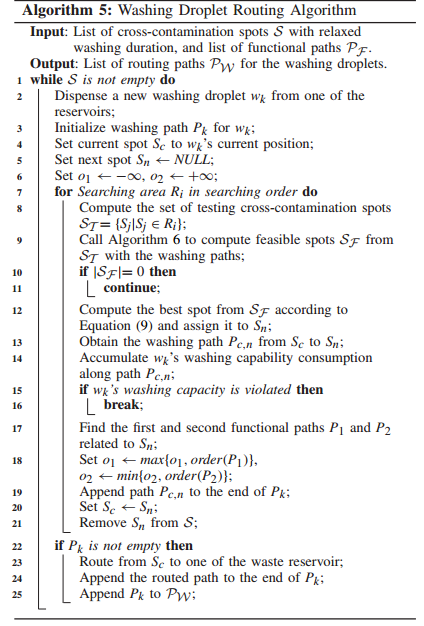
3、基于虚拟拓扑的布局细化，实现无死锁布线解决方案。

接下来，将对 DAG 进行拓扑排序以获得路径列表1的排序。剩余路径列表2根据其路径长度排序。路径长度越长，相应液滴的阶数越小。最后，通过归并排序将两个排序的功能路径列表根据其长度合并在一起。对于任何交叉污染点，我们只允许阶次较小的液滴较早通过该点，总之尽量使两个功能液滴到达交叉点的间隔大以便清洗操作。//不太理解这一步，

2）功能液滴路径压缩：按下图算法执行



3）清洗液滴布线：首先，文中提出一种清洗时间松弛的方法，也就是尽量增加两个功能液滴到达交叉点的间隔，提出一种约束来增加清洗时间。针对多个交叉污染点，首先要决定清洗的顺序，在清洗缓冲液源端口相邻的列中找到若干个候选清洗交叉污染点，然后文中设计了一个成本函数用于筛选出第一个清洗的交叉污染点，其函数的思想主要是选择距离最近且第一个功能液滴较早到的污染点，然后按需求分配权重给这两个因素。然后构建一个搜索区域，在该区域内按上述方法搜索下一个清洗的污染点，还要记录清洗液滴清洗非目标污染点的消耗，迭代进行直至芯片边界或清洗能力消耗完。按修改后的A\*算法计算清洗液滴路径，然后按功能液滴路径排序的规则和约束对清洗液滴路径进行排序。



4）同时进行功能液滴和清洗液滴路径压缩：这个步骤是为了避免清洗液滴和功能液滴的冲突。先将所有清洗路径加入到功能液滴路径中，然后用功能液滴路径压缩算法来对全部路径进行压缩，当清洗液滴和功能液滴发生冲突时，让顺序值较大的液滴等待。当一个液滴等待时，后续的所有液滴都要等待。

**实验结果**：

结果表明，本文的方法将所有错误率降低到0，所有清洗操作在容量限制内都是有效的。从结果来看，也有一些交叉污染点未能清洗干净。且该方法的平均运行时间比[1]快了28倍。

**学习笔记：**

1、文中提出了清洗的一个大问题：容量浪费和容量限制

参考文献：

[1]Huang T W, Lin C H, Ho T Y. A contamination aware droplet routing algorithm for the synthesis of digital microfluidic biochips[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2010, 29(11): 1682-1695.

# 文献二

**文献名**：Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips//Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips（2009）的期刊版本

**作者**：Yang Zhao and Krishnendu Chakrabarty

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

**年份**：2012

**针对问题**：不相交的功能液滴路径发生交叉污染的避免；最小化总路径时间；

**创新点：**提出了一种结合了不相交液滴布线和功能液滴与清洗液滴同步的方案

**贡献点：**提出一种避免不相交的功能液滴路径交叉污染的液滴布线方法；提出了一种最小化清洗操作数量的最佳插入清洗操作的方法；提出了一种结合这两种方案的方法

**痛点：**对于多个交叉污染点建模为有向有环图时的处理没有讨论；

**问题模型及约束**：

输入：生物测定操作的时间表；模块布局；

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、每条路径之间的距离应大于规定的最小间距；

2、流体约束；

3、功能液滴路径应规定上限，避免液滴内化学物质的损失；(RLCC)

4、

**解决方案**：

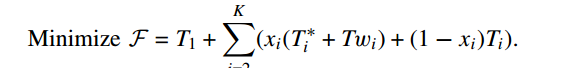
针对功能液滴路径不相交的问题，文中提出了一种寻找互不相交的功能液滴路径的方法并最小化总液滴路径。首先将路径规划问题分解为若干子问题，对于每个子问题分别求解。将芯片结构建模为图，顶点为电极单元，若两顶点对应的电极单元是相邻的，则这两顶点之间有边。对于三引脚网络（两源，一目标），将其转化为二引脚网络，计算三引脚网络中每个电极到目标的曼哈顿长度，根据最小曼哈顿长度找到一个混合点，从而将三引脚网络转化为三个二引脚网络（每个引脚到混合点）。给定一组 n 个 2 引脚网络，为这些网络找到可行的不相交路径的问题等价于寻找G中相互顶点不相交或边不相交的路径的问题。子问题Pi代表确定（ti,si）的二针网络与其他路径不相交的路径。该问题是NP完备问题，所以应用启发式方法解决。此外，难以找到顶点不相交的情况下，找边不相交的路径可以减少清洗的电极单元数量，因为只需清洗交叉点。该算法的输入是子问题中要布线的网络列表，输出是一组受约束且具有最小长度的顶点不相交（首选）或边不相交（折衷）路径。用修改后的Lee算法[1]对子问题Pi的各个网络顺序地进行布线，在此之前，需用[2提出的网络路由排序方法优化各个网络的顺序，具体地说，根据下述式子进行计算，数值越大优先级越低，其中pin（p）是网络p的边框内的引脚数，A是自定义的，X/Yrange（p）是边框横/纵向边长。



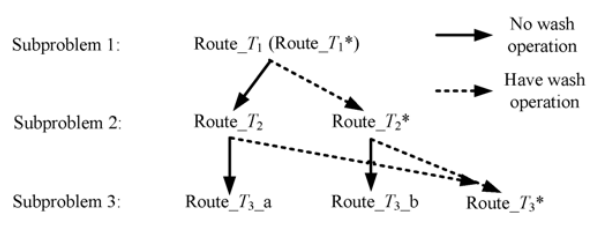
按调整后的顺序进行布线时，已布线的单元为下个布线任务的障碍，若不符合约束了则按[1]的修改规则进行修改。若修改失败则加入冲突列表在最后解决。到此，获得的所有路径是顶点不相交且边不相交的。最后对冲突列表进行处理，不将此前的路径视为障碍，用修改后的Lee算法为每个网络生成几条最短路径，选择一条没有公共相邻单元对的路径，则是边不相交，这条路径也需满足上述约束。当上述的方法无法生成不相交的路径方案时，使用[3]中的布局细化方法重新合成生物测定以提高可布线性。

针对在功能液滴先后经过交叉污染点之间插入清洗操作的情况，本文先对一个交叉污染点的情况进行分析，再扩展到多个交叉污染点。针对一个交叉污染点的情况，第一步是生成清洗液滴从源端口经过污染点到废液槽的路径，用修改后的Lee算法可以找到这两条子路径。接下来解决同步问题，根据两个功能液滴源端口到交叉点的路径估算到达时间，将后到达交叉点的功能液滴暂存片上存储器，等清洗液滴清洗完交叉点再让后到达的功能液滴经过交叉点，记录这三个液滴分别经过交叉点的时间。对于一个功能液滴经过多个交叉污染点的情况，其思路基本一致，先估计各个功能液滴到达交叉点的时间，再用片上存储器辅助同步液滴，记录时间以供下个交叉污染点估计到达时间。但对于多个交叉点，处理顺序会影响整个的工作时间，对于一个功能液滴，按其到达顺序依次处理是最高效的。对于多个液滴经过多个交叉点的情况，先建模为有向无环图，顶点为交叉污染点，有向边为交叉点的先后顺序，然后用[4]的算法进行拓扑结构的调整。若建模的图为有环图，这种情况下，本文并未讨论。

当前子问题的功能液滴布线可能会对下一个子问题的液滴布线造成交叉污染，所以任何一个当前子问题的功能液滴布线过程中，应把前驱子问题的布线和活动模块视为障碍。考虑该问题应结合上述两种布线方案，即统一不相交的布线方案和插入清洗操作的布线方案。对该问题，需考虑时间约束和可行性，总路径时间如下：



Ti\*为子问题i-1后插入清洗操作的子问题i的最大路径时间，Twi是子问题i-1后插入的清洗操作的时间，Ti是子问题i-1不插入清洗操作的子问题i的最大路径时间，当子问题i-1有插入清洗操作，xi=1，否则为0。连续子问题之间的问题可以用下面的样例的思路进行分析：//对统一两种方案这块提出的方法不够通用



**实验结果**：

该文将所提出的方法和两种基线方法在两个案例（多重体外检测和蛋白质生物测定）中进行对比，分别是基线方法1：一个子问题内的连续功能液滴布线步骤之间插入清洗操作，基线方法2：每个功能液滴后跟一个清洗液滴。结果表明，将两种方法统一的方案显著减少了布线时间。与[5]的工作相比，具有更少的交叉污染点，而使用更多的电极单元，更少的液滴传输时间。

**学习笔记：**

1、Lee 算法是网格路由中使用的一种流行技术，可以为每个网络获取单个液滴路径。 Lee 算法保证找到双引脚网络中两个引脚之间的最短路径。对于三引脚网络，[1]提出的修改后的Lee算法可以找到可行路径但不保证最小。

**疑惑：**子问题是怎么分解的？是分解了液滴整个路径吗？建模为图后，为什么不相交路径包括边不相交？二引脚网络的范围多大？长什么样？

[1] Su F, Hwang W, Chakrabarty K. Droplet routing in the synthesis of digital microfluidic biochips[C]//Proceedings of the Design Automation & Test in Europe Conference. IEEE, 2006, 1: 1-6.

[2] Sait S M, Youssef H. VISI Physical Design Automation: Theory and Practice[J]. 1994.

[3] Xu T, Chakrabarty K. Integrated droplet routing in the synthesis of microfluidic biochips[C]//Proceedings of the 44th annual Design Automation Conference. 2007: 948-953.

[4] Kahn A B. Topological sorting of large networks[J]. Communications of the ACM, 1962, 5(11): 558-562.

[5] Huang T W, Lin C H, Ho T Y. A contamination aware droplet routing algorithm for digital microfluidic biochips[C]//2009 IEEE/ACM International Conference on Computer-Aided Design-Digest of Technical Papers. IEEE, 2009: 151-156.