目录

[文献一 2](#_Toc145515889)

[文献二 4](#_Toc145515890)

[文献三 5](#_Toc145515891)

[文献四 7](#_Toc145515892)

[文献五 9](#_Toc145515893)

[文献六 12](#_Toc145515894)

[文献七 15](#_Toc145515895)

[文献八 18](#_Toc145515896)

[文献九 22](#_Toc145515897)

# 文献一

**文章名**：A Contamination Aware Droplet Routing Algorithm for Digital

Microfluidic Biochips

**作者**：Tsung-Wei Huang, Chun-Hsien Lin, and Tsung-Yi Ho∗

**年份**：2009

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

**针对问题**：DMFB的清洗时间和使用的电极单元数量

**问题表述及约束**：

输入：液滴网表、一组清洗液滴、堵塞位置、储存器位置和时间约束。

目标：所有液滴从源端口到目标污染点的布线，同时最大限度地减少污染点、执行时间和使用的电极单元，更好的容错能力。

约束：流体约束（静态：液滴不混和；动态：液滴在流动）、时间约束（不能太长以至于影响生物检测）和污染约束（交叉污染问题）

解决方案：

该方案分为三个步骤：预处理、内部污染感知布线和子问题间污染感知布线。

在预处理阶段，首先利用向量分解，然后为液滴选择一个初始方向，再将确定好方向的液滴的移动轨迹视为类似于车辆交通的问题，文中设计了一个成本函数，根据该函数为每个液滴计算最小成本路径，从而构建初始布线，使液滴在通道上有序流动，最大限度地减少使用的单元电极并减少拥塞的可能性。为了确定液滴的路径优先级，应用了一种基于布线资源的方程来确定液滴的路径顺序，以获得更好的可行性。//该方程及其定义详见4.2

在内部污染感知布线阶段，提出了一种k-最短路径算法来减少一个子问题内的污染点。首先，迭代地将未输送的液滴用上述算法找到最小成本路径，液滴流动后，将其加入优先级队列（优先级取决于路径上的污染点），然后迭代地使用更高的惩罚值来重构路劲图的边，得到一个更新后的图。对重构的路径图采用k-最短路径算法来选择下一种路径，文中建议k取3。当所有液滴按顺序流动至其目标时，应用了一种基于动态编程的路径压缩技术，来达到压缩路径和确定污染点生成的时钟周期，文中定义了一种将二维的路径图描述为一维的字符串的方法，然后设计了一条递归公式，该公式可以迭代地压缩路径。最大限度地减少生物测定的执行时间，该公式考虑全局拥塞和其他网络的干扰，以确定液滴的流动顺序。

为了清洗子问题内部的污染点，采用最小成本循环（MCC）算法来同时清洗这些污染点，最大限度地减少使用的电极单元和执行时间。该算法的构造由四个部分：源端口、一组清洗液滴、一组污染点以及一个水槽。算法的主要任务是节点容量分配和边的成本分配。

在子问题间污染感知布线阶段，应用了一种前瞻布线方案，该方案先解出子问题Pi，再推导Pi+1的解，从而可以推导子问题之间相互污染，再用子问题Pi的内部解决方案来处理。

实验结果：

本文设计的k-最短路径算法可以减少48%的污染点（相比于不用该算法），使用的电极单元数量减少8%，执行时间减少13%。

前瞻布线方案相比于不使用该方案，执行时间减少9%，使用的电极单元数量减少11%。

污染感知液滴布线算法相比于[1]的算法，使用的电极单元数量减少28%，执行时间减少12%。

**参考文献**：

[1] Zhao Y, Chakrabarty K. Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips[C]//2009 Design, Automation and Test in Europe Conference & Exhibition. IEEE, 2009: 1290-1295.

# 文献二

**文献名**：Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips

**作者**：Yang Zhao and Krishnendu Chakrabarty

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

**年份**：2009

**针对问题**：

**问题表述及约束**：

输入：生物测定的各级操作的时序表；模块在芯片上的位置；

目标：避免交叉污染；最小化液滴总布线长度；

约束：

1、不同的液滴布线不相交

2、流体约束（每个液滴都应至少保证一个单位的距离）

3、时间约束

**解决方案**：

三针网络视为两个二针网络的组合，因此仅考虑两针网络的布线。然后把芯片的结构看作一个图，顶点为电极单元，若两电极相邻，则对应顶点有边。把找到不相交的路径问题看作在图中找边不相交且顶点不共享的问题，该问题是一个NP完备问题，因此应用启发式方法。

针对子问题内的交叉污染的问题，该文献提出一种基于Lee算法的算法，来找到三个引脚之间的可行路线，但不能保证最小长度。一条路径布线后，其上所有电极视为障碍。文献中定义了一种定义二针网络优先级的方式，该公式计算出的结果越大优先级越小。按照该优先级按顺序对每个网络应用基于Lee算法的算法，由此获得的每条路径都要经过TDCC（时序延迟约束检查）和FCRC（流体约束规则检查），如果以上两个检查都失败，则应用[1]的修改规则来矫正路径，若矫正失败，则加入冲突列表中。将每个网络布线后，对冲突列表进行处理，还是应用基于Lee算法的算法，区别在于并不将之前已经找到的路径作为障碍，这样会得到可能相交的路径，但边仍不相交？？再进行TDCC和FCRC。

子问题之间可能会存在交叉污染的情况，则需要加入清洗操作，对于子问题Pi，加入清洗操作后，可能会导致Pi的时间变长，而Pi+1的时间减少，对此，文献提出一路径优化方案，可以最小化总的布线时间。

**实验结果**：

为了表征该布线方案，文献使用了两个案例进行测试：多重体外检测和DNA测序，结果表明能够显著地减少总体的布线时间。

**参考文献**：

[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.

# 文献三

**文献名**：Synchronization of Washing Operations with Droplet Routing for Cross-Contamination Avoidance in Digital Microfluidic Biochips

**作者**：Yang Zhao and Krishnendu Chakrabarty

**会议**：Proceedings of the 47th Design Automation Conference

**年份**：2010

**针对问题**：交叉污染和清洗时间优化

**问题表述及约束**：

输入：交叉污染点的位置；清洗缓冲流源端口和废液槽的位置；功能液滴到达交叉污染点的时钟周期；

目标：生成一个时间开销少和使用的清洗液滴数量少的清洗液滴和功能液滴的同步方案；

约束：

1、流体约束（每个液滴都应至少保证一个单位的距离）

2、时间约束

假设：一个清洗液滴的清洗能力足够清洗一个交叉污染点；

**解决方案**：

首先解决一个交叉污染点的清洗液滴路由与功能液滴路由同步的问题，并将该解决方案扩展到多个交叉污染点的同步。

针对一个交叉污染点的情况，第一步生成清洗液滴穿过交叉污染点的路径，用[14]的方法来生成该路径，第二步调整清洗液滴和功能液滴的到达顺序，例如当两个功能液滴的路径交叉，则预估两个功能液滴到交叉点的时间，第二个到达的功能液滴暂存在片上的存储器内，然后让清洗液滴在第一个功能液滴通过后清洗该交叉污染点，再从片上存储器上释放第二个功能液滴。这个过程中记录两个功能液滴到达交叉污染点的时钟周期，以便利用该信息来解决这两个功能液滴到达下一个交叉污染点的同步问题。

针对多个交叉污染点的情况，首先生成多个交叉污染点的清洗液滴的路径，再根据一个交叉污染点的情况进行同步，也就是暂存第二个功能液滴在存储器上，再记录相应的时钟周期，利用这些同步信息来处理下个交叉污染点。然而对于一个会经过多个交叉污染点的功能液滴，处理路径上的交叉污染点的顺序会导致时间开销不同。按该功能液滴穿过污染点的顺序依次进行处理是时间开销最短的。

**实验结果**：

为了表征该布线方案，文献使用了一个蛋白质测定案例进行测试，与两种布线方式进行比较，一种是等一个功能液滴经过全部交叉污染点后，再同时清洗所有污染点，这段时间内发生冲突的功能液滴都存储起来；第二种是让每个功能液滴都跟着一个清洗液滴。结果表明文献所提出的方法较于两种步线方法时间更短，较于基线方法二使用的清洗液滴更少。相对于[2]提出的布线方案，所用清洗时间更短，在较小规模下的测试结果表明，所提出的方法使用的清洗液滴数量接近，而清洗时间更短。较大规模情况下，所提出的方法使用的清洗液滴比[2]的方案更多而时间更短。

**参考文献**：

[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] Zhao Y, Chakrabarty K. Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips[C]//2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2009: 1290-1295.

# 文献四

**文章名**：A Contamination Aware Droplet Routing Algorithm for Digital

Microfluidic Biochips

**作者**：Tsung-Wei Huang, Chun-Hsien Lin, and Tsung-Yi Ho∗

**年份**：2009

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

**针对问题**：DMFB的清洗时间和使用的电极单元数量

**问题表述及约束**：

输入：液滴网表、一组清洗液滴、堵塞位置、储存器位置和时间约束。

目标：所有液滴从源端口到目标污染点的布线，同时最大限度地减少污染点、执行时间和使用的电极单元，更好的容错能力。

约束：流体约束（静态：液滴不混和；动态：液滴在流动）、时间约束（不能太长以至于影响生物检测）和污染约束（交叉污染问题）

解决方案：

该方案分为三个步骤：预处理、内部污染感知布线和子问题间污染感知布线。

在预处理阶段，首先利用向量分解，然后为液滴选择一个初始方向，再将确定好方向的液滴的移动轨迹视为类似于车辆交通的问题，文中设计了一个成本函数，根据该函数为每个液滴计算最小成本路径，从而构建初始布线，使液滴在通道上有序流动，最大限度地减少使用的单元电极并减少拥塞的可能性。为了确定液滴的路径优先级，应用了一种基于布线资源的方程来确定液滴的路径顺序，以获得更好的可行性。//该方程及其定义详见4.2

在内部污染感知布线阶段，提出了一种k-最短路径算法来减少一个子问题内的污染点。首先，迭代地将未输送的液滴用上述算法找到最小成本路径，液滴流动后，将其加入优先级队列（优先级取决于路径上的污染点），然后迭代地使用更高的惩罚值来重构路劲图的边，得到一个更新后的图。对重构的路径图采用k-最短路径算法来选择下一种路径，文中建议k取3。当所有液滴按顺序流动至其目标时，应用了一种基于动态编程的路径压缩技术，来达到压缩路径和确定污染点生成的时钟周期，文中定义了一种将二维的路径图描述为一维的字符串的方法，然后设计了一条递归公式，该公式可以迭代地压缩路径。最大限度地减少生物测定的执行时间，该公式考虑全局拥塞和其他网络的干扰，以确定液滴的流动顺序。

为了清洗子问题内部的污染点，采用最小成本循环（MCC）算法来同时清洗这些污染点，最大限度地减少使用的电极单元和执行时间。该算法的构造由四个部分：源端口、一组清洗液滴、一组污染点以及一个水槽。算法的主要任务是节点容量分配和边的成本分配。

在子问题间污染感知布线阶段，应用了一种前瞻布线方案，该方案先解出子问题Pi，再推导Pi+1的解，从而可以推导子问题之间相互污染，再用子问题Pi的内部解决方案来处理。

实验结果：

本文设计的k-最短路径算法可以减少48%的污染点（相比于不用该算法），使用的电极单元数量减少8%，执行时间减少13%。

前瞻布线方案相比于不使用该方案，执行时间减少9%，使用的电极单元数量减少11%。

污染感知液滴布线算法相比于[1]的算法，使用的电极单元数量减少28%，执行时间减少12%。

**参考文献**：

[1] Zhao Y, Chakrabarty K. Cross-contamination avoidance for droplet routing in digital microfluidic biochips[C]//2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2009: 1290-1295.

# 文献五

**文献名**：Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips//是之前会议的期刊版本

**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**：

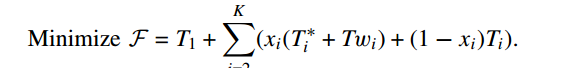
针对功能液滴路径不相交的问题，文中提出了一种寻找互不相交的功能液滴路径的方法并最小化总液滴路径。首先将路径规划问题分解为若干子问题，对于每个子问题分别求解。将芯片结构建模为图，顶点为电极单元，若两顶点对应的电极单元是相邻的，则这两顶点之间有边。对于三引脚网络（两源，一目标），将其转化为二引脚网络，计算三引脚网络中每个电极到目标的曼哈顿长度，根据最小曼哈顿长度找到一个混合点，从而将三引脚网络转化为三个二引脚网络（每个引脚到混合点）。给定一组 n 个 2 引脚网络，为这些网络找到可行的不相交路径的问题等价于寻找G中相互顶点不相交或边不相交的路径的问题。子问题Pi代表确定（ti,si）的二针网络与其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。



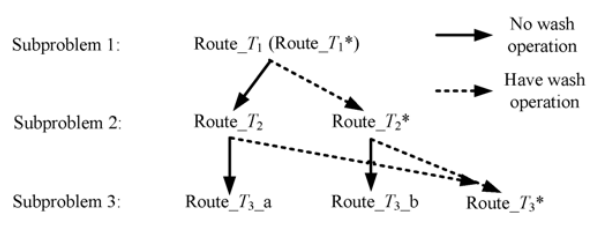
按调整后的顺序进行布线时，已布线的单元为下个布线任务的障碍，若不符合约束了则按[1]的修改规则进行修改。若修改失败则加入冲突列表在最后解决。到此，获得的所有路径是顶点不相交且边不相交的。最后对冲突列表进行处理，不将此前的路径视为障碍，用修改后的Lee算法为每个网络生成几条最短路径，选择一条没有公共相邻单元对的路径，则是边不相交，这条路径也需满足上述约束。当上述的方法无法生成不相交的路径方案时，使用[3]中的布局细化方法重新合成生物测定以提高可布线性。

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.

# 文献六

**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：**

**贡献点：**

**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；

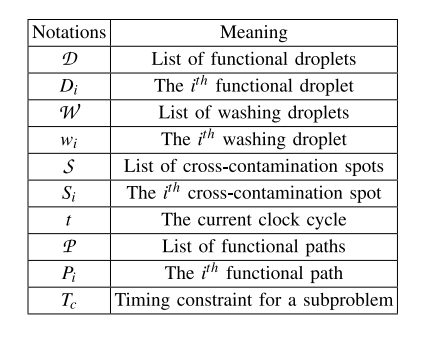
2、流体约束：每条路径之间的距离应大于规定的最小间距；

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：

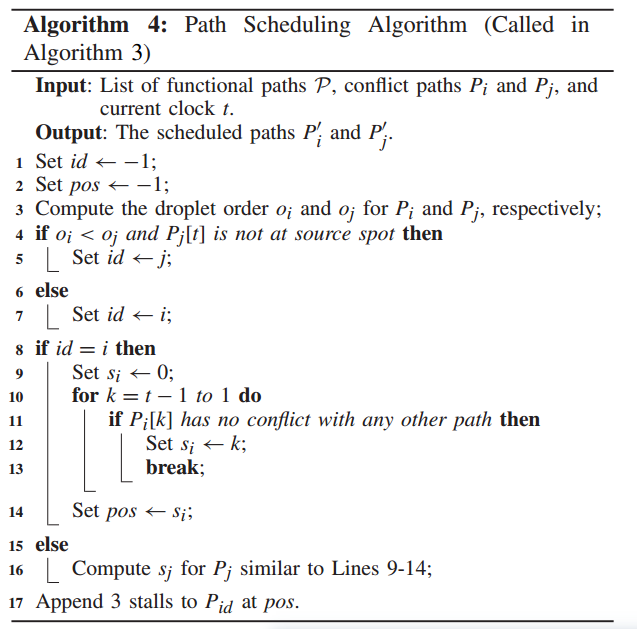
1、基于协商策略的撕裂和重布线[21]，[22]；

2、布线让步方法[7]；

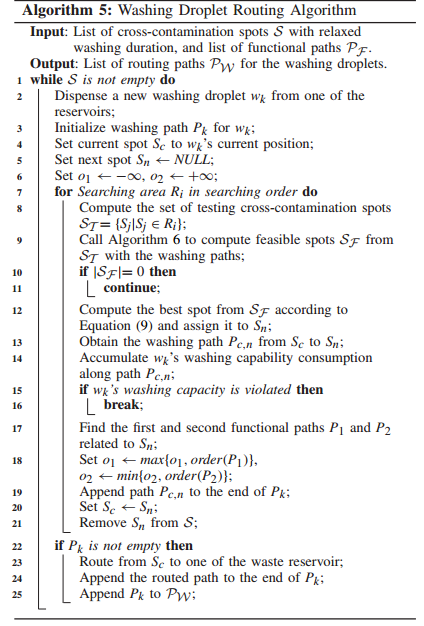
3、基于虚拟拓扑的布局细化，实现无死锁布线解决方案[23]。

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[1/16]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

# 文献七

**文献名**：A low-cost fluid-level synthesis for droplet-based microfluidic biochips integrating design convergence, contamination avoidance, and washing

**作者**：Arpan Chakraborty, Piyali Datta, Rajat Kumar Pal

**期刊**：Design Automation for Embedded Systems

**年份**：2018

**针对问题**：传统综合流程设计周期长的问题

**创新点：**以一种将操作集划分成块再分配对应的资源的思想进行调度；

**贡献点：**提出了一整个综合流程从而减少设计周期；

**痛点：**清洗液滴布线没考虑容量限制；只提出了决定清洗污染点顺序的策略，没有考虑清洗液滴的布线；

**问题模型及约束**：

输入：生物测定图；模块库；生物测定的最大允许时间；芯片的尺寸；

输出：

目标：

(1) 最大限度地缩短总体生物测定完成时间；

(2) 最大限度地减少污染位置的总数；

(3) 最大限度地缩短总体清洗时间；

(4) 最大限度地减少设计由于子任务之间的设计差距而导致的周期；

约束：

1、面积约束：所有资源布局的总面积小于芯片面积；

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、可访问约束：访问的不可重新配置资源不超过总数；

4、操作顺序约束；

5、分离约束：任何两个并发的资源应由一组电极单元隔开；

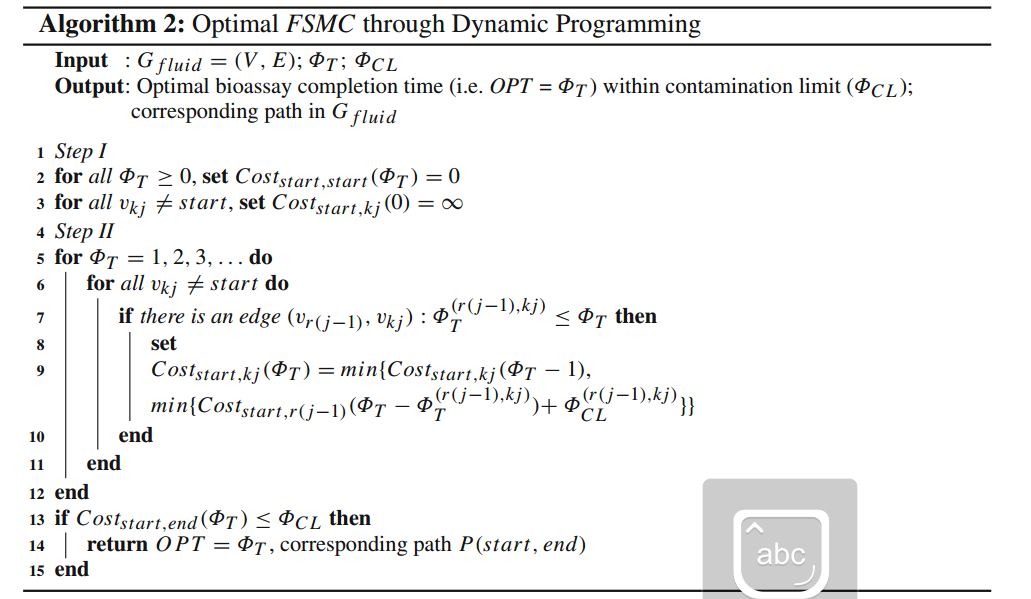
**解决方案**：

针对调度和绑定任务，本文提出了一种最小化生物测定的调度和绑定方案。其方法基于根据资源的分配（混合器在片上的分配）将操作集分成块进行调度的思想，首先用修改的列表调度算法（遵从操作依赖性约束）对操作列表进行排序，定义了如下的两种优先级规则：1）LRT：根据操作依赖性划分优先级，对于一个操作，其离操作依赖链的最后一个操作越远，其优先级越高；2）优先级规则2：根据资源分配的情况，尽可能地使更多的操作并发的执行以提高资源利用率（文中并没有详细介绍该规则的具体内容）；根据这两种优先级规则分配每个操作块的执行时间，主要是利用规则2来划分成操作块，即不断地寻找是否还有空间（包括已经用过的单元）能够分配混合器及绑定操作，利用规则1来安排执行时间。该方案最后输出若干个操作块。

在布局和布线阶段，首先根据绑定方案输出的一组操作块以及对应的混合器布局，可以在满足操作依赖下对各模块进行布线。具体地说，先将布局和布线问题看成若干个子问题，每个子问题包括一个位置和一条路径的解。一个操作块的结束时间取决于流体在路径流动的时间和上个操作块混合时间最久的操作，为了缩短操作块的完成时间，首先要解决布局问题。为了使用[1]提出的布局算法，需构建搜索空间，这个过程分为两步，第一采用两种策略（在寻找合适的位置时需计算所用的电极单元的数量，小于预定义的阈值视为可行；以及改变绑定方案生成模块位置的选择）为每个操作块的操作生成可行的位置，第二步在第一步生成的可行位置中为操作分配位置并生成液滴路径。当确定一个操作块后，要为下个操作块作最优的选择，先将搜索空间分为n级，每级对应一个操作块，每级中的顶点代表一个可行的位置选择，将从分级的搜索空间中提取 n 个组合，每个组合解决一个不同的子问题，以求解流体水平综合。//n级图中每一层选择一个操作，且确定好顺序，则获得了一个布线和布局方案

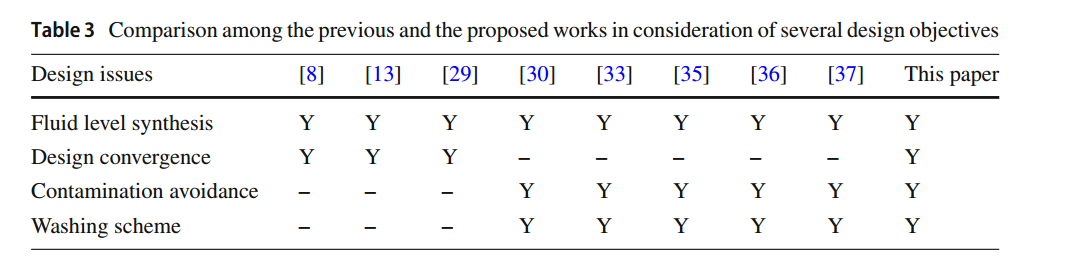
为了最小化生物测定长度和最小化污染点数量，需对产生污染的情况进行分析，第一种是因模块布局重叠产生的，第二种是因布线产生的。对于第一种情况，在第j层的操作之后对下层的顶点进行选择时，要选择时间开销最短的，也就是第j+1层选择的操作要在时间上尽可能早开始，从而能得到第j+1层选择的操作，也就能得到相应的边；对于第二种情况，每个布线子问题是两个操作块之间的问题，其也可以建模为2针网络来解决，[1]所提出的方法可以直接产生边不相交的路径规划，同时路径之间也设置了基于2针网络形成的矩形的周长的优先级的规则，优先级越高，该路径则优先布线，此后的路径都将自己之前布线的路径视为障碍。对于生成的边不相交的布线方案，其路径之间可能存在共享顶点，于是在生成布线方案时针对操作块之间进行操作的选择时，要考虑污染点越少的操作作为选择。

文中证明了只要从上述构造的n级图中按每层选择一个顶点从而形成一个路径，即可确定一个架构级综合方案（包括绑定、调度、布局、布线）。也证明了寻找以最小成本寻找该路径是一个NP完备问题。为了找到解决方案，将该问题看成是NP困难问题，通过算法2来找到能在伪多项式时间内计算出结果的一条n级图中的路径。其主要思想是定义一个成本，然后动态地去在n级图中选择成本最低的顶点加入路径。

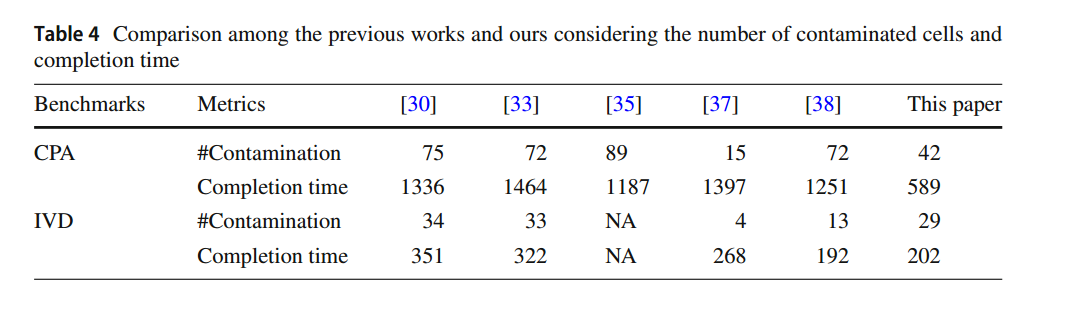


上述算法生成了产生最少污染点和最少执行时间的综合方案，但污染点依然需要清洗，于是需要考虑清洗液滴布线和功能液滴布线之间的同步问题使得生物测定的时间尽可能小。首先需要获得一个清洗计划的可行图，该图为有向图和无向图的结合，代表清洗液滴端口的顶点到各个代表交叉污染点的顶点之间为有向边，交叉污染点之间为无向边。于是该图的拓扑排序也对应着一个交叉污染点访问的方案，本文提出了两种策略来得到该清洗顺序方案。第一种是CLF（最近位置优先），它对各个交叉污染点根据清洗液滴端口到交叉污染点的曼哈顿距离进行优先级排序，当清洗液滴在交叉污染点进行清洗时，需要经过该点的功能液滴需等待。第二种是SATF（最短到达时间优先），即清洗液滴能越快达到污染点，该污染点优先级越高。CLF对于需要的清洗液滴路径长度最少，然而，SATF能够减少所有功能液滴的平均停滞时间，但需要更长的路径长度来清洗。

**实验结果**：



上图所示表格对比了本文跟之前工作的各项定性指标，结果表明本文提出的流程考虑更全面。



上图所示表格对比了本文和之前工作的污染点数量和生物测定的执行时间，结果表明本文的污染点相对较少，生物测定完成时间最短。

**学习笔记：**

1、架构级综合主要包括绑定和调度，物理综合主要是布局和布线。

**疑惑**：

**参考文献：**

[1] Zhao Y, Chakrabarty K (2012) Cross-contamination avoidance for droplet routing in digital microfluidic biochips. IEEE Trans Comput Aided Des Integr Circuits Syst 31(6):817–830

# 文献八

**文献名**：Fluid-level synthesis unifying reliability, contamination avoidance, and capacity-wastage-aware washing for droplet based microfluidic biochips//是2018

**作者**：Arpan Chakraborty, Piyali Datta, Rajat Kumar Pal

**期刊**：IET Computers and Digital Techniques

**年份**：2018

**针对问题**：传统综合流程设计周期长的问题

**创新点：**考虑了电极的共享会加速老化导致安全性的下降；

**贡献点：**提出了一整个考虑了容量浪费和电极老化的综合流程；

**痛点：**清洗液滴布线只考虑了从清洗液端口到污染点的布线，而没有考虑清洗液滴从污染点到废液槽的布线；

**问题模型及约束**：

输入：生物测定图；模块库；设计指标（包括芯片尺寸、时间限制等）；

输出：

目标：

(1) 最大限度地缩短总体生物测定完成时间；

(2) 最大限度地减少污染位置的总数；

(3) 最大限度地缩短总体清洗时间；

(4) 最大限度地减少容量浪费；

(5) 最大限度提升芯片的可靠性；

约束：

1、面积约束：所有资源布局的总面积小于芯片面积；

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、可访问约束：访问的不可重新配置资源不超过总数；

4、操作顺序约束；

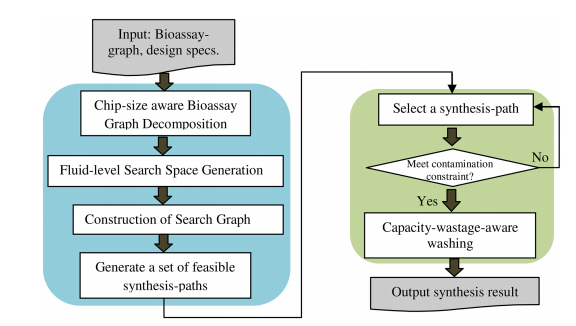
5、分离约束：任何两个并发的资源应由一组电极单元隔开；

6、可靠性约束：连续驱动电极的次数应受限；

7、污染点数量约束；

8、容量约束；

**解决方案**：



如上图所示整个的综合流程生成过程。

如该课题组2018的另一项关于流体级综合流程的工作一样，将操作集分解成操作块进行综合，分块是根据生物测序图中当前节点到最后一个节点的长度进行的（即对图进行深度优先遍历，对于顶点i，它最深能遍历多少个节点优先级就为多少），然后相同优先级的操作组成一块（即在逆拓扑排序序列中他们是在同一层的）。然后对操作块在满足上述约束下进行资源绑定和模块布局，在解决子问题i后，芯片上未使用的电极单元作为下一个子问题可用的资源，然后未使用的单元越多，资源利用率越低，生物测定完成时间更长；使用的单元越多，可靠性就会下降，所以这是一个需要根据需求进行权衡的问题。

对于布局，混合器可以布局的位置很多，本文提出的方法是对子问题逐个进行求解，首先生成初始布局（文中没说用什么生成策略），然后通过对已布局的模块进行平移或者旋转，可以得到很多新的放置位置的选择（这些新的布局选择可能包含不理想的单元（已经使用过的））。我们需要的是从这些选择中找到最优的（使下个子问题的解决时间最短同时具有可靠性）布局位置，这个搜索过程是在构造的搜索空间（当前子问题和下个子问题的可选择布局位置的集合）中进行的。然后对于n个子问题，可以构建出一个n层的有向图，每一层对应一个操作块i，每个顶点为操作块i可选择的布局位置。这样构建的搜索空间可能过于庞大，所以要先进行修剪，这个修剪过程是分两步进行的，第一步根据相邻的上个子问题未使用电极的数量进行修剪，第二步是计算每个选择包含的上个子问题已使用的单元数量，若超过自定义的阈值，则舍去。这样就获得了一个n级图，对于一个操作块的选择组合，其与当前子问题和下个子问题相关联，因此布局问题就转化为了找到n个操作块的选择组合。

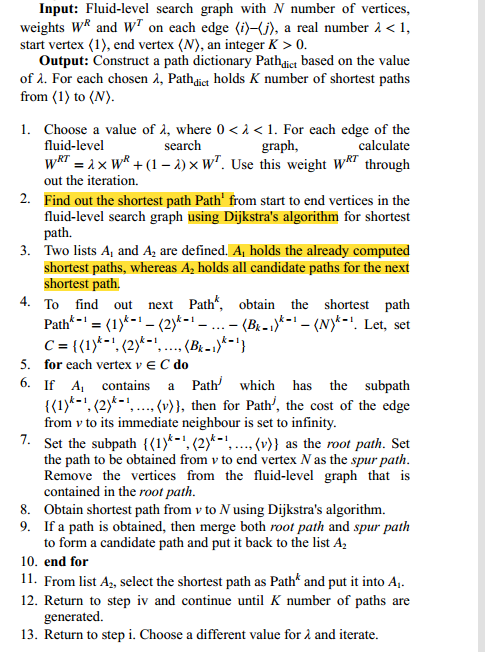
为了权衡生物测定时间和可靠性，本文对n级图的每条边赋予两个权重，第一个权重是连续两个子问题共享的电极单元数量（如代表i层的选择k和i+1层的选择r共享的电极单元数量），i层的一个选择与i+1的若干选择可能存在边，则确定使最小的边和顶点来解决这两个连续子问题。第二个权重是，它表示了子问题Pi+1的开始时间，可以通过子问题i的混合器类型和两个子问题的模块之间最大的路径长度。为了权衡这两个权重和作出最优的选择，构造了下述式子：

= λ × + (1 − λ) ×

通过调节λ来满足需求。

然而上述的综合流程忽略了交叉污染点的数量，可能导致违反时间约束，所以要对污染点的总数量设置一个上界。然后提前根据所给的n级图生成一个包含所有从开始到结束的可行路径的字典且这些路径要满足最小的∑，这样做可以方便在综合过程中当选择的路径污染点或清洗时间超过上界时，可以直接从字典里选择另一条路径，从而避免了重新寻找路径的开销。

构造字典阶段，可以建模为K-DMCS问题（K条不同最小成本的综合路径问题），该问题是NP困难问题，本文对此设计了一种算法来求解，如下所示：



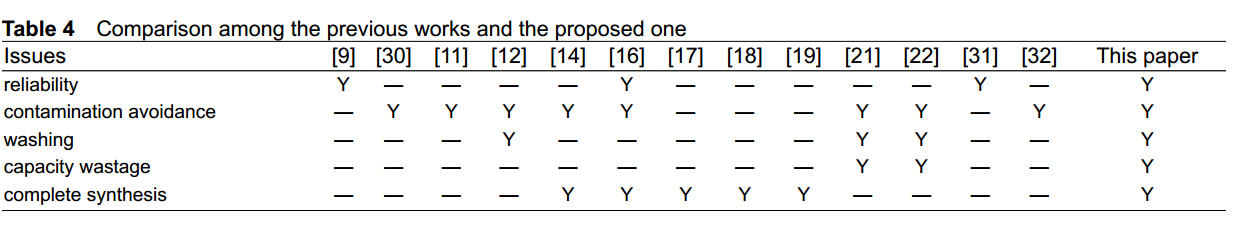
算法的主要思想是：首先用dijkstra算法找到从开始顶点到结束顶点的最短路径加入A1。为了找到下一条路径，从A1中提取出上一条路径Pathk-1，然后将Pathk-1的从开始顶点到顶点i（i从第二个顶点到导数第二个顶点）的子路径视为根路径，然后遍历A1中的全部路径，若有路径Pathj有相同的根路径，则在n级图中，将Pathj从顶点i到顶点i+1的边成本设为无穷大。然后在n级图中从顶点i到最后一个顶点使用dijkstra算法找到最短路径，若能找到这个最短路径，则和根路径合并成为Pathk，若没有，则i改为Pathk-1中i的下一个顶点，继续迭代。最终能构造出一个含有k个最短路径的路径字典。

考虑到容量浪费可能导致清洗失败，需要制定正确的调度策略才能避免这种情况。在清洗液滴布线阶段，也应该将存储中间结果的存储器视为障碍。本文设计了一个利用先前构造的字典来最小化清洗时间和减少容量浪费的清洗液滴布线算法。该算法大致描述如下：

从路径字典中选择满足污染点数量约束的一条路径，该路径是n个子问题的解构成的，对于每个子问题，首先识别目标污染点，对于电极单元构造一个成本函数：f（e）=g（e）+θ（e）（θ（e）=h（e）+w（e）+s（e））；其中g（e）代表清洗端口到当前电极单元的成本，h（e）代表当前电极单元到目标污染点的曼哈顿距离，w（e）用于避免清洗非目标污染点的电极单元，即若电极单元非目标污染点则该值设置为很大，否则为0，类似的，s（e）用于避免清洗存储器。利用该成本函数执行基于A\*的算法可以找到清洗路径。

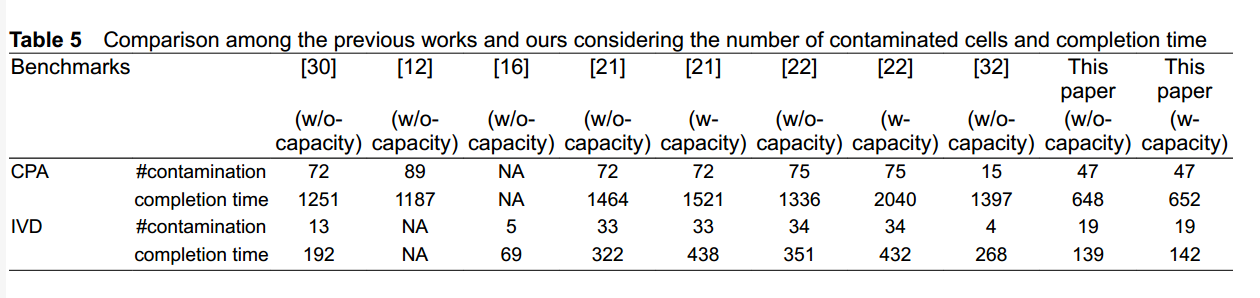
**实验结果**：

在考虑的因素数量方面，与之前的工作对比如下：



显然考虑了更多因素的影响。

在两个生物测定实例中，所提出的方法与之前的工作在完成时间和污染点数量对比如下：



结果表明，虽然所提出的流程会产生不少的污染点，但整体的生物测定完成时间最短，且可靠性方面也是最优的。

**学习笔记：**

1、液位综合的主要目标是最大限度地缩短生物测定完成时间。

2、流体级综合是解决一系列模块布局和中间液滴布线问题

**疑惑**：

1、液位搜索空间是什么意思？

**参考文献：**

# 文献九

**文献名**：Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

**作者**：ZhiPeng Huang, XiQiong Bai, TingShen Lan, XingQuan Li, and Geng Lin

**期刊**：IEEE Access

**年份**：2020

**针对问题**：清洗液滴布线方案各项指标的优化

**创新点：**以将子问题间的污染问题分配到子问题之中解决；

**贡献点：**提出了一个考虑了清洗能力限制的更高效的布线方案；

**痛点：**

**问题模型及约束**：

输入：DMFB阵列；一系列连续子问题；一组要连接的网络；一组清洗液滴；清洗液端口的位置；模块位置列表；时间约束；清洗容量约束；

输出：

目标：

(1) 所有功能液滴均在不违反任何约束的情况下从其源端口传送至目标电极单元，并且最大限度地减少污染点的数量、使用的电极数量和生物测定的执行时间。

约束：

1、时间约束

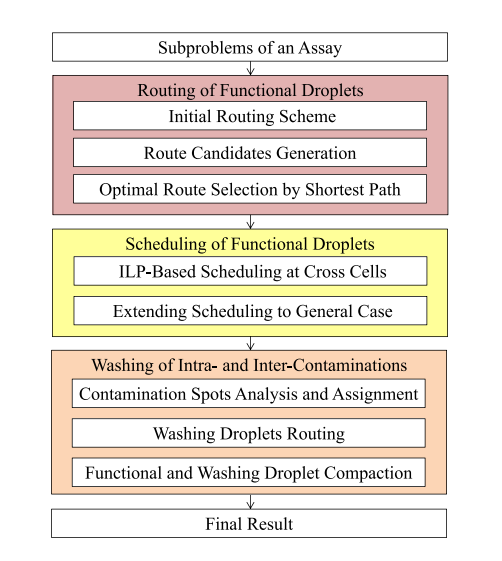
2、流体约束：每条路径之间的距离应大于规定的最小间距；

3、污染约束

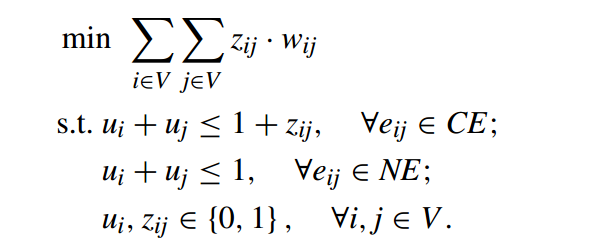
4、容量约束；

**解决方案**：

所提出的流程如下：

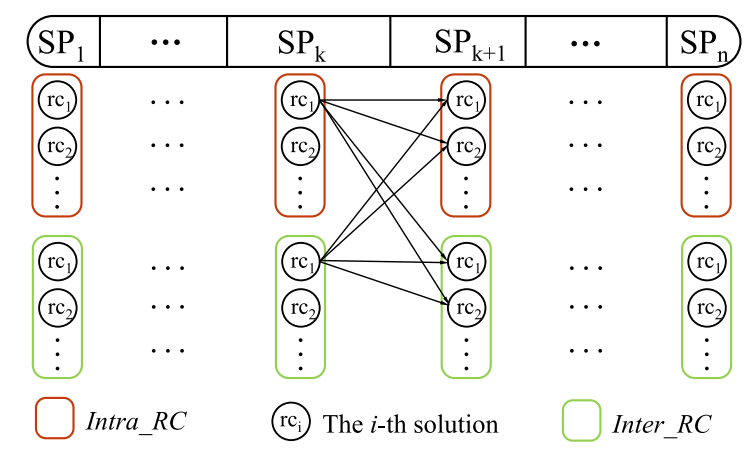


在功能液滴布线阶段，首先要生成初始的功能液滴布线，对于每个2针网络，文中设置其网络内只有两种布线，也就是两个引脚形成的矩形的一长一高形成的L形路径。然后基于所有网络的两种布线方案构建了一个交叉图。交叉图的顶点是每个网络的一种布线，若两个顶点属于同一个网络则用虚线相连，若两个顶点分别代表的路径有交叉点则用实线相连。同时每条实现边要赋予权重，该权重值为这两个顶点代表的路径相交的电极数量。可以将该交叉图视为二分图，进而为每个网络选择一条边可以建模为BLP（二元线性规划）问题，用ILP模型来求解该问题，如下所示，只要选择两个的顶点在二分图中有边且不属于同个网络，则把对应产生的交叉点数量加入到目标值，然后求目标值最小值。



上述获得的每个网络的初始布线可能存在违反约束的情况，例如路径与模块重叠。为此通过局部翻转L形路径来避免。由于路径与模块重叠有两种情况，本文定义了两种翻转规则来解决，针对L形路径的两条边都与模块重叠的情况，直接翻转即可，针对L形路径的一条边与模块重叠的情况，则将路径修改为在外部绕过模块。然后针对了子问题内和子问题间定义了两个成本函数来评价所得的这些路径的质量，分别根据这两种成本函数进行路径规划，会产生很多结果，将这些结果分别加入对应的路径集合。（对于一个子问题会有两个路径集合）。

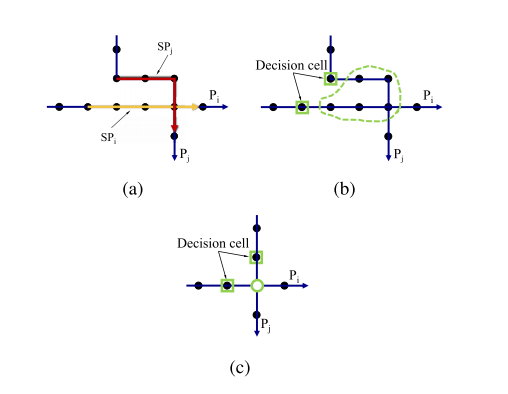
上述过程对于每个子问题形成了若干候选布线方案，为了给每个网络选择最优的路径使得污染点最少，文中提出了一个根据最短路径来找最优路径的模型。首先，定义了一个有向图，如下：



Spi代表子问题i，红色框内是子问题内的候选路径集，绿色框内是子问题间的候选路径集。这些有向边也根据边的两个顶点代表的路径产生的交叉污染点数量被赋予了权重。然后解决最短路径问题即可。

功能液滴布线后，需要考虑功能液滴的调度问题，本文提出了一种基于ILP模型的方法来确定功能液滴到达对应的决策电极单元（也就是交叉污染点的前一个电极单元）的时间，然后拓展到一般情况。这个ILP模型的约束有两个功能液滴分别到达自己的决策电极单元的时间间隔约束、功能液滴从源电极单元到目标电极单元的时间约束以及两个功能液滴分别到决策电极单元的顺序约束，目标是求解功能液滴从源电极单元到目标电极单元在不违法约束的情况下的最小时间。

两条相邻路径也可能违反流体约束，于是将上述ILP模型扩展到一般情况。对于存在子路径相邻的情况，将两条子路径在决策电极单元后的部分合并起来，如下图所示:

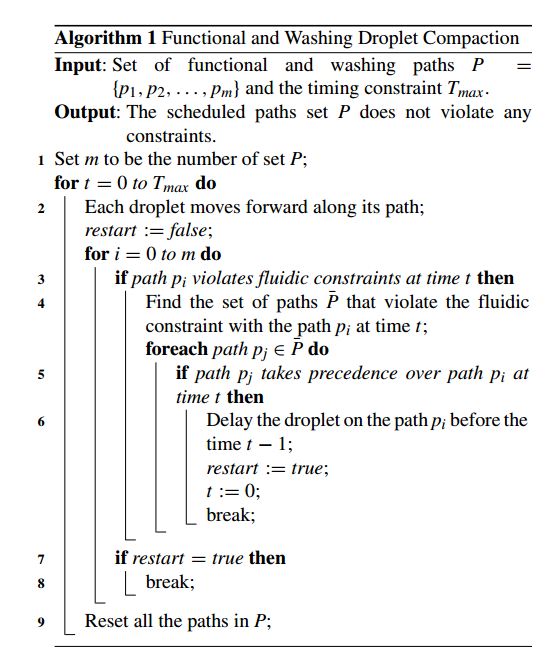


这样就转化为了上述的ILP模型，此外，若是某功能液滴路径的源电极单元和另一功能液滴路径的中间单元相邻的情况，则让后者先调度即可，类似地，目标电极单元和中间电极单元的相邻也这样调度。在上述ILP加入以上的时间约束即可扩展到一般情况。

从上述过程可以看出，子问题内和子问题间都可能发生交叉污染，文中提出了一种将交叉污染点分配给相邻的子问题从而减少生物测定执行时间的方法，此外还提出了一种高效的清洗液滴布线方案。针对子问题间的污染，文中将污染类型分为三种：P2M、P2P、M2P。然后按以下规则分配污染点，对于子问题k，要解决子问题内的污染点和M2P污染点，而产生的P2M和P2P分配给下个子问题k+1。为了保证清洗液滴可以彻底清洗污染点，本文引用了[1]关于清洗时间松弛方法。为了决定污染点的清洗顺序，考虑了污染点的两种权重，第一种是取决于一个污染点的第二个功能液滴到达的时间，时间越早优先级越高；第二种是取决于清洗液滴当前位置到污染点的距离，距离越短优先级越高。

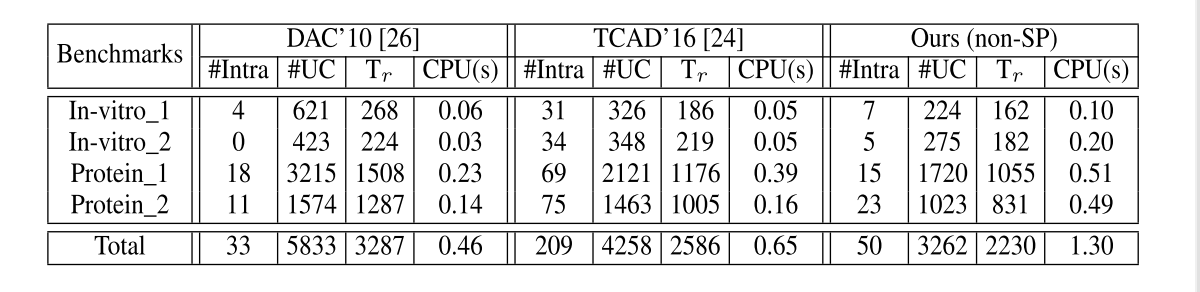
清洗液滴布线阶段，对于一个清洗液滴，首先在它周围的正方形搜索区域构建一个由可行的（可以在松弛后的清洗时间内到达污染点）候选污染点的集合。然后这个搜索区域逐渐扩大，直至找到所有可行的候选污染点。然后设计了一个成本函数来从候选的污染点中选择一个进行清洗，这个函数由清洗液滴到污染点的时间开销和污染点第二个功能液滴到达的时间加权组合而成。接下来，用A\*算法来对清洗液滴从源端口到所选的污染点进行布线。同时，应时刻记录清洗液滴的清洗能力，当清洗能力耗尽，用A\*算法为清洗液滴到废液槽进行布线。按上述过程重复执行，直至清洗液滴清洗能力耗尽或者清洗完所有可行的污染点。这样就得到了当前子问题的清洗液滴布线。

最后，同时对清洗液滴路径和功能液滴路径进行调度，同时最小化整个生物测定执行时间。算法如下：

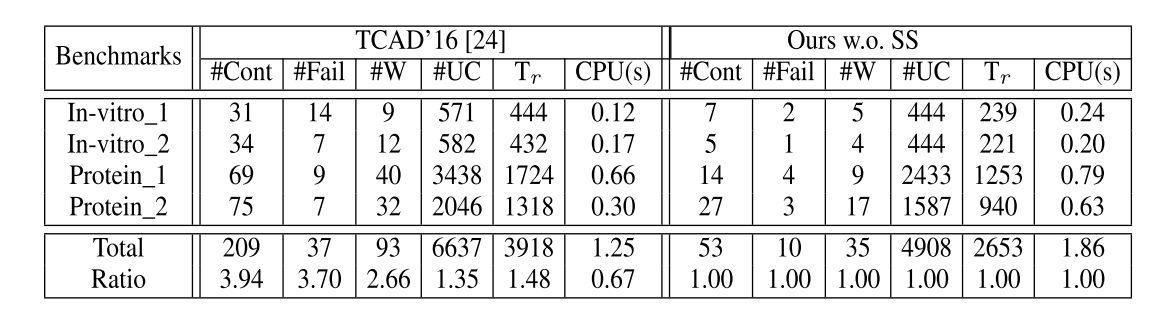


该算法主要针对的是不同路径之间的流体约束，主要思想是当发生违反流体约束的情况时，对于当前路径Pi，构造一个元素都是与当前路径发生冲突的路径Pj的集合，然后在满足文中所述的四种情况时，当前路径Pi内的液滴先通过，否则让Pi的液滴在上个时间单位停止。

**实验结果**：



与其他污染感知的布线工作对比，在不用最短路径算法的情况下，生物测定执行时间和使用的电极单元数量有明显优化。



与考虑容量限制的布线工作相比，生物测定执行时间、使用的电极单元数量、总污染点数量、使用的清洗液滴数量和清洗失败的清洗液滴数量有大幅度优化。

**学习笔记：**

1、液位综合的主要目标是最大限度地缩短生物测定完成时间。

2、流体级综合是解决一系列模块布局和中间液滴布线问题

**疑惑**：

1、路径压缩是什么意思？//是全局地考虑所有路径的调度，然后压缩时间吗？

**参考文献：**

[1] Yao H, Wang Q, Shen Y, et al. Integrated functional and washing routing optimization for cross-contamination removal in digital microfluidic biochips[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 35(8): 1283-1296.