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，有另外一篇比较有难度，看完了在整理。

目录

[文献一 2](#_Toc145448387)

[文献二 5](#_Toc145448388)

[文献三 9](#_Toc145448389)

# 文献一

**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**：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：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

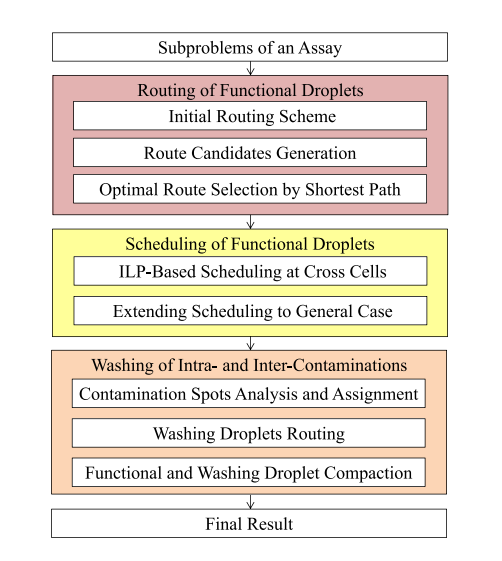
2、流体约束：每条路径之间的距离应大于规定的最小间距；

3、污染约束

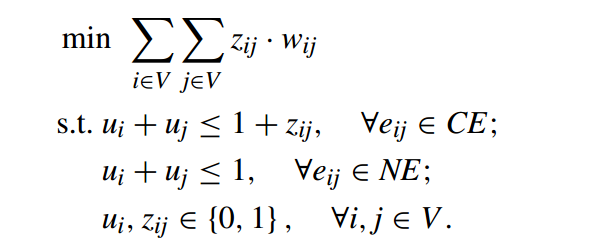
4、容量约束；

**解决方案**：

所提出的流程如下：

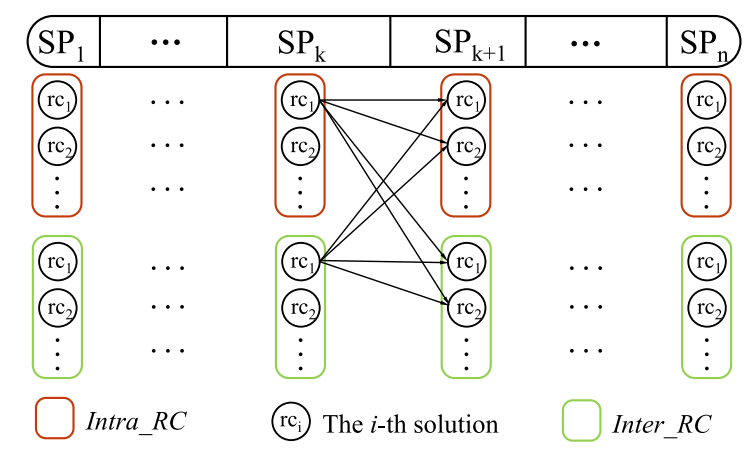


在功能液滴布线阶段，首先要生成初始的功能液滴布线，对于每个2针网络，文中设置其网络内只有两种布线，也就是两个引脚形成的矩形的一长一高形成的L形路径。然后基于所有网络的两种布线方案构建了一个交叉图。交叉图的顶点是每个网络的一种布线，若两个顶点属于同一个网络则用虚线相连，若两个顶点分别代表的路径有交叉点则用实线相连。同时每条实现边要赋予权重，该权重值为这两个顶点代表的路径相交的电极数量。可以将该交叉图视为二分图，进而为每个网络选择一条边可以建模为BLP（二元线性规划）问题，用ILP模型来求解该问题，如下所示，只要选择两个的顶点在二分图中有边且不属于同个网络，则把对应产生的交叉点数量加入到目标值，然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



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）。

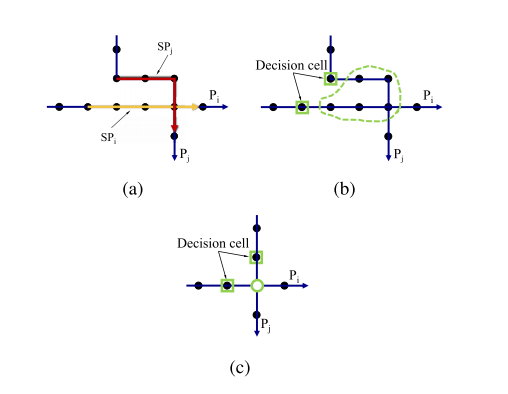
上述过程对于每个子问题形成了若干候选布线方案，为了给每个网络选择最优的路径使得污染点最少，文中提出了一个根据最短路径来找最优路径的模型。首先，定义了一个有向图，如下：



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。

**实验结果**：

**学习笔记：**

**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：**

# 文献二

**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；

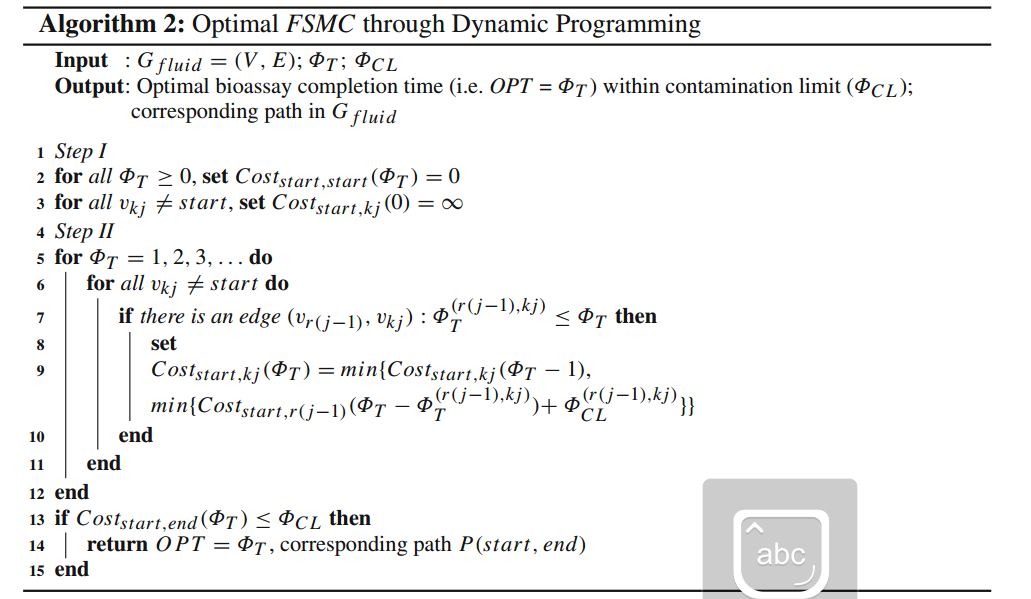
**解决方案**：

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

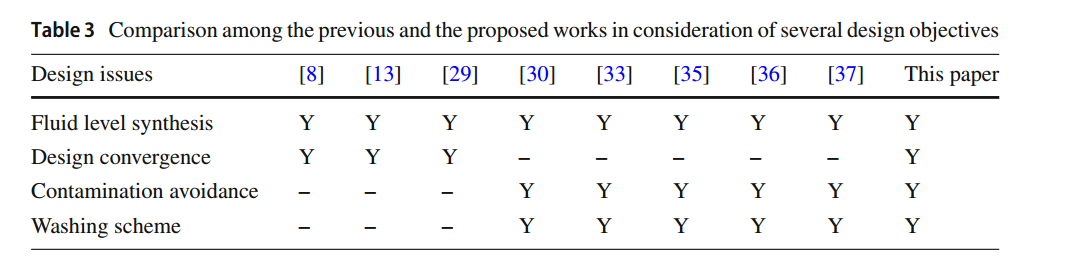
为了最小化生物测定长度和最小化污染点数量，需对产生污染的情况进行分析，第一种是因模块布局重叠产生的，第二种是因布线产生的。对于第一种情况，在第j层的操作之后对下层的顶点进行选择时，要选择时间开销最短的，也就是第j+1层选择的操作要在时间上尽可能早开始，从而能得到第j+1层选择的操作，也就能得到相应的边；对于第二种情况，每个布线子问题是两个操作块之间的问题，其也可以建模为2针网络来解决，[1]所提出的方法可以直接产生边不相交的路径规划，同时路径之间也设置了基于2针网络形成的矩形的周长的优先级的规则，优先级越高，该路径则优先布线，此后的路径都将自己之前布线的路径视为障碍。对于生成的边不相交的布线方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

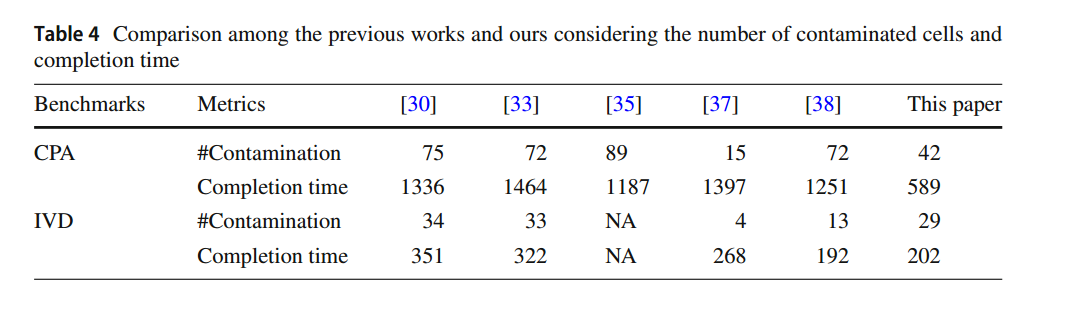


上述算法生成了产生最少污染点和最少执行时间的综合方案，但污染点依然需要清洗，于是需要考虑清洗液滴布线和功能液滴布线之间的同步问题使得生物测定的时间尽可能小。首先需要获得一个清洗计划的可行图，该图为有向图和无向图的结合，代表清洗液滴端口的顶点到各个代表交叉污染点的顶点之间为有向边，交叉污染点之间为无向边。于是该图的拓扑排序也对应着一个交叉污染点访问的方案，本文提出了两种策略来得到该清洗顺序方案。第一种是CLF（最近位置优先），它对各个交叉污染点根据清洗液滴端口到交叉污染点的曼哈顿距离进行优先级排序，当清洗液滴在交叉污染点进行清洗时，需要经过该点的功能液滴需等待。第二种是SATF（最短到达时间优先），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

# 文献三

**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 & Digital Techniques

**年份**：2018

**针对问题**：电极多次驱动会导致老化

**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；

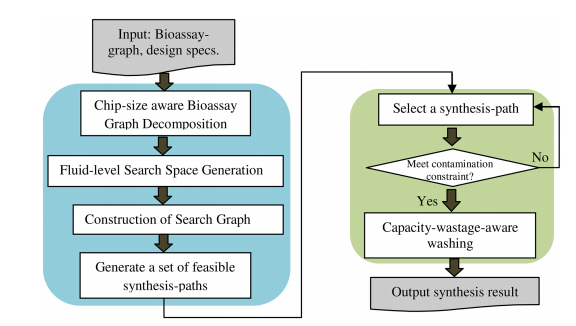
5、分离约束：任何两个并发的资源应由一组电极单元隔开；

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。

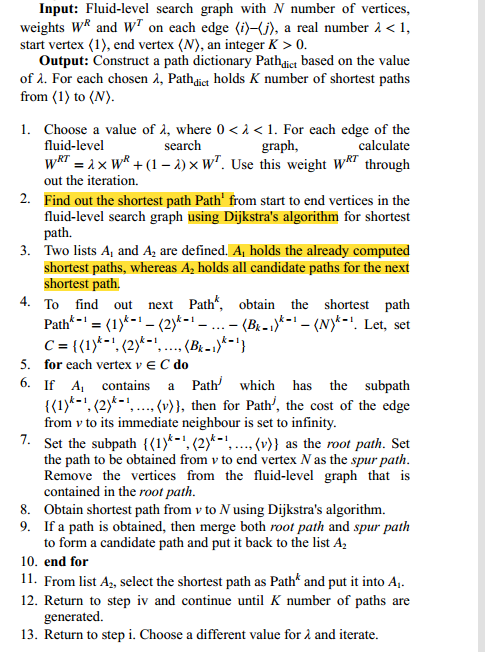
为了权衡生物测定时间和可靠性，本文对n级图的每条边赋予两个权重，第一个权重是连续两个子问题共享的电极单元数量（如代表i层的选择k和i+1层的选择r共享的电极单元数量），i层的一个选择与i+1的若干选择可能存在边，则确定使最小的边和顶点来解决这两个连续子问题。第二个权重是，它表示了子问题Pi+1的开始时间，可以通过子问题i的混合器类型和两个子问题的模块之间最大的路径长度。为了权衡这两个权重和作出最优的选择，构造了下述式子：

= 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：



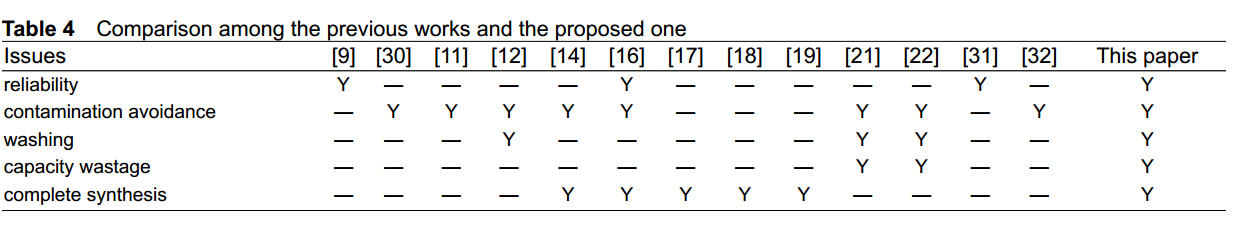
算法的主要思想是：首先用dijkstra算法找到从开始顶点到结束顶点的最短路径加入A1。为了找到下一条路径，从A1中提取出上一条路径Pathk-1，然后将Pathk-1的从开始顶点到顶点i（i从第二个顶点到导数第二个顶点）的子路径视为根路径，然后遍历A1中的全部路径，若有路径Pathj有相同的根路径，则在n级图中，将Pathj从顶点i到顶点i+1的边成本设为无穷大。然后在n级图中从顶点i到最后一个顶点使用dijkstra算法找到最短路径，若能找到这个最短路径，则和根路径合并成为Pathk，若没有，则i改为Pathk-1中i的下一个顶点，继续迭代。最终能构造出一个含有k个最短路径的路径字典。

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。

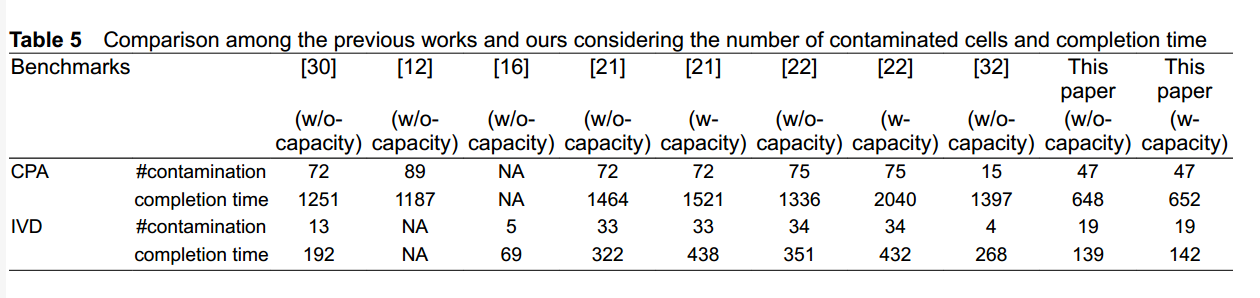
**实验结果**：

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：**