
基于时间序贯寻优的免疫分析仪芯片检测调度方案设计

摘要

生产调度在现代化生产制造系统中起着十分重要的作用，高效的生产调度方案将对降低生产成本具有重要意义。在化学发光免疫分析仪生产的背景下，本文针对非线性流水线多任务调度的优化问题，以最小化设备总运行时间为目标，结合因设备运行情况和空间容量带来的时空约束，建立了相应的生产调度模型。并基于序贯优化、贪心算法等思想对模型进行了有效求解。

针对问题一，首先明确了各生产阶段之间的联系，然后，针对生产任务非线性及平台数量资源有限所造成的约束条件量化难的问题，结合单一种类芯片生产条件特点，从平台数量和使用时间的角度对约束条件进行了分析，并将约束体现在时间线的对应关系上。随后，建立了依次最小化产品投放时间的数学模型，运用搜索求解的方法获取了不同磁珠加样时间下的最优投放时间序列，**耗时最短的投放结果见附录 A 附表 1 和附表 2。**以第一片芯片结束检测流程的时间作为起始时间点，**取时间平均后求得芯片 A、B 在单位时间可完成芯片检测的最大数量分别为 70.48 片和 44.01 片。**

针对问题二和问题三，首先，将因种类差异造成芯片进出顺序不一致的情况纳入约束条件，并将该约束体现在时间区间的对应关系上。而后，结合贪心算法的思想，将全局优化问题转化为局部优化问题，每次调度均选取满足约束条件的投放时间最小的芯片，并按芯片的种类和数量对投放情况进行划分，最终获取耗时最短的最佳投放方案。**问题二的结果见附录 A 附表 3，问题三的结果见附录 A 附表 4。求得两类芯片全部完成检测至少需要 13385s，三类芯片全部完成检测至少需要 13124s。**从对比试验可以看出，算法有效降低了任务的总耗时。

针对问题四，首先分析了约束条件的可推广性，然后将问题二和问题三的模型及算法推广到参数可变的一般情形，最后**验证了算法推广到 N 种类型芯片的可行性**，对此类调度问题建立了一个通用的数学模型，并结合相关实验证明了此通用模型的可推广性。

针对问题五，首先将问题拆分为依次处理两类芯片和三类芯片的问题，在加入芯片的时间节点进行合适的数据传递，从而完成问题的转换。**问题五的结果见附录 A 附表 5，最终求得全部完成检测的时间为 12640s。**

最后，我们分析了模型的优缺点，提出了模型的推广和改进方向。

本文所建立模型和算法通用性好，适用于各类生产调度问题，存在的创新点有：对复杂约束条件结合具体情况进行合理分类，将空间约束转化为时间约束；寻优执行单次任务时的最优磁珠加样时间，进一步缩短任务的总完成时间；使用贪心算法的思想，节省了寻找全局最优解所耗费的大量资源，算法简洁高效，易于工程实现。

关键词：芯片检测调度 序贯寻优 贪心算法 资源约束 局部最优

目录

摘要.....	I
1 问题综述.....	1
1.1 问题背景.....	1
1.2 问题提出.....	2
2 问题分析.....	2
3 模型假设与符号说明.....	3
3.1 模型基本假设.....	3
3.2 符号说明.....	3
4 问题一：模型建立与求解.....	4
4.1 约束条件的分析.....	4
4.2 模型建立.....	5
4.3 算法分析求解.....	6
5 问题二：模型建立与求解.....	8
5.1 约束条件分析.....	8
5.2 模型建立.....	10
5.3 算法分析与求解.....	10
6 问题三：算法分析与求解.....	13
6.1 算法分析.....	13
6.2 问题求解.....	16
7 问题四：推广分析与举例.....	17
7.1 问题二的一般化推广.....	17
7.2 问题三的一般化推广.....	19
7.3 N 种芯片的推广.....	20
8 问题五：算法分析与求解.....	21
8.1 算法分析.....	21
8.2 问题求解.....	22
9 模型评价与推广.....	23
9.1 模型的优点.....	23
9.2 模型的不足.....	23
9.3 模型的推广与改进.....	23
参考文献.....	24
附 录.....	25
附录 A: 附表.....	25
附录 B: 主要程序/关键代码.....	49

1 问题综述

1.1 问题背景

全自动化学发光免疫分析仪是医学上对人体体液样本进行检验的一种仪器，它实现了对医学上免疫诊断的整个流程的自动化操作，克服了人工检测存在的效率低、结果估计精度差、耗时费力等缺点。

化学发光免疫分析仪的结构如图 1，它包括前处理工作台、大转盘（转盘本身具有温育功能，亦称温育盘）和转盘上方固定的 8 个清洗盒、1 个磁珠加样盒，以及 1 个可以转动的检测盘等。转盘上均匀分布着用来放置芯片的 40 个卡槽。含有人体体液样本的生物芯片需要在化学发光免疫分析仪里依次经过前处理（进仓读码、激光超声、过滤、定量、R1R2 加样）、第一次温育、磁珠加样、第二次温育、清洗以及检测等步骤。

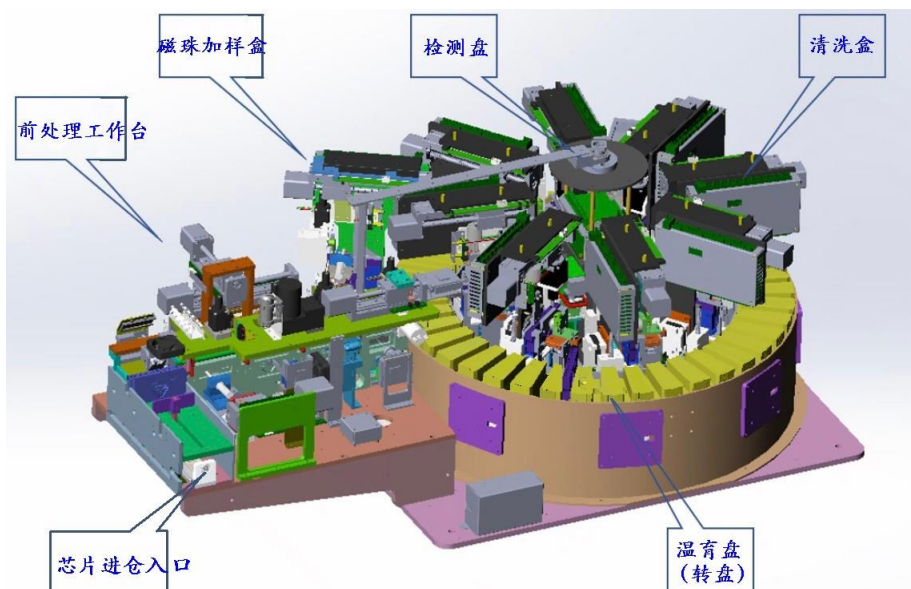


图1 化学发光免疫分析仪

化学发光免疫分析仪各工序流程及对应时长如表 1 所示。其中，前期处理的每个步骤一次只能处理一片芯片，时长 30s，且每片芯片的检测流程一旦开始便不能停止，直到检测完成。

表1 化学发光免疫分析仪各工序所需时间

工序编号	工序	所需时间
1	前处理	150s
2	转盘转动对位	6s
3	第一次温育	0~60min, 不同类型芯片时长不同
4	转盘对位磁珠盒	8s
5	磁珠工位	[21s, 25s]
6	转盘转动对位	8s
7	第二次温育	0~30min, 不同类型芯片时长不同
8	转盘对位清洗盒	16s
9	清洗工位	≥ 325s

10	检测盘对位	12s
11	检测工位	25s

1.2 问题提出

为了提高化学发光免疫分析仪的调度效率,结合上述问题背景,我们需要建立数学模型和相关算法,在保证各检测工序不发生冲突的前提下,使一组待检测芯片总的完成时间尽可能小,并将对应的调度结果表填写至论文附录.具体的问题背景及需求解的问题如下:

- (1) 问题 1: A 型芯片第一次温育时间 10 分钟,第二次温育时间 5 分钟; B 型芯片第一次温育时间 40 分钟,第二次温育时间 20 分钟. 求解单位时间 (1 小时) 内每型芯片最多可完成芯片检测的数量.
- (2) 问题 2: 现有 A, B 两种类型的芯片各 80 片, 求解两类芯片全部检测完毕所需要的时间.
- (3) 问题 3: C 型芯片第一次温育时间 25 分钟,第二次温育时间 10 分钟. 现有 A 型芯片 60 片、B 型芯片 50 片、C 型芯片 50 片, 求解芯片全部检测完毕所需要的时间.
- (4) 问题 4: 针对问题 2 和问题 3, 将模型和算法分别推广到芯片数量为 m_i , 其第一次温育时间为 a_i 分钟、第二次温育时间为 b_i 分钟 ($i=1,2,3$) 的一般情形. 进一步, 将其推广到 N 种类型的芯片, 并自行举例验证.
- (5) 若开始有 A 型芯片 60 片、B 型芯片 70 片, 运行 30 分钟后, 新到 C 型芯片 20 片, 求解全部完成所需要的时间.

2 问题分析

本文要解决的是具有空时约束的流水线产品的调度优化问题. 要求在芯片检测工序不发生冲突的前提下, 最小化产品检测总的完成时间. 问题一要解决的是确定条件下单个种类芯片的调度优化问题, 问题二、三、四、五要解决的是多种类芯片条件下的调度优化问题. 其中问题二、三分别针对芯片数量和温育时间固定时两种类型和三种类型芯片的处理; 问题四扩展到未知芯片数量和温育时间下两种类型和三种类型芯片的生产调度问题, 并要求推广到 N 种类型的芯片; 问题五则考虑了生产中途添加芯片种类的情况. 五个问题是一脉相承, 层层递进的.

本题的难点之一在于对约束条件的量化上. 在问题背景下, 芯片在机器上按照生产序列进行连续不断地无时间等待加工. 生产序列在执行过程中所消耗的时间不同, 所使用的空间资源也是有限的, 部分过程还存在使用同一平台进行工作的情况. 为了避免冲突, 在处理问题的过程中, 需要合理量化时间约束和空间约束, 也可以进行约束间的相互转化.

对于问题一, 芯片类型单一, 温育时间确定. 生产调度过程中, 各芯片在各环节所消耗的时间一致, 最小化产品总检测时间等价于依次最小化产品的投放时间. 在零等待工艺约束下, 后投放的产品可以推断先投放产品的工作状态. 因此, 可以采取逐个优化芯片投放时间的方式, 递增单个芯片的初始解大小, 搜索得到各产品满足约束条件的最短时间, 从而实现总探测时间的最小化.

对于问题二, 芯片类型有两种, 温育时间已知. 与问题一不同的点在于, 问题二需要考虑两种芯片进入的顺序, 通过合理控制 A、B 两种芯片进入检测流程的顺序, 来减少完成检测的时间. 生产调度过程中, 芯片 A、B 在部分环节所消耗的时间不一致, 最小化产品总检测时间将不等价于依次最小化产品投放的间隔时间. 若一味追求最小化全局

检测时间, 所考虑的数据维度和约束条件将极为复杂, 但若采用贪心算法的思想, 将全局最优问题转化为局部最优问题, 在调度过程的每一步都选取使得当下总的完成时间最小的方案, 不考虑对未来的影响, 将省去了为找全局最优解要穷尽所有可能而必须耗费的大量时间. 因此, 问题二可以借助贪心算法的思路, 逐个选取满足约束条件的使当下完成时间最小的芯片进入检测流程, 从而获取最终的优化解.

对于问题三, 芯片类型有三种, 温育时间已知. 相较于问题二, 问题三增加了一种芯片的种类. 虽然芯片的种类增加了, 但是解题思路是一致的, 通过合理控制芯片 A、B、C 三种芯片进入检测的流程, 来减少完成检测的时间. 合理控制的点与问题二一样, 逐个选取满足约束条件的使当下完成时间最小的芯片进入检测流程, 从而获取最终的优化解.

对于问题四, 对芯片种类、芯片数量和温育时间进行推广. 针对所建立的模型和算法, 首先对两种类型的芯片场景进行推广, 分析其推广的可行性和对应的算法流程, 三种类型的芯片场景推广流程类似, 最后, 可以运用数学归纳法来探究模型和算法推广到 N 种芯片的可行性.

对于问题五, 芯片数量刚开始有两种, 设备运行一段时间后芯片数量增加到三种, 温育时间已知. 由于问题二中的算法 (算法二) 可以同时处理两种类型的芯片, 问题三中的算法 (算法三) 可以同时处理三种类型的芯片, 因此可以结合这两种算法来求解问题五. 可以在设备前期运行时使用算法二, 当设备开始添加第三类芯片时, 将处理数据传递给算法三, 最终实现芯片进入检测流程的有效控制, 获取较小的处理时间.

3 模型假设与符号说明

3.1 模型基本假设

- (1) 芯片在检测过程中不会有突发情况, 不考虑设备在运行过程中损坏、芯片掉落等事件的发生.
- (2) 转盘一次只能转动对位一个芯片, 两个及以上芯片在转盘转动时有时间重合就会引起转盘的使用冲突.
- (3) 磁珠工位和清洗工位的用时是可变的, 但在单次任务过程中用时固定, 且假设芯片清洗所需要的时间固定为 325s.
- (4) 转盘或检测盘在每次的转动对位过程中, 无论如何转动、移动多少距离, 所需时长是一样的.
- (5) 芯片进入检测流程后不能停止, 直到检测完成, 且每一步骤不能超时.
- (6) 转盘转动对位期间, 芯片处在原有位置, 转盘对位结束后, 芯片才改变位置.
- (7) 每道工序开始和结束的时间均为整数.

3.2 符号说明

本文定义了如表 2 所示的 10 个使用次数较多的符号, 其余符号在使用时注明.

表2 符号说明

符号	含义	单位
s_{ij}	第 i 个芯片所处的第 j 个状态	/
a_i	状态转换间的第 i 个动作	/
$T_{k,j}$	第 k 类芯片在第 j 个阶段所消耗的时间	s
t_{ij}	第 i 个芯片在第 j 个状态间的时间节点	s

符号	含义	单位
TS_i	第 i 个芯片使用转盘的时间区间	s
TW_i	第 i 个芯片处于卡槽的时间区间	s
TQ_i	第 i 个芯片处于清洗盒的时间区间	s
TZ_i	第 i 个芯片使用磁珠加样盒的时间区间	s
TC_i	第 i 个芯片使用检测盒的时间区间	s
k_i	第 i 类芯片进入检测流程的个数	/

4 问题一：模型建立与求解

4.1 约束条件的分析

设 $s_j, j=1,2,\dots,6$ 为第 i 个芯片所处的第 j 个状态，分别对应前处理、第一次温育、磁珠工位、第二次温育、清洗及检测。状态之间的转换通过转盘转动对位或检测盘转动对位实现。 $a_j, j=1,2,\dots,5$ 为状态转换间的动作，分别对应转盘第一次转动对位、转盘对位磁珠盒、转盘第二次转动对位、转盘对位清洗盒和检测盘对位。 $T_{k,j}, j=1,2,\dots,11$ 为第 k 类芯片在第 j 个阶段所消耗的时间，由于一组检测芯片中，不同芯片仅在 3、7 阶段存在时间消耗差异，因此其他阶段的下标可省略 k 。 $t_{ij}, j=1,2,\dots,7$ 为第 i 个芯片在第 j 个状态间的时间节点，分别对应芯片进仓时间、第一次温育起始时间、磁珠加样起始时间、第二次温育起始时间、清洗起始时间、检测起始时间和检测结束时间，各时间节点之间的关系可通过消耗时间 T 联系起来。

$$\begin{cases} t_{i,2} = t_{i,1} + T_{k,1} + T_{k,2} \\ t_{i,3} = t_{i,2} + T_{k,3} + T_{k,4} \\ t_{i,4} = t_{i,3} + T_{k,5} + T_{k,6} \\ t_{i,5} = t_{i,4} + T_{k,7} + T_{k,8} \\ t_{i,6} = t_{i,5} + T_{k,9} + T_{k,10} \\ t_{i,7} = t_{i,6} + T_{k,11} \end{cases} \quad (1)$$

图 2 直观地展示了第 i 个芯片的工作流程及对应的时刻和位置， k 值的取值对应于第 i 个芯片的种类。各平台的加工数量有限，除卡槽及清洗盒的数量为 8 个和 40 个外，其余的平台数量只有 1 个，且部分平台在一次芯片的处理过程中可能会使用多次。图中相同的颜色块即表示该类流程在一次加工过程中占据了同一平台资源，但时间不连续。如状态 s_2 和 s_4 均需要使用卡槽，动作 $a_1 \sim a_4$ 均依靠转盘的转动。因此，我们需要针对设备运行情况和空间容量，对各平台所产生的约束依次进行分析。

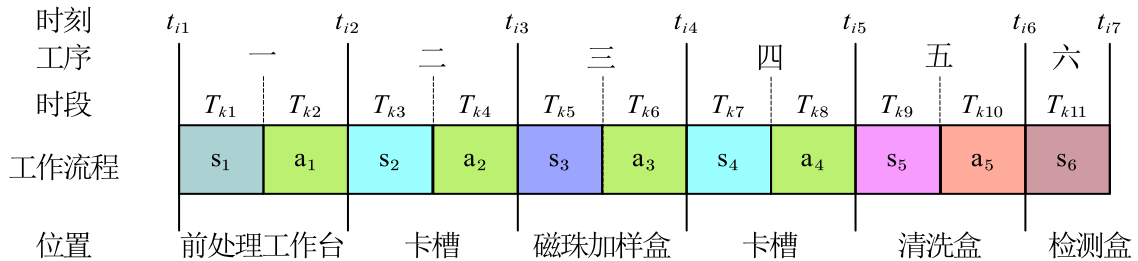


图2 第 i 个芯片的工作流程及对应的时刻和位置

在问题一中，投放芯片种类单一，芯片进仓的顺序与检测结束的顺序是一致的，因此，在平台数量单个的情况下，将会产生相邻芯片投放时间间隔的约束；在平台数量多个的情况下，将会产生间隔数量芯片投放时间间隔的约束；在不连续使用单一平台的情况下，将会产生对前面所有芯片投放时间间隔的约束。具体分析如下：

首先考虑平台数量单个的情况。由于每台机器在同一时刻只能加工一个工件，要保证不发生空间冲突，就必须要求相邻芯片传输的间隔时间不小于单个平台的最大处理时间；同时，尽管进行 a_2 、 a_4 、 a_5 三个动作时，芯片所在位置对应的平台数量不只有一个，但是执行动作所需要的转盘每次只能对位一个芯片，因此也需要将这三个动转盘对位的时间考虑到相邻芯片传输的间隔时间中。结合上述分析，令

$$M = \max\{\frac{T_1}{5} + T_2, T_4, T_5 + T_6, T_8, T_{10}, T_{11}\} \quad (2)$$

可获取如下数学表达式：

$$t_{i,1} - t_{i-1,1} \geq M \quad (3)$$

然后考虑平台数量多个的情况。此时每台机器在同一时刻可以加工多个芯片，且由于种类单一，不存在前 $i-1$ 个芯片的检测程序落后于第 i 个芯片的情况。因此，为了保证卡槽使用不产生冲突，只需要保证第 i 个芯片第一次温育进入卡槽的时间 ($t_{i,2}$) 大于等于第 $i-40$ 个第二次温育结束后离开卡槽的时间 ($t_{i-40,5}$)，数学语言描述如下：对时间间隔的约束可以扩展到多个芯片间。

$$t_{i,2} \geq t_{i-40,5} \quad (4)$$

为了保证清洗盘使用不产生冲突，应保证第 i 个芯片进入清洗盘的时间大于等于第 $i-8$ 个芯片离开清洗盘的时间，数学语言描述如下：

$$t_{i,5} \geq t_{i-8,6} + T_9 \quad (5)$$

最后考虑不连续使用同一单个平台的情况。不连续使用单一平台包括两种情况，一是一个芯片在一次检测流程中不连续使用同一单个平台，二是多类芯片由于时间处理差异导致前后芯片不能前后连续使用同一单个平台的情况，这两种情况都对该平台的使用造成了约束。由于问题一中的芯片种类一致，不需要考虑第二种情况。考虑第一种情况时，由图 2 可知，由于 $a_1 \sim a_4$ 动作均使用到转盘，而转盘每次只能对位一个芯片，因此在设备运行过程中，不同芯片使用转盘的时间不能产生冲突。第 i 个芯片使用转盘的时间区间为：

$$TS_i = [t_{i1} + T_1, t_{i2}] \cup [t_{i2} + T_3, t_{i3}] \cup [t_{i3} + T_5, t_{i4}] \cup [t_{i4} + T_7, t_{i5}] \quad (6)$$

为保证转盘使用不产生冲突，应保证第 i 个芯片使用转盘的时间区间与前 $i-1$ 个芯片使用转盘的时间区间交集的并集为空集，数学语言描述如下：

$$(TS_{i-1} \cup TS_{i-2} \cup \dots \cup TS_1) \cap TS_i = \emptyset \quad (7)$$

4.2 模型建立

结合问题分析，问题一的芯片类型单一，各芯片处理流程对应的时间是一致的。最小化运行总时间等价于逐个最小化芯片投放的时间。考虑上文分析的约束条件，在优化第 i 个芯片的投放时间时，问题一的数学模型可写为：

$$\begin{aligned}
& \min_{t_{i,1}} t_{i,1} \\
& s.t. \begin{cases} t_{i,1} - t_{i-1,1} \geq M \\ t_{i,2} \geq t_{i-40,5} \\ t_{i,5} \geq t_{i-8,6} + T_9 \\ (TS_{i-1} \cup TS_{i-2} \cup \dots \cup TS_1) \cap TS_i = \emptyset \end{cases}
\end{aligned} \tag{8}$$

根据题目背景及所给数据可知, M 值的大小为 36.

4.3 算法分析求解

4.3.1 算法分析

在模型一中, 第 i 个芯片可投放的时间点具备一个可行区间, 求解的目标就是寻找到可行集中的最小值. 本文采取搜索求解的方式, 在给定第 i 个芯片投放时间目标初始值的情况下, 逐步增加搜索大小, 当搜索到满足所有约束条件的最小解时, 对第 i 个芯片投放时间的搜索停止. 同时, 在问题背景下, 芯片在磁珠加样盒中加样时间不定, 为了最小化总检测时间, 我们假设每组检测芯片有 200 个, 遍历磁珠加样消耗时间可行区间, 最终得到使总检测时间最小的磁珠加样时间值. 具体算法流程如下图 3, 算法的伪代码如表 3.

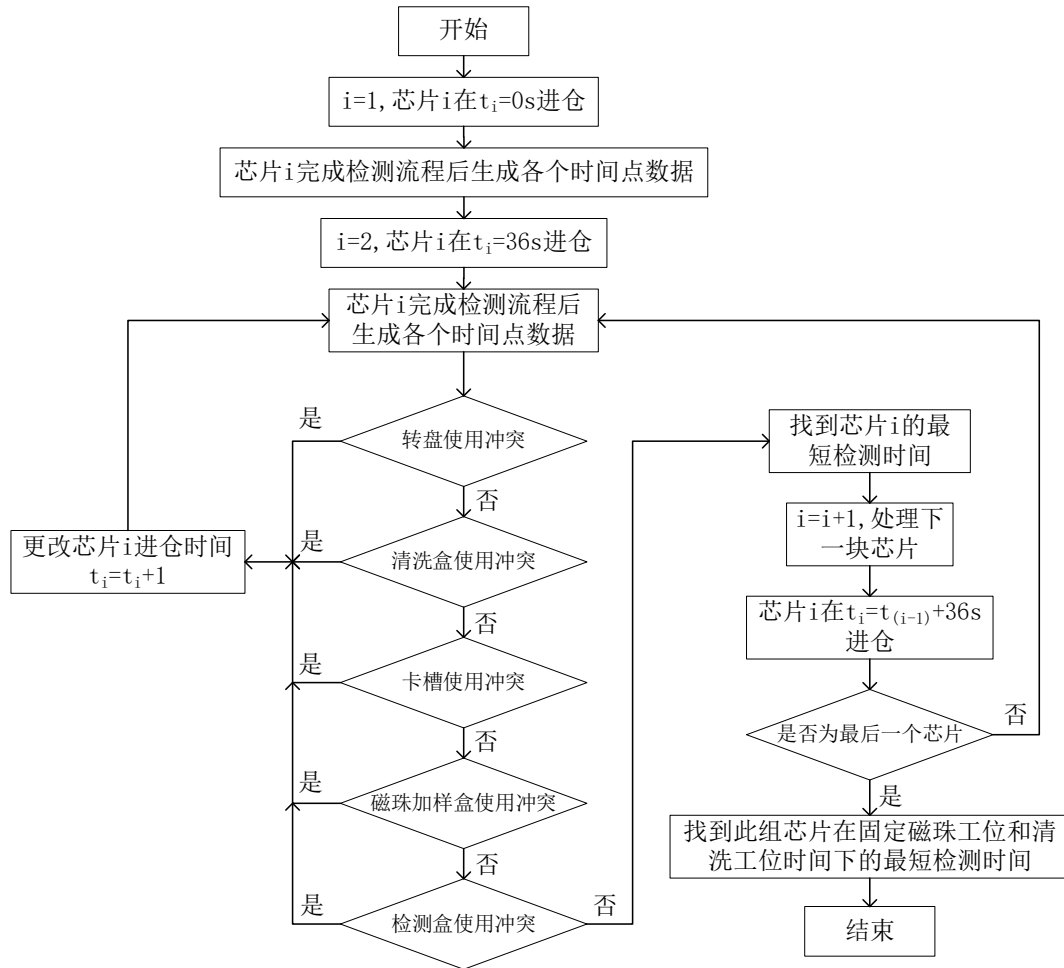


图3 算法一流程图

表3 算法一的伪代码

算法一：面向单类型芯片 A 或 B 的算法

Input: 待检测芯片总数、第一次温育耗时、磁珠加样耗时、第二次温育耗时、清洗耗时

Output: 时间矩阵、所有芯片完成的总时间、单位时间可检测的芯片数量

```

01. 定义初始的时间矩阵 Matrix_time
02. 令  $t_{i1} = 0$ 
03. for j = 2:待检测芯片总数
04.      $t_{i1} = t_{i1} + 36$ 
05.     while 1
06.         根据  $t_{i1}$  计算待检测芯片对应的时间数组 Time(时间节点同 Matrix_time)
07.         if 转盘、磁珠工位、检测工位、温育盘数量、清洗工位数量不冲突
08.             在当前  $t_{i1}$  下可以加入新的芯片进行检测
09.             将 Time 数组的值放入 Matrix_time 矩阵的下一行
10.             break
11.         else
12.              $t_{i1} = t_{i1} + 1$ 
13.         end
14.     end
15. end
16. 根据 max(Matrix_time) 计算得到所有芯片检测完成的时间
17. 计算单位时间内可检测的芯片数量
18. 遍历所有磁珠加样耗时的可能取值, 重复算法流程 1-17, 得到最短的检测时间
19. 输出最佳结果

```

4.3.2 问题求解

针对问题一中芯片 A、B 的温育时间背景, 结合编程工具, 可以获取得到 200 个检测芯片在每道工序的开始时间和结束时间. 表 4 展示了遍历磁珠工位时间区间下芯片 A 及芯片 B 各 200 个总的消耗时间图, 从中选取消耗最少的对应时间作为各自磁珠加样时间的输入. 由表 4 可知, 在进行芯片 A 和芯片 B 的加工时, 我们分别取磁珠加样时间为 23s, 25s. 表 5 和表 6 分别展示了部分芯片 A 和芯片 B 的加工数据, 具体数据见附录 A 中附表 1 及附表 2.

表4 不同磁珠加样时间取值下芯片 A、B 总的时间消耗表

磁珠加样时间/s	21	22	23	24	25
芯片 A 总消耗时间/s	13669	11901	11687	12455	12421
芯片 B 总消耗时间/s	21549	20623	20658	21005	20532

表5 A 型芯片的调度结果部分示意

芯片序号	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	0	156	764	793	1109	1446	1471
2	37	193	801	832	1148	1485	1510
3	76	232	840	871	1187	1524	1549
...	...	...	...	...	...	...	...
46	2225	2381	2989	3020	3336	3673	3698
47	2320	2476	3084	3115	3431	3768	3793
...	...	...	...	...	...	...	...
199	10175	10331	10939	10970	11286	11623	11648
200	10214	10370	10978	11009	11325	11662	11687

表6 B 型芯片的调度结果部分示意

芯片序号	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	0	156	2564	2593	3809	4146	4171
2	37	193	2601	2632	3848	4185	4210
3	76	232	2640	2671	3887	4224	4249
...	...	...	...	...	...	...	...
198	16407	16563	18971	19002	20218	20555	20580
199	16446	16602	19010	19041	20257	20594	20619
200	16485	16641	19049	19080	20296	20633	20658

由于题目所要求解的是单位时间（一小时内）最多可以完成的芯片检测的数量。由题设背景易知，刚开始投入的芯片在小时间间隔上具有优势，因为此时设备的空间资源是最丰富的。据此思想，我们将第一片芯片结束检测流程的时间作为起始时间点，计算200个芯片全部完成后，单位时间内检测完成的芯片数量。

芯片A的第一片芯片在1471s时检测完毕，以该时间作为初始点，200片芯片检测完成总耗时10216s，计算得到芯片A在单位时间内最多可以完成**70.48**片芯片的检测。

芯片B的第一片芯片在4171s时检测完毕，以该时间作为初始点，200片芯片检测完成总耗时16361s，计算得到芯片B在单位时间内最多可以完成**44.01**片芯片的检测。

5 问题二：模型建立与求解

5.1 约束条件分析

问题二与问题一约束条件存在差异。造成的差异主要体现在平台数量多个及不连续使用同一平台的条件下。当连续投放的芯片种类不同时，由于温育时间的差异，芯片投放的顺序不一定为芯片检测结束的顺序，可能存在后进入检测流程的芯片工作流程超前

于先进入检测流程芯片的情况。此时通过比较区间端点值大小来体现平台数量多个约束的方案将不再可行。同时，由问题一对于不连续使用同一单个平台的分析可知，问题二还需要考虑前后芯片不能前后连续使用同一单个平台的情况。综合上述分析，问题二对于约束条件的分析如下。

首先考虑平台数量单个的情况。该情况与问题一中平台数量单个的情况相同，参考问题一。

然后考虑平台数量多个的情况。不同于问题一中在平台数量多个情况下对约束条件的处理，本文需要通过比较时间区间的方式来确定是否产生冲突。比较当前芯片在此步骤的时间区间和前面所有芯片在此步骤的时间区间是否存在交集，通过交集的个数来统计同一时刻对该平台的使用个数。为了保证检测不产生冲突，应满足同一时刻的交集个数不大于该平台的个数。

为了保证卡槽使用不产生冲突，在同一时刻，前 i 个芯片中处于卡槽的个数应小于等于 40 个。第 i 个芯片处于卡槽的时间段为：

$$TW_i = [t_{i2}, t_{i3}] \cup [t_{i4}, t_{i5}] \quad (9)$$

将 TW_i 时间段按整数采样为矩阵 k_i ，取前 i 个芯片所产生的时间采样矩阵拼接为矩阵 K ：

$$K = \text{cat}(k_1, k_2, \dots, k_i) \quad (10)$$

其中， $\text{cat}()$ 函数表示矩阵拼接。

统计矩阵 K 中元素的众数，众数体现了卡槽同时温育芯片的最大个数，为了保证卡槽使用不产生冲突，应保证众数的大小小于等于 40，即：

$$\text{mode}(K) \leq 40 \quad (11)$$

其中， $\text{mode}()$ 函数表示取众数。

为了保证清洗盘使用不产生冲突，在同一时刻，前 i 个芯片中处于清洗盒的个数应小于等于 8 个。第 i 个芯片处于清洗盒的时间段为：

$$TQ_i = [t_{i5}, t_{i6}] \quad (12)$$

将 TQ_i 时间段按整数采样为矩阵 q_i ，取前 i 个芯片所产生的时间采样矩阵拼接为矩阵 Q ：

$$Q = \text{cat}(q_1, q_2, \dots, q_i) \quad (13)$$

统计矩阵 Q 中元素的众数，众数体现了清洗盒同时清洗芯片的最大个数，为了保证清洗盒使用不产生冲突，应保证众数的大小小于等于 8，即：

$$\text{mode}(Q) \leq 8 \quad (14)$$

最后考虑不连续使用同一单个平台的情况。情况一与问题一一致。问题二需要考虑情况二，即多类芯片由于时间处理差异导致前后芯片不能前后连续使用同一单个平台的情况，4.1 图 2 所知，符合条件的平台有磁珠加样盒与检测盒。在设备运行中，不同时间投递的不同种类芯片在磁珠加样盒与检测盒的时间不能产生冲突。第 i 个芯片使用磁珠加样盒的时间区间为：

$$TZ_i = [t_{i3}, t_{i4}] \quad (15)$$

为保证磁珠加样盒使用不产生冲突，应保证第 i 个芯片使用磁珠加样盒的时间区间与前 $i-1$ 个芯片使用磁珠加样盒的时间区间交集的并集为空集，数学语言描述如下：

$$(TZ_{i-1} \cup TZ_{i-2} \cup \dots \cup TZ_1) \cap TZ_i = \emptyset \quad (16)$$

第 i 个芯片使用检测盒的时间区间为:

$$TC_i = [t_{i6}, t_{i7}] \quad (17)$$

为保证检测盒使用不产生冲突, 应保证第 i 个芯片使用检测盒的时间区间与前 $i-1$ 个芯片使用检测盒的时间区间交集的并集为空集, 数学语言描述如下:

$$(TC_{i-1} \cup TC_{i-2} \cup \dots \cup TC_1) \cap TC_i = \emptyset \quad (18)$$

5.2 模型建立

结合问题分析, 问题二具有两种类型的芯片, 相邻两芯片处理流程对应的时间可能是不一致的. 本文基于贪心算法的思想, 在调度过程的每一步都选取使得当下总完成时间最小的方案, 该思路也可转化为使单位时间进入检测流程的芯片数量尽可能多, 即最小化第 i 个芯片进入的时间. 考虑上文分析的约束条件, 在优化第 i 个芯片的投放时间时, 问题二的数学模型可写为:

$$\begin{aligned} & \min_{t_{i,1}} t_{i,1} \\ \text{s.t.} \quad & \begin{cases} t_{i,1} - t_{i-1,1} \geq M \\ \text{mode}(K) \leq 40 \\ \text{mode}(Q) \leq 8 \\ (TS_{i-1} \cup TS_{i-2} \cup \dots \cup TS_1) \cap TS_i = \emptyset \\ (TZ_{i-1} \cup TZ_{i-2} \cup \dots \cup TZ_1) \cap TZ_i = \emptyset \\ (TC_{i-1} \cup TC_{i-2} \cup \dots \cup TC_1) \cap TC_i = \emptyset \end{cases} \end{aligned} \quad (19)$$

5.3 算法分析与求解

5.3.1 算法分析

算法二与算法一的求解思路一致, 即通过时间序贯的方式逐个搜索求解, 确定投放顺序. 由于芯片的数量有限, 算法需要分情况考虑投放顺序. 设置 k_1 和 k_2 两个变量, 初始值设置为 0, 每当一个 A 芯片进入检测流程, 变量 k_1 加一, 每当一个 B 芯片进入检测流程, 变量 k_2 加一, 考虑以下 4 种情况:

(1) $k_1 < 80, k_2 < 80$: 芯片 A 和 B 均未全部进入检测流程, 分别计算第 i 个芯片为芯片 A 或芯片 B 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 则优先选择单个流程总完成时间最小的芯片, 本题背景下, A 的单个流程总完成时间最小, 这种情况下优先芯片 A.

(2) $k_1 < 80, k_2 = 80$: 芯片 B 全部完成检测, 选择芯片 A 进入检测流程.

(3) $k_1 = 80, k_2 < 80$: 芯片 A 全部完成检测, 选择芯片 B 进入检测流程.

(4) $k_1 = 80, k_2 = 80$: 完成检测任务, 算法结束.

具体的算法流程如下图 4 所示, 算法二的伪代码如下表 7 所示.

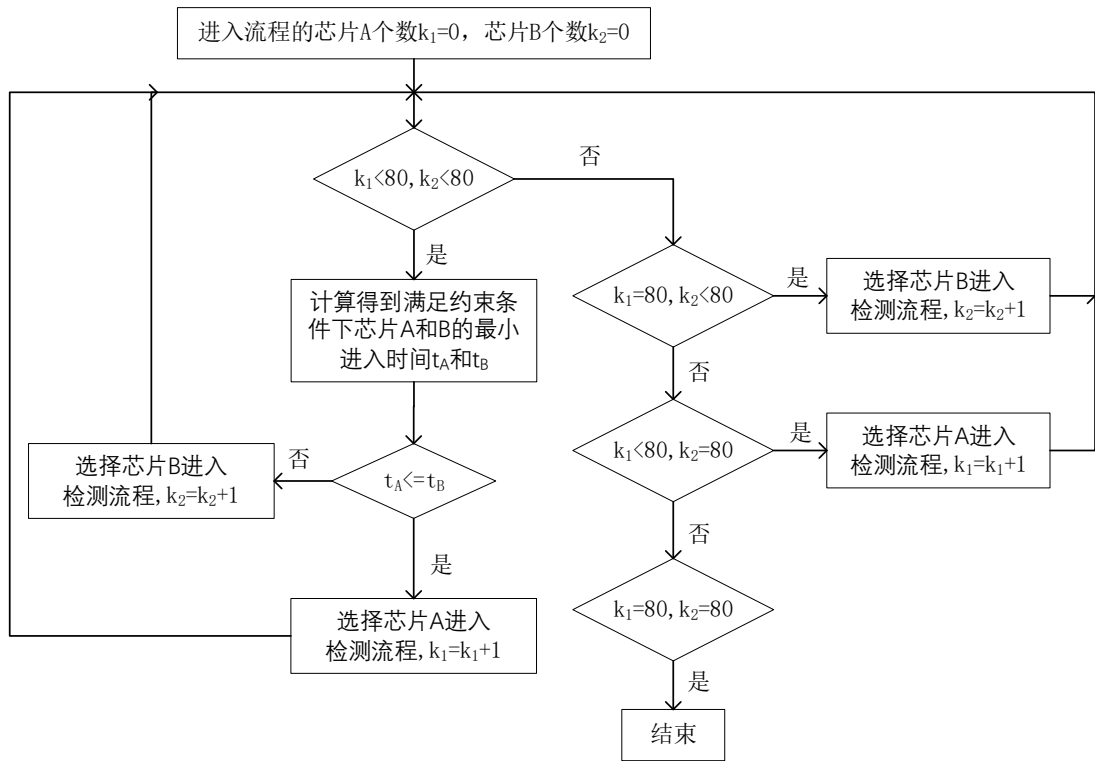


图4 算法二流程图

表7 算法二的伪代码

算法二：面向两种类型芯片 A 和 B 的算法

Input: 芯片 A 数量 80、芯片 B 数量 80、芯片 A 第一次温育耗时、芯片 A 第二次温育耗时、芯片 B 第一次温育耗时、芯片 B 第二次温育耗时、磁珠加样耗时、清洗耗时

Output: 时间矩阵 Matrix_time、芯片类型的加入顺序、所有芯片完成的总时间

01. 令第一个检测的芯片为 A, 定义已检测完成芯片数量 $k_1 = 1, k_2 = 0$

02. for $j = 2:160$

03. $t_{i1} = t_{i1} + 36$

04. if $k_1 < 80$ and $k_2 < 80$

05. while 1

06. 根据 t_{i1} 计算待检测 A 芯片对应的时间数组 TimeA

07. 根据 t_{i1} 计算待检测 B 芯片对应的时间数组 TimeB

08. if TimeA 满足五个约束条件不冲突

09. 将 TimeA 数组的值放入 Matrix_time 矩阵的下一行

10. $k_1 = k_1 + 1$

11. break

12. end

13. if TimeB 满足五个约束条件不冲突

```

14.          将 TimeB 数组的值放入 Matrix_time 矩阵的下一行
15.           $k_2 = k_2 + 1$ 
16.          break
17.      else  $t_{i1} = t_{i1} + 1$ 
18.      end
19.  end
20.  elseif  $k_1 == 80$  and  $k_2 < 80$ 
21.      芯片 A 全部检测完成, 按算法一检测 B,  $k_2 = k_2 + 1$ 
22.  elseif  $k_1 < 80$  and  $k_2 == 80$ 
23.      芯片 B 全部检测完成, 按算法一检测 A,  $k_1 = k_1 + 1$ 
24.  else 全部检测完成, 结束
25. end
26. 根据  $\max(\text{Matrix\_time})$  计算得到所有芯片检测完成的时间  $t_A$ 
27. 令第一个检测的芯片为 B, 定义已检测完成芯片数量  $k_1 = 0, k_2 = 1$ 
28. 重复算法流程 2-26, 得到所有芯片检测完成的时间  $t_B$ 
29. 计算最短检测时间  $\min(t_A, t_B)$ 
30. 遍历所有磁珠加样耗时的可能取值, 重复算法流程 1-29, 得到最短的检测时间

```

5.3.2 问题求解

针对问题一中芯片 A、B 的温育时间背景, 结合编程工具, 可以获取得到 160 个检测芯片的传递顺序、每道工序的开始时间和结束时间. 为了验证算法的性能, 本文做了四组实验来进行对比分析. 分别为先检测 A 型芯片后检测 B 型芯片、先检测 B 型芯片后检测 A 型芯片、按本文算法先加入 A 型芯片检测和按本文算法先加入 B 型芯片检测. 表 8 为不同检测方案中不同磁珠加样时间下总时间消耗对比表, 从表上可以看出, 磁珠加样时间为 22 秒和 23 秒时按芯片种类顺序进入检测流程的实验总耗时最少, 磁珠加样时间为 25 秒和 23 秒时按本文算法加入芯片的两个实验总耗时最少, 但是本文提出的算法的最小总耗时均低于顺序放入芯片的总耗时. 其中, 按本文算法先加入 B 型芯片的总耗时最少, 为 13385 秒, 优于顺序放入芯片 400 秒左右. 表 8 图 5 为不同检测方案下总的时间消耗曲线图, 从图中也可以看出, 按本文算法进行检测总耗时均优于顺序放入芯片, 其中按本文算法先加入芯片 B 效果最好, 在磁珠加样时间小于 25 秒时能达到总耗时最小. 所以当 A、B 两种类型的芯片各 80 片时, 全部检测完至少需要 13385s. 采取此种措施下仪器的工作数据见附录 A 附表 3, 部分调度数据见表 9.

表8 不同检测方案中不同磁珠加样时间下总时间消耗对比表

磁珠加样时间/s		21	22	23	24	25
先检测 A 后检测 B 总耗时/s		14713	13725	13786	13793	14076
先检测 B 后检测 A 总耗时/s		15386	14163	14114	14498	14365
按本文算法检测总耗时/s	先加入芯片 A	14257	13812	13644	13690	13448
	先加入芯片 B	14173	13504	13385	13406	13545

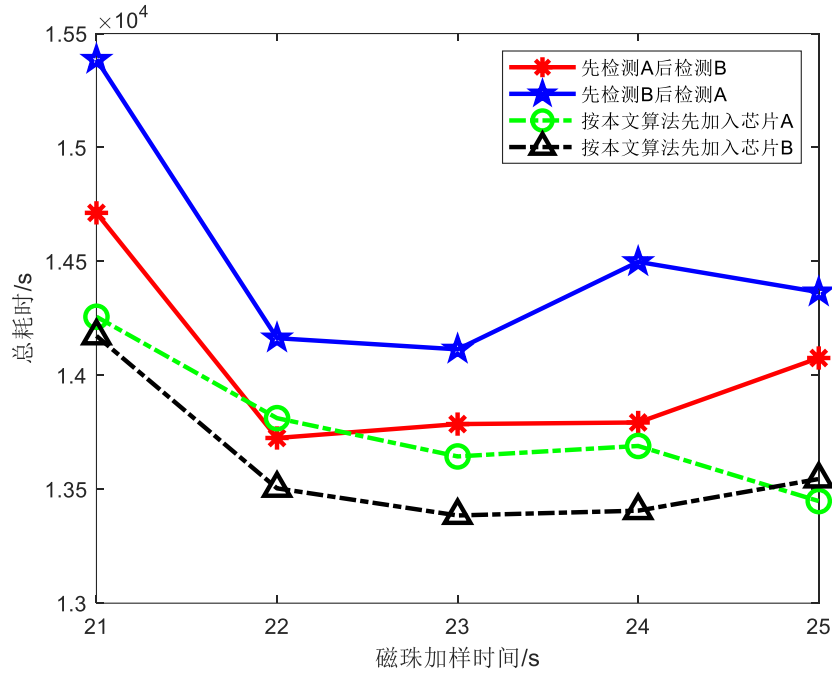


图5 不同检测方案下总的时间消耗曲线图

表9 A、B型芯片的调度结果部分示意

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	B	0	156	2564	2593	3809	4146	4171
2	A	36	192	800	831	1147	1484	1509
3	B	72	228	2636	2667	3883	4220	4245
4	A	108	264	872	903	1219	1556	1581
5	B	144	300	2708	2739	3955	4292	4317

6 问题三：算法分析与求解

问题三在问题二的基础上增加了一种芯片种类, 由前文分析可知, 问题三同样适用问题二的思路, 控制三种芯片进入检测流程的顺序, 求解单次迭代下满足约束条件下的最小投送时间. 此时, 运用于问题二的约束条件及优化目标同样适用于问题三, 两者只是存在算法求解过程的区别. 重复的地方此节将不再赘述.

6.1 算法分析

由于有三种不同的芯片, 故设置 k_1 、 k_2 和 k_3 三个变量来对进入检测流程的芯片计数, 每当一个 A 芯片进入检测流程, 变量 k_1 加 1, 每当一个 B 芯片进入检测流程, 变量 k_2 加 1, 每当一个 C 芯片进入检测流程, 变量 k_3 加一. 此时存在以下 8 种情况:

(1) $k_1 < 60, k_2 < 50, k_3 < 50$: 芯片 A、B 和 C 均未全部进入检测流程, 分别计算第 i 个芯片为芯片 A、芯片 B 及芯片 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先级: $A > C > B$.

(2) $k_1 = 60, k_2 < 50, k_3 < 50$: 芯片 A 全部完成检测, 分别计算第 i 个芯片为芯片 B 或芯片 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择芯片 C.

(3) $k_1 < 60, k_2 = 50, k_3 < 50$: 芯片 B 全部完成检测, 分别计算第 i 个芯片为芯片 A 或芯片 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择芯片 A.

(4) $k_1 < 60, k_2 < 50, k_3 = 50$: 芯片 C 全部完成检测, 分别计算第 i 个芯片为芯片 A 或芯片 B 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择芯片 A.

(5) $k_1 = 60, k_2 = 50, k_3 < 50$: 芯片 A 和 B 全部完成检测, 选择芯片 C 进入检测流程.

(6) $k_1 = 60, k_2 < 50, k_3 = 50$: 芯片 A 和 C 全部完成检测, 选择芯片 B 进入检测流程.

(7) $k_1 < 60, k_2 = 50, k_3 = 50$: 芯片 B 和 C 全部完成检测, 选择芯片 A 进入检测流程.

(8) $k_1 = 60, k_2 = 50, k_3 = 50$: 完成检测任务, 算法结束.

具体的算法流程如下图 6 所示, 算法三的伪代码如表 10.

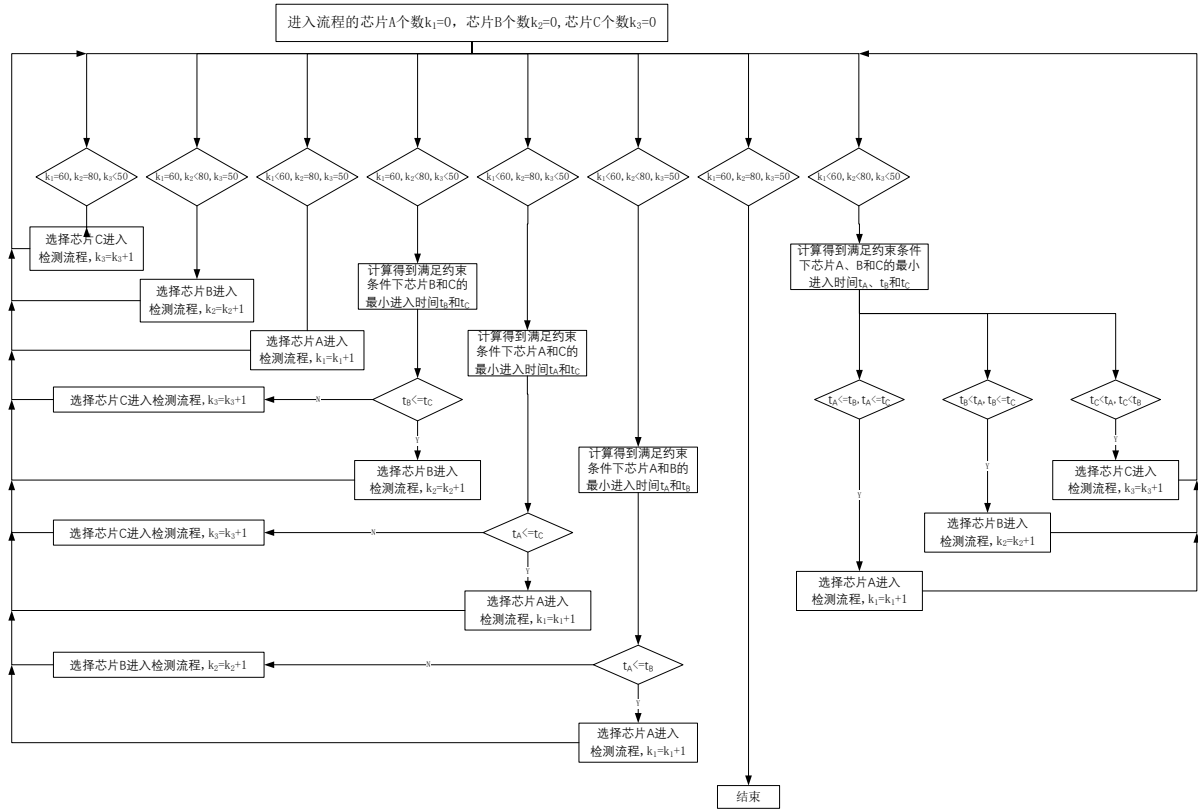


图6 算法三流程图

表10 算法三的伪代码

算法三: 面向三种类型芯片 A、B 和 C 的算法

Input: 芯片 A 数量 60、芯片 B 数量 50、芯片 C 数量 50、芯片 A 第一次温育耗时、芯片 A 第二次温育耗时、芯片 B 第一次温育耗时、芯片 B 第二次温育耗时、芯片 C 第一次温育耗时、芯片 C 第二次温育耗时、磁珠加样耗时、清洗耗时

Output: 时间矩阵 `Matrix_time`、芯片类型的加入顺序、所有芯片完成的总时间

01. 令第一个检测的芯片为 A, 定义已检测完成芯片数量 $k_1 = 1, k_2 = 0, k_3 = 0$

02. for $j = 2:160$

03. $t_{i1} = t_{i1} + 36$

04. if $k_1 < 60$ and $k_2 < 50$ and $k_3 < 50$

05. while 1

06. if 芯片 A 满足五个约束条件

07. 添加芯片 A, $k_1 = k_1 + 1$, break

08. end

09. if 芯片 C 满足五个约束条件

10. 添加芯片 C, $k_3 = k_3 + 1$, break

11. end

12. if 芯片 B 满足五个约束条件

13. 添加芯片 B, $k_2 = k_2 + 1$, break

14. else $t_{i1} = t_{i1} + 1$

15. end

16. elseif $k_1 == 60$ and $k_2 < 50$ and $k_3 < 50$

17. 芯片 A 全部检测完成, 按算法二决策检测 B 或 C

18. $k_2 = k_2 + 1$ or $k_3 = k_3 + 1$

19. elseif $k_1 < 60$ and $k_2 == 50$ and $k_3 < 50$

20. 芯片 B 全部检测完成, 按算法二决策检测 A 或 C

21. $k_1 = k_1 + 1$ or $k_3 = k_3 + 1$

22. elseif $k_1 < 60$ and $k_2 < 50$ and $k_3 == 50$

23. 芯片 C 全部检测完成, 按算法二决策检测 A 或 B

24. $k_1 = k_1 + 1$ or $k_2 = k_2 + 1$

25. elseif $k_1 == 60$ and $k_2 == 50$ and $k_3 < 50$

26. 芯片 A、B 全部检测完成, 按算法一检测 C, $k_3 = k_3 + 1$

27. elseif $k_1 == 60$ and $k_2 < 50$ and $k_3 == 50$

28. 芯片 A、C 全部检测完成, 按算法一检测 B, $k_2 = k_2 + 1$

29. elseif $k_1 < 60$ and $k_2 == 50$ and $k_3 == 50$

30. 芯片 B、C 全部检测完成, 按算法一检测 A, $k_1 = k_1 + 1$

31. else 全部检测完成

32. end

-
33. end
 34. 根据 $\max(\text{Matrix_time})$ 计算得到所有芯片检测完成的时间 t_A
 35. 令第一个检测的芯片为 B, 定义已检测完成芯片数量 $k_1 = 0, k_2 = 1, k_3 = 0$
 36. 重复算法流程 2-34, 得到所有芯片检测完成的时间 t_B
 37. 令第一个检测的芯片为 C, 定义已检测完成芯片数量 $k_1 = 0, k_2 = 0, k_3 = 1$
 38. 重复算法流程 2-34, 得到所有芯片检测完成的时间 t_C
 39. 计算最短检测时间 $\min(t_A, t_B, t_C)$
 40. 遍历所有磁珠加样耗时的可能取值, 重复算法流程 1-39, 得到最短的检测时间
-

6.2 问题求解

针对芯片 A、B、C 的温育时间背景, 结合编程工具, 可以获取得到 160 个检测芯片的传递顺序、每道工序的开始时间和结束时间. 为了验证算法的性能, 本文做了九组实验来进行对比分析. 问题三所做的对比实验主要包含两大类, 一类为按芯片种类顺序放入芯片, 一种芯片检测完毕后下一种芯片进入检测流程, 三种芯片有 6 种可能的情况, 检测顺序分别为 ABC, ACB, BAC, BCA, CAB 和 CBA. 第二类为使用本文建立的算法, 改变第一次进入检测流程的芯片的类型, 一共有三种情况. 针对以上 9 种情况, 本文改变磁珠加样时间, 计算每种情况下完成检测任务的总耗时, 实验结果如表 11 所示. 从实验结果可以看出, 按本文算法检测总耗时低于绝大部分顺序放入芯片检测总耗时. 按本文算法检测的总耗时在磁珠加样时间为 22 秒时达到最小, 且先加入芯片 B 的总耗时最小, 为 13124 秒, 优于其他所有情况. 表 11 图 7 为不同检测方案下总的时间消耗对比曲线图, 其中直线表示第一大类, 虚线表示第二大类, 可以看出第二大类, 即本文算法性能较好, 能大幅降低检测的总耗时. 所以当 A 型芯片 60 片、B 型芯片 50 片、C 型芯片 50 片时, 全部检测完至少需要 13124 秒. 此种工作状态下设备的调度表见附录 A 附表 4, 部分调度数据见表 12.

表11 不同检测方案下总的消耗时间对比表

磁珠加样时间/s		21	22	23	24	25
检测顺序 A、B、C 总耗时/s		14607	13913	13535	13741	14375
检测顺序 A、C、B 总耗时/s		14857	14263	14318	14167	14311
检测顺序 B、A、C 总耗时/s		14993	14429	14555	14624	14722
检测顺序 B、C、A 总耗时/s		14707	14132	14273	13886	14324
检测顺序 C、A、B 总耗时/s		16654	15675	15438	15402	15812
检测顺序 C、B、A 总耗时/s		14444	13744	13873	14017	14162
按本文算法检测总耗时/s	先加入芯片 A	13243	13138	13373	13389	13484
	先加入芯片 B	13340	13124	13564	13285	13294
	先加入芯片 C	13173	13490	13280	13422	13667

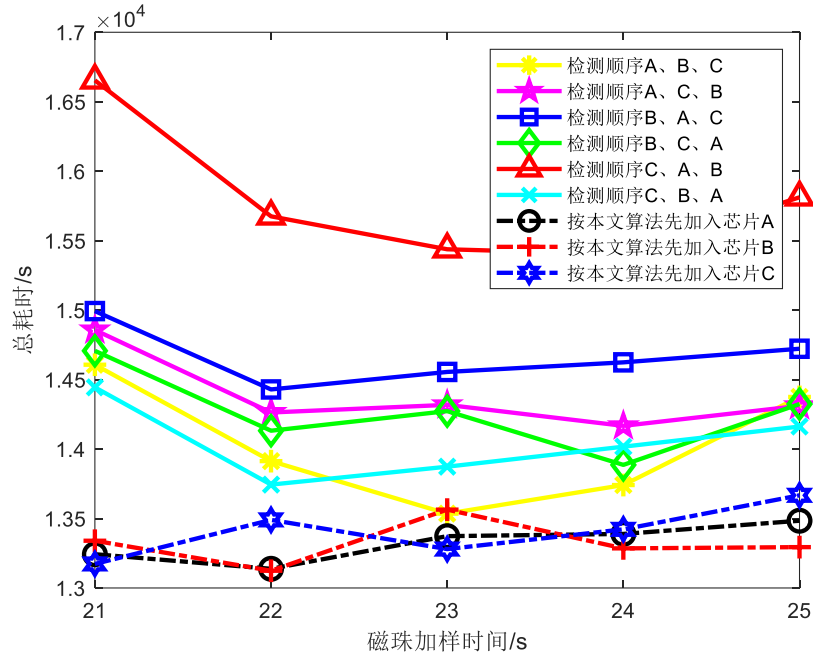


图7 不同检测方案下总的消耗时间对比图

表12 A、B、C型芯片的调度结果部分示意 (单位: s)

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	B	0	156	2564	2593	3809	4146	4171
2	A	36	192	800	830	1146	1483	1508
3	C	72	228	1736	1766	2382	2719	2744
4	A	108	264	872	902	1218	1555	1580
5	C	144	300	1808	1838	2454	2791	2816
6	A	180	336	944	974	1290	1627	1652
7	C	216	372	1880	1910	2526	2863	2888
8	A	252	408	1016	1046	1362	1699	1724
9	B	288	444	2852	2882	4098	4435	4460
10	A	324	480	1088	1118	1434	1771	1796

7 问题四：推广分析与举例

7.1 问题二的一般化推广

7.1.1 对问题二推广的分析

首先分析问题二的约束条件是否可推广。问题二的约束条件如公式 (19) 所示。由约束条件可以看出，约束条件的成立与芯片数量无关。此外，虽然约束条件的取值与温育时间有关，但是改变温育时间并不会引起约束条件之间的矛盾，这是因为约束条件之间是相互独立的。因此问题二中约束条件可推广。

其次分析两次温育时间是否可推广。于本文算法而言，两次温育时间的一般化会影响约束条件的构成，但是温育时间的一般化并不会使得约束条件矛盾。除此之外，在算

法决策方面，相同条件下算法会优先选择耗时较短的芯片进入，该决策环节可以用温育时间简单表示。因此温育时间可以推广。

最后分析芯片数量是否可推广。算法求解过程中涉及到 4 种情况：AB 皆未完成检测、A 全部完成 B 还未完成、B 全部完成 A 还未完成、AB 都全部完成检测。这四种情况的判定准则可以由芯片数量表示。所以芯片数量可推广。

7.1.2 问题二算法的推广

设 A、B 型芯片的数量分别为 m_1 、 m_2 ，A 型芯片的两次温育时间分别为 a_1 、 b_1 分钟，B 型芯片的两次温育时间分别为 a_2 、 b_2 分钟。将其推广到 A、B 型芯片数量、两次温育时间皆可变的场景。

问题二仍可使用算法二求解，只需要对其中的部分参数进行抽象化。将算法求解可能面临的过程抽象为以下四种情况：

- (1) $k_1 < m_1, k_2 < m_2$ ：芯片 A 和 B 均未全部进入检测流程，分别计算第 i 个芯片为 A 或 B 时满足约束条件的最小投放时间，选择投放时间最小的芯片进入当前检测流程。若投放时间一致，优先选择 $a_1 + b_1$ 和 $a_2 + b_2$ 中较小值对应的芯片。
- (2) $k_1 < m_1, k_2 = m_2$ ：芯片 B 全部完成检测，选择芯片 A 进入检测流程。
- (3) $k_1 = m_1, k_2 < m_2$ ：芯片 A 全部完成检测，选择芯片 B 进入检测流程。
- (4) $k_1 = m_1, k_2 = m_2$ ：完成检测任务，算法结束。

7.1.3 举例验证

当芯片种类为两种时，表 13 和表 14 分别为固定温育时间改变芯片数量和固定芯片数量改变温育时间的实验结果。其中，芯片数量和温育时间都按倍数上升，由表可以看出完成检测的总耗时随着芯片数量和温育时间的增加而上升。

表13 两种芯片时固定温育时间、改变芯片数量

芯片 1			芯片 2			总耗时/s
数量 /片	第一次温育时间/min	第二次温育时间/min	数量 /片	第一次温育时间/min	第二次温育时间/min	
3	4	13	13	11	2	2142
6	4	13	26	11	2	3354
9	4	13	39	11	2	4282
12	4	13	52	11	2	5024
15	4	13	65	11	2	6108

表14 两种芯片时固定芯片数量、改变温育时间

芯片 1			芯片 2			总耗时/s
数量 /片	第一次温育时间/min	第二次温育时间/min	数量 /片	第一次温育时间/min	第二次温育时间/min	
3	4	13	13	11	2	2142
3	8	26	13	22	4	2914
3	12	39	13	33	6	3512
3	16	52	13	44	8	4306
3	20	65	13	55	10	5063

7.2 问题三的一般化推广

7.2.1 对问题三推广的分析

问题三只在问题二的基础上增加了一种芯片类型, 所以算法的处理思路相近.

问题三的约束条件和两次温育时间的可推广性同问题二. 而芯片数量的可推广性则有问题二的四种情况变为八种, 但是这八种情况的判定准则可以分别由芯片数量表示, 所以芯片数量可推广.

7.2.2 问题三算法的推广

设 A、B、C 型芯片的数量分别为 m_1 、 m_2 、 m_3 , A 型芯片两次温育时间为 a_1 、 b_1 分钟, B 型芯片两次温育时间为 a_2 、 b_2 分钟, C 型芯片两次温育时间为 a_3 、 b_3 分钟. 将其推广到 A、B、C 型芯片数量、两次温育时间皆可变的场景.

问题三仍可使用算法三求解, 只需要对其中的部分参数进行抽象化. 将算法求解可能面临的过程抽象为以下八种情况:

(1) $k_1 < m_1, k_2 < m_2, k_3 < m_3$: 芯片 A、B 和 C 均未全部进入检测流程, 分别计算第 i 个芯片为 A、B 及 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先级按照 $a_1 + b_1$ 、 $a_2 + b_2$ 和 $a_3 + b_3$ 的大小顺序从小到大排序.

(2) $k_1 = m_1, k_2 < m_2, k_3 < m_3$: 芯片 A 全部完成检测, 分别计算第 i 个芯片为 B 或 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择 $a_2 + b_2$ 和 $a_3 + b_3$ 中较小的值对应的芯片.

(3) $k_1 < m_1, k_2 = m_2, k_3 < m_3$: 芯片 B 全部完成检测, 分别计算第 i 个芯片为 A 或 C 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择 $a_1 + b_1$ 和 $a_3 + b_3$ 中较小的值对应的芯片.

(4) $k_1 < m_1, k_2 < m_2, k_3 = m_3$: 芯片 C 全部完成检测, 分别计算第 i 个芯片为 A 或 B 时满足约束条件的最小投放时间, 选择投放时间最小的芯片进入当前检测流程. 若投放时间一致, 优先选择 $a_1 + b_1$ 和 $a_2 + b_2$ 中较小的值对应的芯片.

(5) $k_1 = m_1, k_2 = m_2, k_3 < m_3$: 芯片 A 和 B 全部完成检测, 选择芯片 C 进入检测流程.

(6) $k_1 = m_1, k_2 < m_2, k_3 = m_3$: 芯片 A 和 C 全部完成检测, 选择芯片 B 进入检测流程.

(7) $k_1 < m_1, k_2 = m_2, k_3 = m_3$: 芯片 B 和 C 全部完成检测, 选择芯片 A 进入检测流程.

(8) $k_1 = m_1, k_2 = m_2, k_3 = m_3$: 完成检测任务, 算法结束.

7.2.3 举例验证

当芯片种类为 3 种时, 表 15 和表 16 分别为固定温育时间改变芯片数量和固定芯片数量改变温育时间的实验结果. 改变的芯片数量和温育时间都按倍数上升, 完成检测的总耗时随着芯片数量和温育时间的增加而上升, 模型能推广到芯片数量为 3 的情况.

表15 三种芯片时固定温育时间、改变芯片数量

芯片 1			芯片 2			芯片 3			总耗时/s
数量/片	第一次温育时间/min	第二次温育时间/min	数量/片	第一次温育时间/min	第二次温育时间/min	数量/片	第一次温育时间/min	第二次温育时间/min	
5	3	6	10	20	14	15	11	5	3639
10	3	6	20	20	14	30	11	5	5760
15	3	6	30	20	14	45	11	5	7496
20	3	6	40	20	14	60	11	5	9178
25	3	6	50	20	14	75	11	5	11142

表16 三种芯片时固定芯片数量、改变温育时间

芯片 1			芯片 2			芯片 3			总耗时/s
数量/片	第一次温育时间/min	第二次温育时间/min	数量/片	第一次温育时间/min	第二次温育时间/min	数量/片	第一次温育时间/min	第二次温育时间/min	
5	3	6	10	20	14	15	11	5	3639
5	6	12	10	40	28	15	22	10	3714
5	9	18	10	60	42	15	33	15	4568
5	12	24	10	80	56	15	44	20	5478
5	15	30	10	100	70	15	55	25	11830

7.3 N 种芯片的推广

7.3.1 对推广的可行性分析

本文采用算法的思想为时间序贯算法,即求解当前时刻下的最优解,并且在时间上进行迭代.对于 N 类芯片的情况,当可以加入新的芯片进行检测的时候,需要判断不同类型芯片是否与约束条件冲突,并找到最小的进入时间.在这个过程中,芯片种类的多少并不会引起约束条件的矛盾,因此可以推广到 N 类芯片.

除此之外,还可以利用数学归纳法的方式进行推断.假设算法在 $N-1$ 种类型的情况下成立,则可以利用由 7.1 节推广到 7.2 节的思想,推导出在 N 种类型的情况下也成立,并且在 7.1 节和 7.2 节中共已经证明两种和三种类型的情况是成立的.所以综上所述,可以推断出芯片种类推广到 N 类时,算法也是可行的.

7.3.2 算法在 N 种芯片情况下的推广

设一共有 N 种类型的芯片,第 i 种类型的芯片数量为 m_i ,两次温育时间为 a_i 、 b_i ,其中 $i = \{1, 2, \dots, N\}$. 通过比较 $\{m_1, m_2, \dots, m_N\}$ 和 $\{k_1, k_2, \dots, k_N\}$ 的大小关系,可以得出算法执行时可能会遇到 2^N 种不同的情况,分别对应于不同的芯片集合,具体分析流程与 2 种芯片、3 种芯片的情况类似.

8 问题五：算法分析与求解

问题五要求开始时处理两种芯片，运行三十分钟后新增加一种芯片，同时处理三种芯片，目标是最小化检测时间. 由上文的分析可知，问题五前期的处理背景与问题二类似，后期的处理背景与问题三类似，需要结合这两个算法来解决问题. 此时，运用于问题二的约束条件及优化目标同样适用于问题五，此处不再赘述.

8.1 算法分析

首先处理两种类型的芯片. 将代表进入检测流程的 A 芯片和 B 芯片个数的 k_1 和 k_2 赋值为 0, 使用算法二求解前 30 分钟进入检测流程的最优顺序. 1800 秒后，此时待处理的芯片类型变为 A、B、C 三种. 算法二停止运行，保存前 30 分钟进入检测流程的芯片的时间数据，同时将 k_1 、 k_2 的值以及存储时间数据的矩阵传递给算法三. 将进入检测流程的 C 芯片的个数赋值为 0, 然后开始处理三种类型的芯片，最后求出最短检测完成时间. 问题五的处理流程如下图 8 所示，算法的伪代码如表 17 所示.

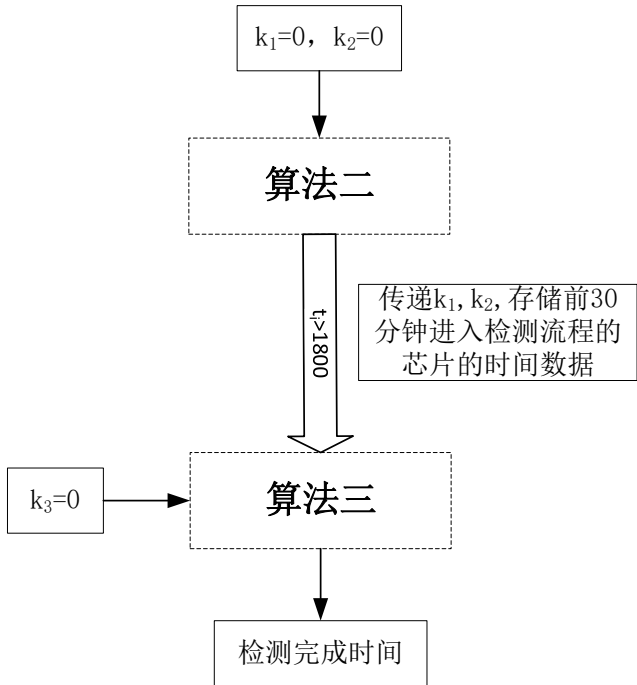


图8 算法四流程图

表17 算法四的伪代码

算法四：先有 A、B 型芯片后有 C 型芯片的算法
Input: 芯片 A 数量 60、芯片 B 数量 70、芯片 C 数量 20、芯片 A 第一次温育耗时、芯片 A 第二次温育耗时、芯片 B 第一次温育耗时、芯片 B 第二次温育耗时、芯片 C 第一次温育耗时、芯片 C 第二次温育耗时、磁珠加样耗时、清洗耗时、C 到达时间
Output: 时间矩阵 Matrix_time、芯片类型的加入顺序、所有芯片完成的总时间
01. 令第一个检测的芯片为 A, 定义已检测完成芯片数量 $k_1 = 1, k_2 = 0$
02. $t_{i1} = 0$
03. while 1

```

04.      if  $t_{i1} < 1800$ 
05.          按照算法二决策检测 A 或 B,  $k_1 = k_1 + 1$  or  $k_2 = k_2 + 1$ 
06.      else
07.          break
08.      end
09. end
10. 传递参数  $k_1$  和  $k_2$ , 令  $k_3 = 0$ 
11. while 1
12.     if  $k_1 == 60$  and  $k_2 == 70$  and  $k_3 == 20$ 
13.         break
14.     else
15.         按照算法三决策检测 A 或 B 或 C
16.     end
17. end
18. 根据  $\max(\text{Matrix\_time})$  计算得到所有芯片检测完成的时间  $t_A$ 
19. 令第一个检测的芯片为 B, 定义已检测完成芯片数量  $k_1 = 0, k_2 = 1$ 
20. 重复算法流程 2-18, 得到所有芯片检测完成的时间  $t_B$ 
21. 计算最短检测时间  $\min(t_A, t_B)$ 
22. 遍历所有磁珠加样耗时的可能取值, 重复算法流程 1-21, 得到最短的检测时间

```

8.2 问题求解

针对芯片 A、B、C 的温育时间背景, 结合编程工具, 可以获取得到 160 个检测芯片的传递顺序、每道工序的开始时间和结束时间. 表 18 为不同磁珠加样时间下本文算法总的时间消耗表, 其中先加入 A 型芯片在磁珠加样时间为 24s 时总耗时最小, 总耗时 12640 秒, 先加入 B 型芯片在磁珠加样时间为 22s 时总耗时最小, 总耗时 12906 秒. 先加入 B 型芯片比先加入 A 型芯片总耗时少 266s, 为了最小化总耗时, 选择先加入 B 型芯片. 所以若开始有 A 型芯片 60 片、B 型芯片 70 片, 运行 30 分钟后, 新到 C 型芯片 20 片, 全部检测完至少需要 12640s. 采取此种措施下仪器的工作数据见附录 A 附表 5, 部分调度数据见表 19.

表18 不同磁珠加样时间下本文算法总的时间消耗表

磁珠加样时间/s		21	22	23	24	25
按本文算法检测总耗时/s	先加入芯片 A	13366	12944	13066	12640	13070
	先加入芯片 B	13243	12906	13302	13078	13008

表19 先有 A、B 型芯片后有 C 型芯片的调度结果部分示意 (单位: s)

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	A	0	156	764	793	1109	1446	1471
2	B	36	192	2600	2632	3848	4185	4210
3	A	72	228	836	868	1184	1521	1546
4	B	108	264	2672	2704	3920	4257	4282
...	...	...	...	...	...	...	...	...
49	A	1768	1924	2532	2564	2880	3217	3242
50	B	1804	1960	4368	4400	5616	5953	5978
51	B	1844	2000	4408	4440	5656	5993	6018
52	B	1884	2040	4448	4480	5696	6033	6058
53	C	1928	2084	3592	3624	4240	4577	4602
54	B	1982	2138	4546	4578	5794	6131	6156

9 模型评价与推广

9.1 模型的优点

- (1) 本文所提出的模型在充分考虑约束条件的限制下, 采用序贯寻优的思想, 最小化每一个新加入芯片的处理时间, 最终减少任务完成时间.
- (2) 本文对约束条件进行量化, 充分考虑时间约束和空间约束, 将两种约束体现在时间线上, 易于编程实现.
- (3) 本文所提出的模型具有通用性, 算法一的模型可以推广至所有单个种类芯片的生产调度问题, 算法二的模型可以推广至所有情况下所有种类芯片的生产调度问题.
- (4) 本文所使用的算法寻优了执行单次任务时的最优磁珠加样时间.
- (5) 本文提出的算法和思路逻辑性强、通用性好, 计算效率高, 适用于各类生产调度问题.
- (6) 本文输出的调度策略相较于其他调度策略有效提高了生产效率.

9.2 模型的不足

- (1) 在建立问题二的数学模型时, 采用了贪心算法的思想, 最终获取的解并非全局最优.
- (2) 实际应用中转盘每次转动对位花费的时间不同, 磁珠工位和清洗工位的用时也不固定, 模型考虑的环境较为理想, 影响模型在实际环境中的使用效果.

9.3 模型的推广与改进

- (1) 本文从化学发光仪运行的调度问题出发, 综合考虑生产器件数量与生产时间带来的约束问题, 这对于模型有一定的实际应用价值.
- (2) 本文所提出的模型与算法可以方便地推广至更多芯片类型, 更复杂场景的情况, 也可为其他生产调度问题提供良好的参考.
- (3) 可以引入高效的全局寻优算法对模型进行求解, 以便获取更优的解.

参考文献

- [1] 司守奎, 孙玺菁主编; 孙兆亮, 周刚, 司宛灵, 康淑瑰参编. 数学建模算法与应用 第3版[M]. 北京: 国防工业出版社, 2021.
- [2] 窦建平, 李俊, 苏春. 可重构流水线构形选择和作业调度集成优化[J]. 东南大学学报(自然科学版), 2015, (5): 886-896.
- [3] 刘胜军. 混合流水线多目标调度优化研究[D]. 山东理工大学, 2016.
- [4] 钱斌, 刘荻飞, 胡蓉, 张梓琪. 混合迭代贪婪算法求解准时生产分布式流水线调度问题[J]. 控制与决策, 2022, 第37卷(11): 3042-3051.
- [5] 胡思贵, 王红蕾. 基于 GB/T2828.1(2012)的群序贯优化检验及应用[J]. 系统工程学报, 2020, 第35卷(5): 700-710.
- [6] 钱家昌, 程远胜, 张锦岚. 基于置信区间的约束多精度序贯代理模型优化方法及应用[J]. 中国舰船研究, 2021, 第16卷(4): 37-43.
- [7] 董海, 王瀚鹏. 基于种群迭代贪婪算法无等待流水车间调度[J]. 控制工程, 2023, 第30卷(5): 944-953.
- [8] 赵志明, 王建华, 朱康. 零空闲置换流水车间总延误最小迭代贪婪算法[J]. 组合机床与自动化加工技术, 2023, (3): 177-182.

附 录

附录 A: 附表

附表 1 问题 1. A 型芯片的调度结果(单位/s)

芯片 序号	进仓 时间	第一次温育 起始时间	磁珠加样 起始时间	第二次温育 起始时间	清洗起 始时间	检测起 始时间	检测结 束时间
1	0	156	764	793	1109	1446	1471
2	37	193	801	832	1148	1485	1510
3	76	232	840	871	1187	1524	1549
4	115	271	879	910	1226	1563	1588
5	154	310	918	949	1265	1602	1627
6	193	349	957	988	1304	1641	1666
7	232	388	996	1027	1343	1680	1705
8	271	427	1035	1066	1382	1719	1744
9	361	517	1125	1156	1472	1809	1834
10	400	556	1164	1195	1511	1848	1873
11	439	595	1203	1234	1550	1887	1912
12	478	634	1242	1273	1589	1926	1951
13	517	673	1281	1312	1628	1965	1990
14	556	712	1320	1351	1667	2004	2029
15	595	751	1359	1390	1706	2043	2068
16	651	807	1415	1446	1762	2099	2124
17	729	885	1493	1524	1840	2177	2202
18	768	924	1532	1563	1879	2216	2241
19	807	963	1571	1602	1918	2255	2280
20	846	1002	1610	1641	1957	2294	2319
21	885	1041	1649	1680	1996	2333	2358
22	924	1080	1688	1719	2035	2372	2397
23	975	1131	1739	1770	2086	2423	2448
24	1014	1170	1778	1809	2125	2462	2487
25	1092	1248	1856	1887	2203	2540	2565
26	1131	1287	1895	1926	2242	2579	2604
27	1170	1326	1934	1965	2281	2618	2643
28	1209	1365	1973	2004	2320	2657	2682
29	1248	1404	2012	2043	2359	2696	2721
30	1299	1455	2063	2094	2410	2747	2772
31	1572	1728	2336	2367	2683	3020	3045

32	1628	1784	2392	2423	2739	3076	3101
33	1667	1823	2431	2462	2778	3115	3140
34	1706	1862	2470	2501	2817	3154	3179
35	1745	1901	2509	2540	2856	3193	3218
36	1784	1940	2548	2579	2895	3232	3257
37	1823	1979	2587	2618	2934	3271	3296
38	1862	2018	2626	2657	2973	3310	3335
39	1952	2108	2716	2747	3063	3400	3425
40	1991	2147	2755	2786	3102	3439	3464
41	2030	2186	2794	2825	3141	3478	3503
42	2069	2225	2833	2864	3180	3517	3542
43	2108	2264	2872	2903	3219	3556	3581
44	2147	2303	2911	2942	3258	3595	3620
45	2186	2342	2950	2981	3297	3634	3659
46	2225	2381	2989	3020	3336	3673	3698
47	2320	2476	3084	3115	3431	3768	3793
48	2359	2515	3123	3154	3470	3807	3832
49	2398	2554	3162	3193	3509	3846	3871
50	2437	2593	3201	3232	3548	3885	3910
51	2476	2632	3240	3271	3587	3924	3949
52	2549	2705	3313	3344	3660	3997	4022
53	2605	2761	3369	3400	3716	4053	4078
54	2644	2800	3408	3439	3755	4092	4117
55	2683	2839	3447	3478	3794	4131	4156
56	2722	2878	3486	3517	3833	4170	4195
57	2761	2917	3525	3556	3872	4209	4234
58	2800	2956	3564	3595	3911	4248	4273
59	2839	2995	3603	3634	3950	4287	4312
60	3124	3280	3888	3919	4235	4572	4597
61	3163	3319	3927	3958	4274	4611	4636
62	3202	3358	3966	3997	4313	4650	4675
63	3258	3414	4022	4053	4369	4706	4731
64	3297	3453	4061	4092	4408	4745	4770
65	3336	3492	4100	4131	4447	4784	4809
66	3375	3531	4139	4170	4486	4823	4848
67	3414	3570	4178	4209	4525	4862	4887
68	3487	3643	4251	4282	4598	4935	4960

69	3526	3682	4290	4321	4637	4974	4999
70	3582	3738	4346	4377	4693	5030	5055
71	3621	3777	4385	4416	4732	5069	5094
72	3660	3816	4424	4455	4771	5108	5133
73	3699	3855	4463	4494	4810	5147	5172
74	3738	3894	4502	4533	4849	5186	5211
75	3777	3933	4541	4572	4888	5225	5250
76	3850	4006	4614	4645	4961	5298	5323
77	3911	4067	4675	4706	5022	5359	5384
78	3950	4106	4714	4745	5061	5398	5423
79	3989	4145	4753	4784	5100	5437	5462
80	4028	4184	4792	4823	5139	5476	5501
81	4101	4257	4865	4896	5212	5549	5574
82	4140	4296	4904	4935	5251	5588	5613
83	4179	4335	4943	4974	5290	5627	5652
84	4235	4391	4999	5030	5346	5683	5708
85	4274	4430	5038	5069	5385	5722	5747
86	4313	4469	5077	5108	5424	5761	5786
87	4352	4508	5116	5147	5463	5800	5825
88	4391	4547	5155	5186	5502	5839	5864
89	4464	4620	5228	5259	5575	5912	5937
90	4503	4659	5267	5298	5614	5951	5976
91	4676	4832	5440	5471	5787	6124	6149
92	4715	4871	5479	5510	5826	6163	6188
93	4754	4910	5518	5549	5865	6202	6227
94	4827	4983	5591	5622	5938	6275	6300
95	4888	5044	5652	5683	5999	6336	6361
96	4927	5083	5691	5722	6038	6375	6400
97	4966	5122	5730	5761	6077	6414	6439
98	5005	5161	5769	5800	6116	6453	6478
99	5078	5234	5842	5873	6189	6526	6551
100	5117	5273	5881	5912	6228	6565	6590
101	5156	5312	5920	5951	6267	6604	6629
102	5212	5368	5976	6007	6323	6660	6685
103	5251	5407	6015	6046	6362	6699	6724
104	5290	5446	6054	6085	6401	6738	6763
105	5329	5485	6093	6124	6440	6777	6802

106	5368	5524	6132	6163	6479	6816	6841
107	5441	5597	6205	6236	6552	6889	6914
108	5480	5636	6244	6275	6591	6928	6953
109	5541	5697	6305	6336	6652	6989	7014
110	5580	5736	6344	6375	6691	7028	7053
111	5653	5809	6417	6448	6764	7101	7126
112	5692	5848	6456	6487	6803	7140	7165
113	5731	5887	6495	6526	6842	7179	7204
114	5804	5960	6568	6599	6915	7252	7277
115	5865	6021	6629	6660	6976	7313	7338
116	5904	6060	6668	6699	7015	7352	7377
117	5943	6099	6707	6738	7054	7391	7416
118	5982	6138	6746	6777	7093	7430	7455
119	6055	6211	6819	6850	7166	7503	7528
120	6094	6250	6858	6889	7205	7542	7567
121	6133	6289	6897	6928	7244	7581	7606
122	6228	6384	6992	7023	7339	7676	7701
123	6267	6423	7031	7062	7378	7715	7740
124	6306	6462	7070	7101	7417	7754	7779
125	6345	6501	7109	7140	7456	7793	7818
126	6418	6574	7182	7213	7529	7866	7891
127	6457	6613	7221	7252	7568	7905	7930
128	6518	6674	7282	7313	7629	7966	7991
129	6557	6713	7321	7352	7668	8005	8030
130	6630	6786	7394	7425	7741	8078	8103
131	6669	6825	7433	7464	7780	8117	8142
132	6708	6864	7472	7503	7819	8156	8181
133	6781	6937	7545	7576	7892	8229	8254
134	6842	6998	7606	7637	7953	8290	8315
135	6881	7037	7645	7676	7992	8329	8354
136	6920	7076	7684	7715	8031	8368	8393
137	6959	7115	7723	7754	8070	8407	8432
138	7032	7188	7796	7827	8143	8480	8505
139	7071	7227	7835	7866	8182	8519	8544
140	7110	7266	7874	7905	8221	8558	8583
141	7205	7361	7969	8000	8316	8653	8678
142	7244	7400	8008	8039	8355	8692	8717

143	7283	7439	8047	8078	8394	8731	8756
144	7322	7478	8086	8117	8433	8770	8795
145	7395	7551	8159	8190	8506	8843	8868
146	7434	7590	8198	8229	8545	8882	8907
147	7495	7651	8259	8290	8606	8943	8968
148	7534	7690	8298	8329	8645	8982	9007
149	7607	7763	8371	8402	8718	9055	9080
150	7646	7802	8410	8441	8757	9094	9119
151	7685	7841	8449	8480	8796	9133	9158
152	7758	7914	8522	8553	8869	9206	9231
153	7819	7975	8583	8614	8930	9267	9292
154	7858	8014	8622	8653	8969	9306	9331
155	7897	8053	8661	8692	9008	9345	9370
156	7936	8092	8700	8731	9047	9384	9409
157	8009	8165	8773	8804	9120	9457	9482
158	8048	8204	8812	8843	9159	9496	9521
159	8087	8243	8851	8882	9198	9535	9560
160	8182	8338	8946	8977	9293	9630	9655
161	8221	8377	8985	9016	9332	9669	9694
162	8260	8416	9024	9055	9371	9708	9733
163	8299	8455	9063	9094	9410	9747	9772
164	8372	8528	9136	9167	9483	9820	9845
165	8411	8567	9175	9206	9522	9859	9884
166	8472	8628	9236	9267	9583	9920	9945
167	8511	8667	9275	9306	9622	9959	9984
168	8584	8740	9348	9379	9695	10032	10057
169	8623	8779	9387	9418	9734	10071	10096
170	8662	8818	9426	9457	9773	10110	10135
171	8735	8891	9499	9530	9846	10183	10208
172	8796	8952	9560	9591	9907	10244	10269
173	8835	8991	9599	9630	9946	10283	10308
174	8874	9030	9638	9669	9985	10322	10347
175	8913	9069	9677	9708	10024	10361	10386
176	8986	9142	9750	9781	10097	10434	10459
177	9025	9181	9789	9820	10136	10473	10498
178	9064	9220	9828	9859	10175	10512	10537
179	9159	9315	9923	9954	10270	10607	10632

180	9198	9354	9962	9993	10309	10646	10671
181	9237	9393	10001	10032	10348	10685	10710
182	9276	9432	10040	10071	10387	10724	10749
183	9349	9505	10113	10144	10460	10797	10822
184	9388	9544	10152	10183	10499	10836	10861
185	9449	9605	10213	10244	10560	10897	10922
186	9488	9644	10252	10283	10599	10936	10961
187	9561	9717	10325	10356	10672	11009	11034
188	9600	9756	10364	10395	10711	11048	11073
189	9639	9795	10403	10434	10750	11087	11112
190	9712	9868	10476	10507	10823	11160	11185
191	9773	9929	10537	10568	10884	11221	11246
192	9812	9968	10576	10607	10923	11260	11285
193	9851	10007	10615	10646	10962	11299	11324
194	9890	10046	10654	10685	11001	11338	11363
195	9963	10119	10727	10758	11074	11411	11436
196	10002	10158	10766	10797	11113	11450	11475
197	10041	10197	10805	10836	11152	11489	11514
198	10136	10292	10900	10931	11247	11584	11609
199	10175	10331	10939	10970	11286	11623	11648
200	10214	10370	10978	11009	11325	11662	11687

附表 2 问题 1. B 型芯片的调度结果(单位/s)

芯片序号	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	0	156	2564	2593	3809	4146	4171
2	37	193	2601	2632	3848	4185	4210
3	76	232	2640	2671	3887	4224	4249
4	115	271	2679	2710	3926	4263	4288
5	154	310	2718	2749	3965	4302	4327
6	193	349	2757	2788	4004	4341	4366
7	232	388	2796	2827	4043	4380	4405
8	271	427	2835	2866	4082	4419	4444
9	335	491	2899	2930	4146	4483	4508
10	374	530	2938	2969	4185	4522	4547
11	413	569	2977	3008	4224	4561	4586
12	452	608	3016	3047	4263	4600	4625

13	491	647	3055	3086	4302	4639	4664
14	530	686	3094	3125	4341	4678	4703
15	569	725	3133	3164	4380	4717	4742
16	608	764	3172	3203	4419	4756	4781
17	672	828	3236	3267	4483	4820	4845
18	711	867	3275	3306	4522	4859	4884
19	750	906	3314	3345	4561	4898	4923
20	789	945	3353	3384	4600	4937	4962
21	828	984	3392	3423	4639	4976	5001
22	867	1023	3431	3462	4678	5015	5040
23	906	1062	3470	3501	4717	5054	5079
24	945	1101	3509	3540	4756	5093	5118
25	1009	1165	3573	3604	4820	5157	5182
26	1048	1204	3612	3643	4859	5196	5221
27	1087	1243	3651	3682	4898	5235	5260
28	1126	1282	3690	3721	4937	5274	5299
29	1165	1321	3729	3760	4976	5313	5338
30	1222	1378	3786	3817	5033	5370	5395
31	1261	1417	3825	3856	5072	5409	5434
32	1300	1456	3864	3895	5111	5448	5473
33	1346	1502	3910	3941	5157	5494	5519
34	1385	1541	3949	3980	5196	5533	5558
35	1424	1580	3988	4019	5235	5572	5597
36	1463	1619	4027	4058	5274	5611	5636
37	1502	1658	4066	4097	5313	5650	5675
38	1559	1715	4123	4154	5370	5707	5732
39	1598	1754	4162	4193	5409	5746	5771
40	1637	1793	4201	4232	5448	5785	5810
41	3675	3831	6239	6270	7486	7823	7848
42	3714	3870	6278	6309	7525	7862	7887
43	3776	3932	6340	6371	7587	7924	7949
44	3815	3971	6379	6410	7626	7963	7988
45	3854	4010	6418	6449	7665	8002	8027
46	3893	4049	6457	6488	7704	8041	8066
47	3932	4088	6496	6527	7743	8080	8105
48	3973	4129	6537	6568	7784	8121	8146
49	4012	4168	6576	6607	7823	8160	8185

50	4051	4207	6615	6646	7862	8199	8224
51	4113	4269	6677	6708	7924	8261	8286
52	4152	4308	6716	6747	7963	8300	8325
53	4191	4347	6755	6786	8002	8339	8364
54	4230	4386	6794	6825	8041	8378	8403
55	4269	4425	6833	6864	8080	8417	8442
56	4310	4466	6874	6905	8121	8458	8483
57	4349	4505	6913	6944	8160	8497	8522
58	4388	4544	6952	6983	8199	8536	8561
59	4450	4606	7014	7045	8261	8598	8623
60	4489	4645	7053	7084	8300	8637	8662
61	4528	4684	7092	7123	8339	8676	8701
62	4567	4723	7131	7162	8378	8715	8740
63	4606	4762	7170	7201	8417	8754	8779
64	4647	4803	7211	7242	8458	8795	8820
65	4686	4842	7250	7281	8497	8834	8859
66	4725	4881	7289	7320	8536	8873	8898
67	4787	4943	7351	7382	8598	8935	8960
68	4826	4982	7390	7421	8637	8974	8999
69	4899	5055	7463	7494	8710	9047	9072
70	4938	5094	7502	7533	8749	9086	9111
71	5007	5163	7571	7602	8818	9155	9180
72	5046	5202	7610	7641	8857	9194	9219
73	5085	5241	7649	7680	8896	9233	9258
74	5124	5280	7688	7719	8935	9272	9297
75	5163	5319	7727	7758	8974	9311	9336
76	5236	5392	7800	7831	9047	9384	9409
77	5275	5431	7839	7870	9086	9423	9448
78	5337	5493	7901	7932	9148	9485	9510
79	5376	5532	7940	7971	9187	9524	9549
80	5415	5571	7979	8010	9226	9563	9588
81	7352	7508	9916	9947	11163	11500	11525
82	7391	7547	9955	9986	11202	11539	11564
83	7437	7593	10001	10032	11248	11585	11610
84	7476	7632	10040	10071	11287	11624	11649
85	7515	7671	10079	10110	11326	11663	11688
86	7554	7710	10118	10149	11365	11702	11727

87	7593	7749	10157	10188	11404	11741	11766
88	7634	7790	10198	10229	11445	11782	11807
89	7689	7845	10253	10284	11500	11837	11862
90	7728	7884	10292	10323	11539	11876	11901
91	7790	7946	10354	10385	11601	11938	11963
92	7829	7985	10393	10424	11640	11977	12002
93	7868	8024	10432	10463	11679	12016	12041
94	7907	8063	10471	10502	11718	12055	12080
95	7946	8102	10510	10541	11757	12094	12119
96	7985	8141	10549	10580	11796	12133	12158
97	8026	8182	10590	10621	11837	12174	12199
98	8065	8221	10629	10660	11876	12213	12238
99	8127	8283	10691	10722	11938	12275	12300
100	8166	8322	10730	10761	11977	12314	12339
101	8205	8361	10769	10800	12016	12353	12378
102	8244	8400	10808	10839	12055	12392	12417
103	8283	8439	10847	10878	12094	12431	12456
104	8322	8478	10886	10917	12133	12470	12495
105	8363	8519	10927	10958	12174	12511	12536
106	8402	8558	10966	10997	12213	12550	12575
107	8464	8620	11028	11059	12275	12612	12637
108	8503	8659	11067	11098	12314	12651	12676
109	8576	8732	11140	11171	12387	12724	12749
110	8615	8771	11179	11210	12426	12763	12788
111	8668	8824	11232	11263	12479	12816	12841
112	8707	8863	11271	11302	12518	12855	12880
113	8746	8902	11310	11341	12557	12894	12919
114	8785	8941	11349	11380	12596	12933	12958
115	8824	8980	11388	11419	12635	12972	12997
116	8913	9069	11477	11508	12724	13061	13086
117	8952	9108	11516	11547	12763	13100	13125
118	9014	9170	11578	11609	12825	13162	13187
119	9053	9209	11617	11648	12864	13201	13226
120	9092	9248	11656	11687	12903	13240	13265
121	11029	11185	13593	13624	14840	15177	15202
122	11068	11224	13632	13663	14879	15216	15241
123	11137	11293	13701	13732	14948	15285	15310

124	11176	11332	13740	13771	14987	15324	15349
125	11215	11371	13779	13810	15026	15363	15388
126	11254	11410	13818	13849	15065	15402	15427
127	11295	11451	13859	13890	15106	15443	15468
128	11366	11522	13930	13961	15177	15514	15539
129	11405	11561	13969	14000	15216	15553	15578
130	11467	11623	14031	14062	15278	15615	15640
131	11506	11662	14070	14101	15317	15654	15679
132	11545	11701	14109	14140	15356	15693	15718
133	11584	11740	14148	14179	15395	15732	15757
134	11623	11779	14187	14218	15434	15771	15796
135	11662	11818	14226	14257	15473	15810	15835
136	11703	11859	14267	14298	15514	15851	15876
137	11742	11898	14306	14337	15553	15890	15915
138	11804	11960	14368	14399	15615	15952	15977
139	11843	11999	14407	14438	15654	15991	16016
140	11882	12038	14446	14477	15693	16030	16055
141	11921	12077	14485	14516	15732	16069	16094
142	11960	12116	14524	14555	15771	16108	16133
143	11999	12155	14563	14594	15810	16147	16172
144	12040	12196	14604	14635	15851	16188	16213
145	12079	12235	14643	14674	15890	16227	16252
146	12141	12297	14705	14736	15952	16289	16314
147	12180	12336	14744	14775	15991	16328	16353
148	12253	12409	14817	14848	16064	16401	16426
149	12292	12448	14856	14887	16103	16440	16465
150	12331	12487	14895	14926	16142	16479	16504
151	12407	12563	14971	15002	16218	16555	16580
152	12446	12602	15010	15041	16257	16594	16619
153	12485	12641	15049	15080	16296	16633	16658
154	12524	12680	15088	15119	16335	16672	16697
155	12590	12746	15154	15185	16401	16738	16763
156	12629	12785	15193	15224	16440	16777	16802
157	12691	12847	15255	15286	16502	16839	16864
158	12730	12886	15294	15325	16541	16878	16903
159	12769	12925	15333	15364	16580	16917	16942
160	12808	12964	15372	15403	16619	16956	16981

161	14706	14862	17270	17301	18517	18854	18879
162	14745	14901	17309	17340	18556	18893	18918
163	14798	14954	17362	17393	18609	18946	18971
164	14837	14993	17401	17432	18648	18985	19010
165	14876	15032	17440	17471	18687	19024	19049
166	14915	15071	17479	17510	18726	19063	19088
167	14969	15125	17533	17564	18780	19117	19142
168	15043	15199	17607	17638	18854	19191	19216
169	15082	15238	17646	17677	18893	19230	19255
170	15144	15300	17708	17739	18955	19292	19317
171	15183	15339	17747	17778	18994	19331	19356
172	15222	15378	17786	17817	19033	19370	19395
173	15261	15417	17825	17856	19072	19409	19434
174	15300	15456	17864	17895	19111	19448	19473
175	15339	15495	17903	17934	19150	19487	19512
176	15380	15536	17944	17975	19191	19528	19553
177	15419	15575	17983	18014	19230	19567	19592
178	15481	15637	18045	18076	19292	19629	19654
179	15520	15676	18084	18115	19331	19668	19693
180	15559	15715	18123	18154	19370	19707	19732
181	15598	15754	18162	18193	19409	19746	19771
182	15637	15793	18201	18232	19448	19785	19810
183	15676	15832	18240	18271	19487	19824	19849
184	15717	15873	18281	18312	19528	19865	19890
185	15756	15912	18320	18351	19567	19904	19929
186	15818	15974	18382	18413	19629	19966	19991
187	15857	16013	18421	18452	19668	20005	20030
188	15930	16086	18494	18525	19741	20078	20103
189	15969	16125	18533	18564	19780	20117	20142
190	16022	16178	18586	18617	19833	20170	20195
191	16068	16224	18632	18663	19879	20216	20241
192	16107	16263	18671	18702	19918	20255	20280
193	16146	16302	18710	18741	19957	20294	20319
194	16193	16349	18757	18788	20004	20341	20366
195	16267	16423	18831	18862	20078	20415	20440
196	16306	16462	18870	18901	20117	20454	20479
197	16368	16524	18932	18963	20179	20516	20541

198	16407	16563	18971	19002	20218	20555	20580
199	16446	16602	19010	19041	20257	20594	20619
200	16485	16641	19049	19080	20296	20633	20658

附表 3 问题 2. A、B 型芯片的调度结果(单位/s)

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	B	0	156	2564	2593	3809	4146	4171
2	A	36	192	800	831	1147	1484	1509
3	B	72	228	2636	2667	3883	4220	4245
4	A	108	264	872	903	1219	1556	1581
5	B	144	300	2708	2739	3955	4292	4317
6	A	180	336	944	975	1291	1628	1653
7	B	216	372	2780	2811	4027	4364	4389
8	A	252	408	1016	1047	1363	1700	1725
9	B	288	444	2852	2883	4099	4436	4461
10	A	324	480	1088	1119	1435	1772	1797
11	B	360	516	2924	2955	4171	4508	4533
12	A	396	552	1160	1191	1507	1844	1869
13	B	432	588	2996	3027	4243	4580	4605
14	A	468	624	1232	1263	1579	1916	1941
15	B	504	660	3068	3099	4315	4652	4677
16	A	540	696	1304	1335	1651	1988	2013
17	B	576	732	3140	3171	4387	4724	4749
18	A	612	768	1376	1407	1723	2060	2085
19	B	650	806	3214	3245	4461	4798	4823
20	A	686	842	1450	1481	1797	2134	2159
21	B	722	878	3286	3317	4533	4870	4895
22	A	758	914	1522	1553	1869	2206	2231
23	B	794	950	3358	3389	4605	4942	4967
24	A	830	986	1594	1625	1941	2278	2303
25	B	866	1022	3430	3461	4677	5014	5039
26	A	902	1058	1666	1697	2013	2350	2375
27	B	938	1094	3502	3533	4749	5086	5111
28	A	974	1130	1738	1769	2085	2422	2447
29	B	1010	1166	3574	3605	4821	5158	5183
30	A	1046	1202	1810	1841	2157	2494	2519

31	B	1082	1238	3646	3677	4893	5230	5255
32	A	1118	1274	1882	1913	2229	2566	2591
33	B	1154	1310	3718	3749	4965	5302	5327
34	A	1190	1346	1954	1985	2301	2638	2663
35	B	1226	1382	3790	3821	5037	5374	5399
36	A	1262	1418	2026	2057	2373	2710	2735
37	B	1300	1456	3864	3895	5111	5448	5473
38	A	1372	1528	2136	2167	2483	2820	2845
39	A	1444	1600	2208	2239	2555	2892	2917
40	A	1516	1672	2280	2311	2627	2964	2989
41	A	1588	1744	2352	2383	2699	3036	3061
42	B	1624	1780	4188	4219	5435	5772	5797
43	A	1660	1816	2424	2455	2771	3108	3133
44	B	1696	1852	4260	4291	5507	5844	5869
45	A	1732	1888	2496	2527	2843	3180	3205
46	B	1768	1924	4332	4363	5579	5916	5941
47	B	1807	1963	4371	4402	5618	5955	5980
48	B	1876	2032	4440	4471	5687	6024	6049
49	B	1946	2102	4510	4541	5757	6094	6119
50	B	2018	2174	4582	4613	5829	6166	6191
51	B	2090	2246	4654	4685	5901	6238	6263
52	B	2162	2318	4726	4757	5973	6310	6335
53	B	2234	2390	4798	4829	6045	6382	6407
54	B	2274	2430	4838	4869	6085	6422	6447
55	B	2346	2502	4910	4941	6157	6494	6519
56	B	2414	2570	4978	5009	6225	6562	6587
57	B	2453	2609	5017	5048	6264	6601	6626
58	B	2492	2648	5056	5087	6303	6640	6665
59	B	2558	2714	5122	5153	6369	6706	6731
60	B	2597	2753	5161	5192	6408	6745	6770
61	B	2636	2792	5200	5231	6447	6784	6809
62	B	2702	2858	5266	5297	6513	6850	6875
63	B	2741	2897	5305	5336	6552	6889	6914
64	B	2780	2936	5344	5375	6591	6928	6953
65	B	3677	3833	6241	6272	7488	7825	7850
66	B	3747	3903	6311	6342	7558	7895	7920
67	B	3821	3977	6385	6416	7632	7969	7994

68	B	3891	4047	6455	6486	7702	8039	8064
69	B	3965	4121	6529	6560	7776	8113	8138
70	B	4038	4194	6602	6633	7849	8186	8211
71	B	4093	4249	6657	6688	7904	8241	8266
72	B	4165	4321	6729	6760	7976	8313	8338
73	B	4237	4393	6801	6832	8048	8385	8410
74	B	4321	4477	6885	6916	8132	8469	8494
75	B	4391	4547	6955	6986	8202	8539	8564
76	B	4463	4619	7027	7058	8274	8611	8636
77	B	4535	4691	7099	7130	8346	8683	8708
78	B	4607	4763	7171	7202	8418	8755	8780
79	A	4688	4844	5452	5483	5799	6136	6161
80	B	4743	4899	7307	7338	8554	8891	8916
81	B	4828	4984	7392	7423	8639	8976	9001
82	A	4906	5062	5670	5701	6017	6354	6379
83	B	4972	5128	7536	7567	8783	9120	9145
84	B	5317	5473	7881	7912	9128	9465	9490
85	B	5357	5513	7921	7952	9168	9505	9530
86	B	5429	5585	7993	8024	9240	9577	9602
87	B	5468	5624	8032	8063	9279	9616	9641
88	B	5551	5707	8115	8146	9362	9699	9724
89	B	5615	5771	8179	8210	9426	9763	9788
90	B	5654	5810	8218	8249	9465	9802	9827
91	B	5693	5849	8257	8288	9504	9841	9866
92	B	5759	5915	8323	8354	9570	9907	9932
93	B	5831	5987	8395	8426	9642	9979	10004
94	B	5870	6026	8434	8465	9681	10018	10043
95	B	5909	6065	8473	8504	9720	10057	10082
96	B	5967	6123	8531	8562	9778	10115	10140
97	B	6007	6163	8571	8602	9818	10155	10180
98	A	6076	6232	6840	6871	7187	7524	7549
99	B	6122	6278	8686	8717	9933	10270	10295
100	B	6161	6317	8725	8756	9972	10309	10334
101	B	6235	6391	8799	8830	10046	10383	10408
102	B	6274	6430	8838	8869	10085	10422	10447
103	B	6313	6469	8877	8908	10124	10461	10486
104	B	6363	6519	8927	8958	10174	10511	10536

105	B	6410	6566	8974	9005	10221	10558	10583
106	B	6452	6608	9016	9047	10263	10600	10625
107	B	7055	7211	9619	9650	10866	11203	11228
108	A	8421	8577	9185	9216	9532	9869	9894
109	A	8783	8939	9547	9578	9894	10231	10256
110	A	9185	9341	9949	9980	10296	10633	10658
111	A	9224	9380	9988	10019	10335	10672	10697
112	A	9434	9590	10198	10229	10545	10882	10907
113	A	9509	9665	10273	10304	10620	10957	10982
114	A	9548	9704	10312	10343	10659	10996	11021
115	A	9587	9743	10351	10382	10698	11035	11060
116	A	9628	9784	10392	10423	10739	11076	11101
117	A	9668	9824	10432	10463	10779	11116	11141
118	A	9707	9863	10471	10502	10818	11155	11180
119	A	9799	9955	10563	10594	10910	11247	11272
120	A	9872	10028	10636	10667	10983	11320	11345
121	A	9911	10067	10675	10706	11022	11359	11384
122	A	9952	10108	10716	10747	11063	11400	11425
123	A	9992	10148	10756	10787	11103	11440	11465
124	A	10031	10187	10795	10826	11142	11479	11504
125	A	10079	10235	10843	10874	11190	11527	11552
126	A	10123	10279	10887	10918	11234	11571	11596
127	A	10162	10318	10926	10957	11273	11610	11635
128	A	10242	10398	11006	11037	11353	11690	11715
129	A	10282	10438	11046	11077	11393	11730	11755
130	A	10321	10477	11085	11116	11432	11769	11794
131	A	10360	10516	11124	11155	11471	11808	11833
132	A	10447	10603	11211	11242	11558	11895	11920
133	A	10486	10642	11250	11281	11597	11934	11959
134	A	10525	10681	11289	11320	11636	11973	11998
135	A	10566	10722	11330	11361	11677	12014	12039
136	A	10606	10762	11370	11401	11717	12054	12079
137	A	10645	10801	11409	11440	11756	12093	12118
138	A	10737	10893	11501	11532	11848	12185	12210
139	A	10776	10932	11540	11571	11887	12224	12249
140	A	11005	11161	11769	11800	12116	12453	12478
141	A	11061	11217	11825	11856	12172	12509	12534

142	A	11100	11256	11864	11895	12211	12548	12573
143	A	11139	11295	11903	11934	12250	12587	12612
144	A	11180	11336	11944	11975	12291	12628	12653
145	A	11220	11376	11984	12015	12331	12668	12693
146	A	11259	11415	12023	12054	12370	12707	12732
147	A	11298	11454	12062	12093	12409	12746	12771
148	A	11360	11516	12124	12155	12471	12808	12833
149	A	11424	11580	12188	12219	12535	12872	12897
150	A	11463	11619	12227	12258	12574	12911	12936
151	A	11504	11660	12268	12299	12615	12952	12977
152	A	11544	11700	12308	12339	12655	12992	13017
153	A	11583	11739	12347	12378	12694	13031	13056
154	A	11622	11778	12386	12417	12733	13070	13095
155	A	11715	11871	12479	12510	12826	13163	13188
156	A	11754	11910	12518	12549	12865	13202	13227
157	A	11794	11950	12558	12589	12905	13242	13267
158	A	11834	11990	12598	12629	12945	13282	13307
159	A	11873	12029	12637	12668	12984	13321	13346
160	A	11912	12068	12676	12707	13023	13360	13385

附表 4 问题 3. A、B、C 型芯片的调度结果(单位/s)

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	B	0	156	2564	2593	3809	4146	4171
2	A	36	192	800	830	1146	1483	1508
3	C	72	228	1736	1766	2382	2719	2744
4	A	108	264	872	902	1218	1555	1580
5	C	144	300	1808	1838	2454	2791	2816
6	A	180	336	944	974	1290	1627	1652
7	C	216	372	1880	1910	2526	2863	2888
8	A	252	408	1016	1046	1362	1699	1724
9	B	288	444	2852	2882	4098	4435	4460
10	A	324	480	1088	1118	1434	1771	1796
11	C	360	516	2024	2054	2670	3007	3032
12	A	396	552	1160	1190	1506	1843	1868
13	C	432	588	2096	2126	2742	3079	3104
14	A	468	624	1232	1262	1578	1915	1940

15	C	504	660	2168	2198	2814	3151	3176
16	A	540	696	1304	1334	1650	1987	2012
17	B	576	732	3140	3170	4386	4723	4748
18	A	612	768	1376	1406	1722	2059	2084
19	C	650	806	2314	2344	2960	3297	3322
20	A	686	842	1450	1480	1796	2133	2158
21	B	722	878	3286	3316	4532	4869	4894
22	A	758	914	1522	1552	1868	2205	2230
23	B	794	950	3358	3388	4604	4941	4966
24	A	830	986	1594	1624	1940	2277	2302
25	B	866	1022	3430	3460	4676	5013	5038
26	A	902	1058	1666	1696	2012	2349	2374
27	C	938	1094	2602	2632	3248	3585	3610
28	B	974	1130	3538	3568	4784	5121	5146
29	B	1012	1168	3576	3606	4822	5159	5184
30	B	1068	1224	3632	3662	4878	5215	5240
31	B	1112	1268	3676	3706	4922	5259	5284
32	A	1154	1310	1918	1948	2264	2601	2626
33	B	1190	1346	3754	3784	5000	5337	5362
34	B	1228	1384	3792	3822	5038	5375	5400
35	B	1284	1440	3848	3878	5094	5431	5456
36	B	1330	1486	3894	3924	5140	5477	5502
37	B	1372	1528	3936	3966	5182	5519	5544
38	C	1428	1584	3092	3122	3738	4075	4100
39	B	1474	1630	4038	4068	5284	5621	5646
40	B	1516	1672	4080	4110	5326	5663	5688
41	B	1572	1728	4136	4166	5382	5719	5744
42	B	1616	1772	4180	4210	5426	5763	5788
43	B	1658	1814	4222	4252	5468	5805	5830
44	B	1696	1852	4260	4290	5506	5843	5868
45	B	1734	1890	4298	4328	5544	5881	5906
46	B	1800	1956	4364	4394	5610	5947	5972
47	B	1838	1994	4402	4432	5648	5985	6010
48	B	1876	2032	4440	4470	5686	6023	6048
49	A	1914	2070	2678	2708	3024	3361	3386
50	B	1950	2106	4514	4544	5760	6097	6122
51	B	1988	2144	4552	4582	5798	6135	6160

52	B	2048	2204	4612	4642	5858	6195	6220
53	B	2090	2246	4654	4684	5900	6237	6262
54	B	2128	2284	4692	4722	5938	6275	6300
55	B	2236	2392	4800	4830	6046	6383	6408
56	C	2319	2475	3983	4013	4629	4966	4991
57	B	2376	2532	4940	4970	6186	6523	6548
58	B	2558	2714	5122	5152	6368	6705	6730
59	B	2596	2752	5160	5190	6406	6743	6768
60	B	2664	2820	5228	5258	6474	6811	6836
61	B	2840	2996	5404	5434	6650	6987	7012
62	B	2882	3038	5446	5476	6692	7029	7054
63	B	3100	3256	5664	5694	6910	7247	7272
64	B	3604	3760	6168	6198	7414	7751	7776
65	C	3672	3828	5336	5366	5982	6319	6344
66	B	3960	4116	6524	6554	7770	8107	8132
67	B	4252	4408	6816	6846	8062	8399	8424
68	B	4402	4558	6966	6996	8212	8549	8574
69	B	4504	4660	7068	7098	8314	8651	8676
70	B	4542	4698	7106	7136	8352	8689	8714
71	B	4580	4736	7144	7174	8390	8727	8752
72	C	4634	4790	6298	6328	6944	7281	7306
73	B	4680	4836	7244	7274	8490	8827	8852
74	B	4728	4884	7292	7322	8538	8875	8900
75	B	4772	4928	7336	7366	8582	8919	8944
76	B	4858	5014	7422	7452	8668	9005	9030
77	B	4896	5052	7460	7490	8706	9043	9068
78	A	5042	5198	5806	5836	6152	6489	6514
79	C	5078	5234	6742	6772	7388	7725	7750
80	A	5152	5308	5916	5946	6262	6599	6624
81	C	5196	5352	6860	6890	7506	7843	7868
82	A	5326	5482	6090	6120	6436	6773	6798
83	C	5362	5518	7026	7056	7672	8009	8034
84	A	5476	5632	6240	6270	6586	6923	6948
85	A	5620	5776	6384	6414	6730	7067	7092
86	A	5722	5878	6486	6516	6832	7169	7194
87	C	5850	6006	7514	7544	8160	8497	8522
88	A	5906	6062	6670	6700	7016	7353	7378

89	C	5942	6098	7606	7636	8252	8589	8614
90	C	6120	6276	7784	7814	8430	8767	8792
91	C	6297	6453	7961	7991	8607	8944	8969
92	C	6436	6592	8100	8130	8746	9083	9108
93	C	6474	6630	8138	8168	8784	9121	9146
94	C	6596	6752	8260	8290	8906	9243	9268
95	C	6634	6790	8298	8328	8944	9281	9306
96	C	6710	6866	8374	8404	9020	9357	9382
97	A	6794	6950	7558	7588	7904	8241	8266
98	A	6918	7074	7682	7712	8028	8365	8390
99	C	6956	7112	8620	8650	9266	9603	9628
100	C	7024	7180	8688	8718	9334	9671	9696
101	C	7062	7218	8726	8756	9372	9709	9734
102	C	7100	7256	8764	8794	9410	9747	9772
103	C	7142	7298	8806	8836	9452	9789	9814
104	C	7186	7342	8850	8880	9496	9833	9858
105	C	7288	7444	8952	8982	9598	9935	9960
106	C	7364	7520	9028	9058	9674	10011	10036
107	C	7408	7564	9072	9102	9718	10055	10080
108	C	7456	7612	9120	9150	9766	10103	10128
109	C	7494	7650	9158	9188	9804	10141	10166
110	C	7532	7688	9196	9226	9842	10179	10204
111	C	7580	7736	9244	9274	9890	10227	10252
112	C	7620	7776	9284	9314	9930	10267	10292
113	C	7686	7842	9350	9380	9996	10333	10358
114	C	7724	7880	9388	9418	10034	10371	10396
115	C	7766	7922	9430	9460	10076	10413	10438
116	C	7811	7967	9475	9505	10121	10458	10483
117	C	7849	8005	9513	9543	10159	10496	10521
118	C	7887	8043	9551	9581	10197	10534	10559
119	C	7950	8106	9614	9644	10260	10597	10622
120	C	7988	8144	9652	9682	10298	10635	10660
121	C	8032	8188	9696	9726	10342	10679	10704
122	C	8080	8236	9744	9774	10390	10727	10752
123	C	8118	8274	9782	9812	10428	10765	10790
124	C	8204	8360	9868	9898	10514	10851	10876
125	A	9373	9529	10137	10167	10483	10820	10845

126	A	9448	9604	10212	10242	10558	10895	10920
127	A	9697	9853	10461	10491	10807	11144	11169
128	A	9802	9958	10566	10596	10912	11249	11274
129	A	9846	10002	10610	10640	10956	11293	11318
130	A	9884	10040	10648	10678	10994	11331	11356
131	A	9926	10082	10690	10720	11036	11373	11398
132	A	9971	10127	10735	10765	11081	11418	11443
133	A	10021	10177	10785	10815	11131	11468	11493
134	A	10062	10218	10826	10856	11172	11509	11534
135	A	10126	10282	10890	10920	11236	11573	11598
136	A	10170	10326	10934	10964	11280	11617	11642
137	A	10208	10364	10972	11002	11318	11655	11680
138	A	10250	10406	11014	11044	11360	11697	11722
139	A	10295	10451	11059	11089	11405	11742	11767
140	A	10386	10542	11150	11180	11496	11833	11858
141	A	10424	10580	11188	11218	11534	11871	11896
142	A	10498	10654	11262	11292	11608	11945	11970
143	A	10619	10775	11383	11413	11729	12066	12091
144	A	10676	10832	11440	11470	11786	12123	12148
145	A	10714	10870	11478	11508	11824	12161	12186
146	A	10784	10940	11548	11578	11894	12231	12256
147	A	10822	10978	11586	11616	11932	12269	12294
148	A	10864	11020	11628	11658	11974	12311	12336
149	A	10909	11065	11673	11703	12019	12356	12381
150	A	10947	11103	11711	11741	12057	12394	12419
151	A	11000	11156	11764	11794	12110	12447	12472
152	A	11038	11194	11802	11832	12148	12485	12510
153	A	11146	11302	11910	11940	12256	12593	12618
154	A	11188	11344	11952	11982	12298	12635	12660
155	A	11233	11389	11997	12027	12343	12680	12705
156	A	11271	11427	12035	12065	12381	12718	12743
157	A	11362	11518	12126	12156	12472	12809	12834
158	A	11400	11556	12164	12194	12510	12847	12872
159	A	11595	11751	12359	12389	12705	13042	13067
160	A	11652	11808	12416	12446	12762	13099	13124

附表 5 问题 5. 先有 A、B 型芯片后有 B 型芯片的调度结果(单位/s)

芯片序号	芯片类型	进仓时间	第一次温育起始时间	磁珠加样起始时间	第二次温育起始时间	清洗起始时间	检测起始时间	检测结束时间
1	A	0	156	764	793	1109	1446	1471
2	B	36	192	2600	2632	3848	4185	4210
3	A	72	228	836	868	1184	1521	1546
4	B	108	264	2672	2704	3920	4257	4282
5	A	144	300	908	940	1256	1593	1618
6	B	180	336	2744	2776	3992	4329	4354
7	A	216	372	980	1012	1328	1665	1690
8	B	252	408	2816	2848	4064	4401	4426
9	A	288	444	1052	1084	1400	1737	1762
10	B	324	480	2888	2920	4136	4473	4498
11	A	360	516	1124	1156	1472	1809	1834
12	B	396	552	2960	2992	4208	4545	4570
13	A	432	588	1196	1228	1544	1881	1906
14	B	468	624	3032	3064	4280	4617	4642
15	A	504	660	1268	1300	1616	1953	1978
16	B	540	696	3104	3136	4352	4689	4714
17	A	576	732	1340	1372	1688	2025	2050
18	B	614	770	3178	3210	4426	4763	4788
19	A	650	806	1414	1446	1762	2099	2124
20	B	686	842	3250	3282	4498	4835	4860
21	A	722	878	1486	1518	1834	2171	2196
22	B	758	914	3322	3354	4570	4907	4932
23	A	794	950	1558	1590	1906	2243	2268
24	B	830	986	3394	3426	4642	4979	5004
25	A	866	1022	1630	1662	1978	2315	2340
26	B	902	1058	3466	3498	4714	5051	5076
27	B	959	1115	3523	3555	4771	5108	5133
28	A	1006	1162	1770	1802	2118	2455	2480
29	A	1046	1202	1810	1842	2158	2495	2520
30	B	1082	1238	3646	3678	4894	5231	5256
31	A	1118	1274	1882	1914	2230	2567	2592
32	B	1154	1310	3718	3750	4966	5303	5328
33	A	1190	1346	1954	1986	2302	2639	2664

34	B	1226	1382	3790	3822	5038	5375	5400
35	A	1264	1420	2028	2060	2376	2713	2738
36	B	1300	1456	3864	3896	5112	5449	5474
37	A	1336	1492	2100	2132	2448	2785	2810
38	B	1372	1528	3936	3968	5184	5521	5546
39	A	1408	1564	2172	2204	2520	2857	2882
40	B	1444	1600	4008	4040	5256	5593	5618
41	A	1480	1636	2244	2276	2592	2929	2954
42	B	1516	1672	4080	4112	5328	5665	5690
43	A	1552	1708	2316	2348	2664	3001	3026
44	B	1588	1744	4152	4184	5400	5737	5762
45	A	1624	1780	2388	2420	2736	3073	3098
46	B	1660	1816	4224	4256	5472	5809	5834
47	A	1696	1852	2460	2492	2808	3145	3170
48	B	1732	1888	4296	4328	5544	5881	5906
49	A	1768	1924	2532	2564	2880	3217	3242
50	B	1804	1960	4368	4400	5616	5953	5978
51	B	1844	2000	4408	4440	5656	5993	6018
52	B	1884	2040	4448	4480	5696	6033	6058
53	C	1928	2084	3592	3624	4240	4577	4602
54	B	1982	2138	4546	4578	5794	6131	6156
55	B	2022	2178	4586	4618	5834	6171	6196
56	B	2094	2250	4658	4690	5906	6243	6268
57	B	2183	2339	4747	4779	5995	6332	6357
58	B	2238	2394	4802	4834	6050	6387	6412
59	B	2310	2466	4874	4906	6122	6459	6484
60	B	2382	2538	4946	4978	6194	6531	6556
61	B	2450	2606	5014	5046	6262	6599	6624
62	B	2524	2680	5088	5120	6336	6673	6698
63	B	2596	2752	5160	5192	6408	6745	6770
64	B	2668	2824	5232	5264	6480	6817	6842
65	B	2740	2896	5304	5336	6552	6889	6914
66	B	3714	3870	6278	6310	7526	7863	7888
67	B	3786	3942	6350	6382	7598	7935	7960
68	B	3858	4014	6422	6454	7670	8007	8032
69	B	3930	4086	6494	6526	7742	8079	8104
70	B	4002	4158	6566	6598	7814	8151	8176

71	C	4058	4214	5722	5754	6370	6707	6732
72	B	4106	4262	6670	6702	7918	8255	8280
73	B	4146	4302	6710	6742	7958	8295	8320
74	B	4202	4358	6766	6798	8014	8351	8376
75	C	4276	4432	5940	5972	6588	6925	6950
76	B	4348	4504	6912	6944	8160	8497	8522
77	B	4436	4592	7000	7032	8248	8585	8610
78	B	4492	4648	7056	7088	8304	8641	8666
79	B	4564	4720	7128	7160	8376	8713	8738
80	B	4629	4785	7193	7225	8441	8778	8803
81	A	4756	4912	5520	5552	5868	6205	6230
82	B	4828	4984	7392	7424	8640	8977	9002
83	B	4896	5052	7460	7492	8708	9045	9070
84	B	4970	5126	7534	7566	8782	9119	9144
85	B	5042	5198	7606	7638	8854	9191	9216
86	B	5114	5270	7678	7710	8926	9263	9288
87	B	5186	5342	7750	7782	8998	9335	9360
88	B	5258	5414	7822	7854	9070	9407	9432
89	B	5330	5486	7894	7926	9142	9479	9504
90	B	5402	5558	7966	7998	9214	9551	9576
91	B	5466	5622	8030	8062	9278	9615	9640
92	B	5506	5662	8070	8102	9318	9655	9680
93	B	5546	5702	8110	8142	9358	9695	9720
94	B	5660	5816	8224	8256	9472	9809	9834
95	B	5718	5874	8282	8314	9530	9867	9892
96	B	5758	5914	8322	8354	9570	9907	9932
97	B	5822	5978	8386	8418	9634	9971	9996
98	A	5858	6014	6622	6654	6970	7307	7332
99	B	5900	6056	8464	8496	9712	10049	10074
100	B	5972	6128	8536	8568	9784	10121	10146
101	A	6064	6220	6828	6860	7176	7513	7538
102	C	6504	6660	8168	8200	8816	9153	9178
103	C	7094	7250	8758	8790	9406	9743	9768
104	C	7134	7290	8798	8830	9446	9783	9808
105	C	7347	7503	9011	9043	9659	9996	10021
106	C	7497	7653	9161	9193	9809	10146	10171
107	C	7560	7716	9224	9256	9872	10209	10234

108	C	7632	7788	9296	9328	9944	10281	10306
109	C	7672	7828	9336	9368	9984	10321	10346
110	C	7718	7874	9382	9414	10030	10367	10392
111	C	7816	7972	9480	9512	10128	10465	10490
112	C	7882	8038	9546	9578	10194	10531	10556
113	A	7920	8076	8684	8716	9032	9369	9394
114	C	7978	8134	9642	9674	10290	10627	10652
115	C	8024	8180	9688	9720	10336	10673	10698
116	C	8184	8340	9848	9880	10496	10833	10858
117	C	8296	8452	9960	9992	10608	10945	10970
118	C	8346	8502	10010	10042	10658	10995	11020
119	C	8386	8542	10050	10082	10698	11035	11060
120	A	9128	9284	9892	9924	10240	10577	10602
121	A	9340	9496	10104	10136	10452	10789	10814
122	A	9452	9608	10216	10248	10564	10901	10926
123	A	9611	9767	10375	10407	10723	11060	11085
124	A	9664	9820	10428	10460	10776	11113	11138
125	A	9742	9898	10506	10538	10854	11191	11216
126	A	9910	10066	10674	10706	11022	11359	11384
127	A	9988	10144	10752	10784	11100	11437	11462
128	A	10066	10222	10830	10862	11178	11515	11540
129	A	10106	10262	10870	10902	11218	11555	11580
130	A	10146	10302	10910	10942	11258	11595	11620
131	A	10186	10342	10950	10982	11298	11635	11660
132	A	10234	10390	10998	11030	11346	11683	11708
133	A	10278	10434	11042	11074	11390	11727	11752
134	A	10318	10474	11082	11114	11430	11767	11792
135	A	10358	10514	11122	11154	11470	11807	11832
136	A	10430	10586	11194	11226	11542	11879	11904
137	A	10470	10626	11234	11266	11582	11919	11944
138	A	10510	10666	11274	11306	11622	11959	11984
139	A	10602	10758	11366	11398	11714	12051	12076
140	A	10642	10798	11406	11438	11754	12091	12116
141	A	10682	10838	11446	11478	11794	12131	12156
142	A	10722	10878	11486	11518	11834	12171	12196
143	A	10762	10918	11526	11558	11874	12211	12236
144	A	10802	10958	11566	11598	11914	12251	12276

145	A	10892	11048	11656	11688	12004	12341	12366
146	A	10972	11128	11736	11768	12084	12421	12446
147	A	11046	11202	11810	11842	12158	12495	12520
148	A	11086	11242	11850	11882	12198	12535	12560
149	A	11126	11282	11890	11922	12238	12575	12600
150	A	11166	11322	11930	11962	12278	12615	12640

附录 B: 主要程序/关键代码

代	操作系统: Windows 11 (Version 22H2.21H2)
码	编程语言: MATLAB
环	编辑器: MATLAB R2022a
境	代码详见: code

代码清单 1 问题 1 主函数

```

tttt = [];
for iiii = 1:5
warmgrowth1 = 600; % 温育时间
warmgrowth2 = 300;
N = 79; % 芯片数量
B = 20+iiii; % 21-25
Wash = 325; % 清洗时间
% 初始时间矩阵
matrix time = [0 150 156 156+warmgrowth1 156+warmgrowth1+8 156+warmgrowth1+8+21 156+w
armgrowth1+8+21+8 156+warmgrowth1+8+21+8+warmgrowth2 156+warmgrowth1+8+21+8+warmgro
wth2+16 156+warmgrowth1+8+21+8+warmgrowth2+16+Wash 156+warmgrowth1+8+21+8+warmgro
h2+16+Wash+12 156+warmgrowth1+8+21+8+warmgrowth2+16+Wash+12+25];

ti = 0;
% 遍历所有芯片
for i = 1:N
ti = ti + 36;
ifcorrect = 1;
while ifcorrect
% 新加入芯片的时间数组
matrix x = [ti ti+150 ti+156 ti+156+warmgrowth1 ti+156+warmgrowth1+8 ti+156+warmgrowth
1+8+B ti+156+warmgrowth1+8+B+8 ti+156+warmgrowth1+8+B+8+warmgrowth2 ti+156+warmgrowt
h1+8+B+8+warmgrowth2+16 ti+156+warmgrowth1+8+B+8+warmgrowth2+16+Wash ti+156+warmgro
wth1+8+B+8+warmgrowth2+16+Wash+12 ti+156+warmgrowth1+8+B+8+warmgrowth2+16+Wash+12+
25];
[chongtu Matrix] = calculate chongtu(matrix time);
chongtu 1 = F1(matrix x(2),matrix x(3));
chongtu 2 = F1(matrix x(4),matrix x(5));
chongtu 3 = F1(matrix x(6),matrix x(7));
matrix y = [chongtu Matrix chongtu 1 chongtu 2 chongtu 3];
[~, YY] = mode(matrix y); % 计算转轮是否冲突
H = size(matrix time,1);
cizhu pre = F1(matrix time(H,5),matrix time(H,6));
cizhu hou = F1(matrix x(5),matrix x(6));
cizhu = [cizhu hou cizhu pre];
[~, XX] = mode(cizhu); % 计算磁珠是否冲突
% 小于8 不用考虑位置不足
if H < 8
if YY == 1 && XX == 1
ifcorrect = 0;
matrix time = [matrix time; matrix x];
break
else
ti = ti + 1;

```

```

        end
        % 考虑清洗盘位置是否足够
        elseif H >= 8 && H < 40
            if YY == 1 && XX == 1 && ti+188+B+warmgrowth2+warmgrowth1 >= matrix_time(H-
7,11)
                ifcorrect = 0;
                matrix_time = [matrix_time; matrix_x];
                break
            else
                ti = ti + 1;
            end
        end
        % 考虑转盘上的位置是否足够
        elseif H>=40
            if YY == 1 && XX == 1 && ti+188+B+warmgrowth2+warmgrowth1 >= matrix_time(H-
7,11) && ti+156 >= matrix_time(H-39,9)
                ifcorrect = 0;
                matrix_time = [matrix_time; matrix_x];
                break
            else
                ti = ti + 1;
            end
        end
    end
end
end
tttt = [tttt max(matrix_time(:,12'))];
end
plot(tttt)
% xlswrite('问1芯片C25',matrix_time,'Sheet1')

```

代码清单 2 区间采样函数

```

function [M] = F1(x1,x2)
    %UNTITLED3 此处提供此函数的摘要
    % 此处提供详细说明
    M = x1:(x2-1);
end

```

代码清单 3 冲突计算函数

```

function [chongtu_Matrix] = calculate_chongtu(matrix_time)
    N = size(matrix_time,1);
    chongtu_Matrix = [];
    for i = 1:N
        chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,2),matrix_time(i,3))];
        chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,4),matrix_time(i,5))];
        chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,6),matrix_time(i,7))];
        chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,8),matrix_time(i,9))];
    end
end

```

代码清单 4 问题 2 主函数

```

% ttttt = [];
% for iiii = 1:5
warm_growth_A1 = 600;
warm_growth_A2 = 300;
warm_growth_B1 = 2400;
warm_growth_B2 = 1200;
M = 80; % A 芯片数量

```

```

N = 80; % B 芯片数量
B = 23; % 磁珠工位时间
Wash = 325; % 清洗时间

% 初始的时间矩阵有两种，分别考虑 A 第一个和 B 第一个
% 假设 A 为第一个
% matrix_time = [0 150 156 156+warm_growth_A1 156+warm_growth_A1+8
    156+warm_growth_A1+8+21 156+warm_growth_A1+8+21+8
    156+warm_growth_A1+8+21+8+warm_growth_A2
    156+warm_growth_A1+8+21+8+warm_growth_A2+16
    156+warm_growth_A1+8+21+8+warm_growth_A2+16+Wash
    156+warm_growth_A1+8+21+8+warm_growth_A2+16+Wash+12
    156+warm_growth_A1+8+21+8+warm_growth_A2+16+Wash+12+25 1];
% 假设 B 为第一个
matrix_time = [0 150 156 156+warm_growth_B1 156+warm_growth_B1+8 156+warm_growth_B1+8+21
    156+warm_growth_B1+8+21+8 156+warm_growth_B1+8+21+8+warm_growth_B2
    156+warm_growth_B1+8+21+8+warm_growth_B2+16
    156+warm_growth_B1+8+21+8+warm_growth_B2+16+Wash
    156+warm_growth_B1+8+21+8+warm_growth_B2+16+Wash+12
    156+warm_growth_B1+8+21+8+warm_growth_B2+16+Wash+12+25 2];

ti = 0;
k1 = 0; k2 = 1; % A、B 芯片的加入个数
% 遍历所有芯片
for i = 1:M+N-1
    ti = ti + 36;

    if k1 < M && k2 < N
        while 1
            % 写出两个芯片对应的新时间节点矩阵
            matrix_x_B = [ti ti+150 ti+156 ti+156+warm_growth_B1 ti+156+warm_growth_B1+8
                ti+156+warm_growth_B1+8+B ti+156+warm_growth_B1+8+B+8
                ti+156+warm_growth_B1+8+B+8+warm_growth_B2
                ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16
                ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash
                ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash+12
                ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash+12+25 2];
            matrix_x_A = [ti ti+150 ti+156 ti+156+warm_growth_A1 ti+156+warm_growth_A1+8
                ti+156+warm_growth_A1+8+B ti+156+warm_growth_A1+8+B+8
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash+12
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash+12+25 1];

            % 分别计算当前 ti 下 A、B 的冲突参数
            % 计算当前 ti 下 A 的冲突参数
            % 计算转轮是否冲突
            [YA] = zhuanpan(matrix_time, matrix_x_A);
            % 计算磁珠是否冲突
            [XA] = cizhugongwei(matrix_time, matrix_x_A);
            % 计算检测工位是否冲突
            [ZA] = jiancegongwei(matrix_time, matrix_x_A);
            % 计算温育盘是否冲突
            [TA] = wenyushi(matrix_time, matrix_x_A);
            % 计算清洗工位是否冲突

```

```

[PA] = qingxihe(matrix_time,matrix_x_A);

% 计算当前 ti 下 B 的冲突参数
% 计算转轮是否冲突
[YB] = zhuanpan(matrix_time,matrix_x_B);
% 计算磁珠是否冲突
[XB] = cizhugongwei(matrix_time,matrix_x_B);
% 计算检测工位是否冲突
[ZB] = jiancegongwei(matrix_time,matrix_x_B);
% 计算温育盘是否冲突
[TB] = wenyushi(matrix_time,matrix_x_B);
% 计算清洗工位是否冲突
[PB] = qingxihe(matrix_time,matrix_x_B);

% 考虑数量约束, 判断添加 A 还是 B
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
    matrix_time = [matrix_time; matrix_x_A];
    k1 = k1 + 1;
    break
end
if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
    matrix_time = [matrix_time; matrix_x_B];
    k2 = k2 + 1;
    break
else
    ti = ti + 1;
end
end

% A 芯片满了, 只添加 B 芯片
elseif k1==M && k2<N
    while 1
        matrix_x_B = [ti ti+150 ti+156 ti+156+warm_growth_B1 ti+156+warm_growth_B1+8
            ti+156+warm_growth_B1+8+B ti+156+warm_growth_B1+8+B+8
            ti+156+warm_growth_B1+8+B+8+warm_growth_B2
            ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16
            ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash
            ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash+12
            ti+156+warm_growth_B1+8+B+8+warm_growth_B2+16+Wash+12+25 2];
        % 计算转轮是否冲突
        [YB] = zhuanpan(matrix_time,matrix_x_B);
        % 计算磁珠是否冲突
        [XB] = cizhugongwei(matrix_time,matrix_x_B);
        % 计算检测工位是否冲突
        [ZB] = jiancegongwei(matrix_time,matrix_x_B);
        % 计算温育盘是否冲突
        [TB] = wenyushi(matrix_time,matrix_x_B);
        % 计算清洗工位是否冲突
        [PB] = qingxihe(matrix_time,matrix_x_B);

        if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
            matrix_time = [matrix_time; matrix_x_B];
            k2 = k2 + 1;
            break
        else
            ti = ti + 1;
        end
    end
end

```

```

        end
    end

    % B 芯片满了, 只添加 A 芯片
    elseif k1<M && k2==N
        while 1
            matrix_x_A = [ti ti+150 ti+156 ti+156+warm_growth_A1 ti+156+warm_growth_A1+8
                ti+156+warm_growth_A1+8+B ti+156+warm_growth_A1+8+B+8
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash+12
                ti+156+warm_growth_A1+8+B+8+warm_growth_A2+16+Wash+12+25 1];

            % 计算转轮是否冲突
            [YA] = zhuanpan(matrix_time,matrix_x_A);
            % 计算磁珠是否冲突
            [XA] = cizhugongwei(matrix_time,matrix_x_A);
            % 计算检测工位是否冲突
            [ZA] = jiancegongwei(matrix_time,matrix_x_A);
            % 计算温育盘是否冲突
            [TA] = wenyushi(matrix_time,matrix_x_A);
            % 计算清洗工位是否冲突
            [PA] = qingxihe(matrix_time,matrix_x_A);

            if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
                matrix_time = [matrix_time; matrix_x_A];
                k1 = k1 + 1;
                break
            else
                ti = ti + 1;
            end
        end
    end

    % A、B 芯片都满了
    elseif k1==M && k2==N
        break
    end
end

end
% tttttt = [ttttt max(matrix_time(:,12))];
% end
% plot(tttttt)

% xlswrite('表 3',matrix_time,'Sheet1')

```

代码清单 5 转盘冲突判断函数

```

function [tcount] = zhuanpan(matrix_time,matrix_x)
N = size(matrix_time,1);
chongtu_Matrix = [];
for i = 1:N
    chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,2),matrix_time(i,3))];
    chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,4),matrix_time(i,5))];
    chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,6),matrix_time(i,7))];
    chongtu_Matrix = [chongtu_Matrix F1(matrix_time(i,8),matrix_time(i,9))];
end
chongtu_Matrix = [chongtu_Matrix F1(matrix_x(2),matrix_x(3))];
chongtu_Matrix = [chongtu_Matrix F1(matrix_x(4),matrix_x(5))];
chongtu_Matrix = [chongtu_Matrix F1(matrix_x(6),matrix_x(7))];
chongtu_Matrix = [chongtu_Matrix F1(matrix_x(8),matrix_x(9))];

```

```
[~, Y] = mode(chongtu Matrix);
if Y == 1
    tcount = 1; % 1代表没有冲突
else
    tcount = 0;
end
end
```

代码清单 6 温育冲突判断函数

```
function [tcount] = wenyushi(matrix time,matrix x)
N = size(matrix time,1);
if N > 40
chongtu Matrix = [];
for i = 1:N
    chongtu Matrix = [chongtu Matrix F1(matrix time(i,3),matrix time(i,9))];
end
chongtu Matrix = [chongtu Matrix F1(matrix x(3),matrix x(9))];
[~, Y] = mode(chongtu Matrix);
if Y <= 40
    tcount = 1; % 1代表没有冲突
else
    tcount = 0;
end
else
    tcount = 1;
end
end
```

代码清单 7 清洗盒冲突判断函数

```
function [tcount] = qingxihe(matrix time,matrix x)
N = size(matrix time,1);
if N > 8
chongtu Matrix = [];
for i = 1:N
    chongtu Matrix = [chongtu Matrix F1(matrix time(i,9),matrix time(i,11))];
end
chongtu Matrix = [chongtu Matrix F1(matrix x(9),matrix x(11))];
[~, Y] = mode(chongtu Matrix);
if Y <= 8
    tcount = 1; % 1代表没有冲突
else
    tcount = 0;
end
else
    tcount = 1;
end
end
```

代码清单 8 检测工位冲突判断函数

```
function [tcount] = jiancegongwei(matrix time,matrix x)
N = size(matrix time,1);
chongtu Matrix = [];
for i = 1:N
    chongtu Matrix = [chongtu Matrix F1(matrix time(i,11),matrix time(i,12))];
end
chongtu Matrix = [chongtu Matrix F1(matrix x(11),matrix x(12))];
[~, Y] = mode(chongtu Matrix);
if Y == 1
    tcount = 1; % 1代表没有冲突
else
    tcount = 0;
end
end
```

代码清单 9 磁珠工位冲突判断函数

```
function [tcount] = cizhugongwei(matrix time,matrix x)
N = size(matrix time,1);
chongtu Matrix = [];
for i = 1:N
    chongtu Matrix = [chongtu Matrix F1(matrix time(i,5),matrix time(i,7))];
end
chongtu Matrix = [chongtu Matrix F1(matrix x(5),matrix x(7))];
[~, Y] = mode(chongtu Matrix);
if Y == 1
    tcount = 1; % 1代表没有冲突
else
    tcount = 0;
end
end
```

代码清单 10 问题 3 主函数

```
% ttttt = [];
% for iiiii = 1:5

warm growth A1 = 600;
warm growth A2 = 300;
warm growth B1 = 2400;
warm growth B2 = 1200;
warm growth C1 = 1500;
warm growth C2 = 600;

M = 60; % A芯片数量
N = 50; % B芯片数量
G = 50; % C芯片数量
B = 22; % 磁珠工位时间
Wash = 325; % 清洗时间

% 初始的时间矩阵有三种，分别考虑A第一个、B第一个和C第一个
% 假设A为第一个
% matrix time = [0 150 156 156+warm growth A1 156+warm growth A1+8 156+warm growth A1+8+21 156+warm growth A1+8+21+8 156+warm growth A1+8+21+8+warm growth A2 156+warm growth A1+8+21+8+warm growth A2+16 156+warm growth A1+8+21+8+warm growth A2+16+Wash 156+warm growth A1+8+21+8+warm growth A2+16+Wash+12 156+warm growth A1+8+21+8+warm growth A2+16+Wash+12+25 1];
% 假设B为第一个
matrix time = [0 150 156 156+warm growth B1 156+warm growth B1+8 156+warm growth B1+8+21 156+warm growth B1+8+21+8 156+warm growth B1+8+21+8+warm growth B2 156+warm growth B1+8+21+8+warm growth B2+16 156+warm growth B1+8+21+8+warm growth B2+16+Wash 156+warm growth B1+8+21+8+warm growth B2+16+Wash+12 156+warm growth B1+8+21+8+warm growth B2+16+Wash+12+25 2];
% 假设C为第一个
% matrix time = [0 150 156 156+warm growth C1 156+warm growth C1+8 156+warm growth C1+8+21 156+warm growth C1+8+21+8 156+warm growth C1+8+21+8+warm growth C2 156+warm growth C1+8+21+8+warm growth C2+16 156+warm growth C1+8+21+8+warm growth C2+16+Wash 156+warm growth C1+8+21+8+warm growth C2+16+Wash+12 156+warm growth C1+8+21+8+warm growth C2+16+Wash+12+25 3];

ti = 0;
k1 = 0; k2 = 1; k3 = 0; % A、B、C芯片的加入个数
% 遍历所有芯片
for i = 1:M+N+G-1
    ti = ti + 36;
    if k1 < M && k2 < N && k3 < G
        while 1
            % 写出两个芯片对应的新时间节点矩阵
            matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+warm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12 ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25];
            matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+21 ti+156+warm growth B1+8+21+8 ti+156+warm growth B1+8+21+8+warm growth B2 ti+156+warm growth B1+8+21+8+warm growth B2+16 ti+156+warm growth B1+8+21+8+warm growth B2+16+Wash ti+156+warm growth B1+8+21+8+warm growth B2+16+Wash+12 ti+156+warm growth B1+8+21+8+warm growth B2+16+Wash+12+25];
            matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+21 ti+156+warm growth C1+8+21+8 ti+156+warm growth C1+8+21+8+warm growth C2 ti+156+warm growth C1+8+21+8+warm growth C2+16 ti+156+warm growth C1+8+21+8+warm growth C2+16+Wash ti+156+warm growth C1+8+21+8+warm growth C2+16+Wash+12 ti+156+warm growth C1+8+21+8+warm growth C2+16+Wash+12+25];
            [tcount, Y] = cizhugongwei(matrix x A, matrix x B);
            [tcount, Y] = cizhugongwei(matrix x A, matrix x C);
            [tcount, Y] = cizhugongwei(matrix x B, matrix x C);
            if Y == 1
                break;
            end
        end
        k1 = k1 + 1;
        k2 = k2 + 1;
        k3 = k3 + 1;
    end
end
```

```

+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
% 分别计算当前ti下A、B、C的冲突参数
% 计算当前ti下A的冲突参数
% 计算转轮是否冲突
[YA] = zhuanpan(matrix time,matrix x A);
% 计算磁珠是否冲突
[XA] = cizhugongwei(matrix time,matrix x A);
% 计算检测工位是否冲突
[ZA] = jiancegongwei(matrix time,matrix x A);
% 计算温育盘是否冲突
[TA] = wenyushi(matrix time,matrix x A);
% 计算清洗工位是否冲突
[PA] = qingxihe(matrix time,matrix x A);

% 计算当前ti下B的冲突参数
% 计算转轮是否冲突
[YB] = zhuanpan(matrix time,matrix x B);
% 计算磁珠是否冲突
[XB] = cizhugongwei(matrix time,matrix x B);
% 计算检测工位是否冲突
[ZB] = jiancegongwei(matrix time,matrix x B);
% 计算温育盘是否冲突
[TB] = wenyushi(matrix time,matrix x B);
% 计算清洗工位是否冲突
[PB] = qingxihe(matrix time,matrix x B);

% 计算当前ti下C的冲突参数
% 计算转轮是否冲突
[YC] = zhuanpan(matrix time,matrix x C);
% 计算磁珠是否冲突
[XC] = cizhugongwei(matrix time,matrix x C);
% 计算检测工位是否冲突
[ZC] = jiancegongwei(matrix time,matrix x C);
% 计算温育盘是否冲突
[TC] = wenyushi(matrix time,matrix x C);
% 计算清洗工位是否冲突
[PC] = qingxihe(matrix time,matrix x C);

% 考虑数量约束, 判断添加A、B还是C
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
matrix time = [matrix time; matrix x A];
k1 = k1 + 1;
break
end
if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
matrix time = [matrix time; matrix x C];
k3 = k3 + 1;
break
end
if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
matrix time = [matrix time; matrix x B];
k2 = k2 + 1;
break
else
ti = ti + 1;
end
end
elseif k1 == M && k2 < N && k3 < G

```

```

while 1
    % 写出两个芯片对应的新时间节点矩阵
    matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
    matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
    % 分别计算当前ti下B、C的冲突参数

    % 计算当前ti下B的冲突参数
    % 计算转轮是否冲突
    [YB] = zhuanpan(matrix time,matrix x B);
    % 计算磁珠是否冲突
    [XB] = cizhugongwei(matrix time,matrix x B);
    % 计算检测工位是否冲突
    [ZB] = jiancegongwei(matrix time,matrix x B);
    % 计算温育盘是否冲突
    [TB] = wenyushi(matrix time,matrix x B);
    % 计算清洗工位是否冲突
    [PB] = qingxihe(matrix time,matrix x B);

    % 计算当前ti下C的冲突参数
    % 计算转轮是否冲突
    [YC] = zhuanpan(matrix time,matrix x C);
    % 计算磁珠是否冲突
    [XC] = cizhugongwei(matrix time,matrix x C);
    % 计算检测工位是否冲突
    [ZC] = jiancegongwei(matrix time,matrix x C);
    % 计算温育盘是否冲突
    [TC] = wenyushi(matrix time,matrix x C);
    % 计算清洗工位是否冲突
    [PC] = qingxihe(matrix time,matrix x C);

    % 考虑数量约束, 判断添加B还是C
    if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
        matrix time = [matrix time; matrix x C];
        k3 = k3 + 1;
        break
    end
    if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
        matrix time = [matrix time; matrix x B];
        k2 = k2 + 1;
        break
    else
        ti = ti + 1;
    end
end
elseif k1 < M && k2 == N && k3 < G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
        matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
        % 分别计算当前ti下A、C的冲突参数
        % 计算当前ti下A的冲突参数
        % 计算转轮是否冲突
        [YA] = zhuanpan(matrix time,matrix x A);

```

```

% 计算磁珠是否冲突
[XA] = cizhugongwei(matrix time,matrix x A);
% 计算检测工位是否冲突
[ZA] = jiancegongwei(matrix time,matrix x A);
% 计算温育盘是否冲突
[TA] = wenyushi(matrix time,matrix x A);
% 计算清洗工位是否冲突
[PA] = qingxihe(matrix time,matrix x A);

% 计算当前ti下C的冲突参数
% 计算转轮是否冲突
[YC] = zhuanpan(matrix time,matrix x C);
% 计算磁珠是否冲突
[XC] = cizhugongwei(matrix time,matrix x C);
% 计算检测工位是否冲突
[ZC] = jiancegongwei(matrix time,matrix x C);
% 计算温育盘是否冲突
[TC] = wenyushi(matrix time,matrix x C);
% 计算清洗工位是否冲突
[PC] = qingxihe(matrix time,matrix x C);

% 考虑数量约束, 判断添加A、B还是C
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
    matrix time = [matrix time; matrix x A];
    k1 = k1 + 1;
    break
end
if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
    matrix time = [matrix time; matrix x C];
    k3 = k3 + 1;
    break
else
    ti = ti + 1;
end
end
elseif k1 < M && k2 < N && k3 == G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
        matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
        % 分别计算当前ti下A、B的冲突参数
        % 计算当前ti下A的冲突参数
        % 计算转轮是否冲突
        [YA] = zhuanpan(matrix time,matrix x A);
        % 计算磁珠是否冲突
        [XA] = cizhugongwei(matrix time,matrix x A);
        % 计算检测工位是否冲突
        [ZA] = jiancegongwei(matrix time,matrix x A);
        % 计算温育盘是否冲突
        [TA] = wenyushi(matrix time,matrix x A);
        % 计算清洗工位是否冲突
        [PA] = qingxihe(matrix time,matrix x A);

        % 计算当前ti下B的冲突参数
        % 计算转轮是否冲突
        [YB] = zhuanpan(matrix time,matrix x B);
        % 计算磁珠是否冲突
        [XB] = cizhugongwei(matrix time,matrix x B);
        % 计算检测工位是否冲突
        [ZB] = jiancegongwei(matrix time,matrix x B);
        % 计算温育盘是否冲突
    
```

```

[TB] = wenyushi(matrix time,matrix x B);
% 计算清洗工位是否冲突
[PB] = qingxihe(matrix time,matrix x B);

% 考虑数量约束, 判断添加A还是B
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
    matrix time = [matrix time; matrix x A];
    k1 = k1 + 1;
    break
end
if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
    matrix time = [matrix time; matrix x B];
    k2 = k2 + 1;
    break
else
    ti = ti + 1;
end
end
elseif k1 == M && k2 == N && k3 < G
    while 1
        % 写出C芯片对应的新时间节点矩阵
        matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
        % 分别计算当前ti下C的冲突参数
        % 计算转轮是否冲突
        [YC] = zhuanpan(matrix time,matrix x C);
        % 计算磁珠是否冲突
        [XC] = cizhugongwei(matrix time,matrix x C);
        % 计算检测工位是否冲突
        [ZC] = jiancegongwei(matrix time,matrix x C);
        % 计算温育盘是否冲突
        [TC] = wenyushi(matrix time,matrix x C);
        % 计算清洗工位是否冲突
        [PC] = qingxihe(matrix time,matrix x C);

        % 考虑数量约束, 判断是否添加C

        if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
            matrix time = [matrix time; matrix x C];
            k3 = k3 + 1;
            break
        else
            ti = ti + 1;
        end
    end
elseif k1 == M && k2 < N && k3 == G
    while 1
        % 写出B芯片对应的新时间节点矩阵
        matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
        % 计算当前ti下B的冲突参数
        % 计算转轮是否冲突
        [YB] = zhuanpan(matrix time,matrix x B);
        % 计算磁珠是否冲突
        [XB] = cizhugongwei(matrix time,matrix x B);
        % 计算检测工位是否冲突
        [ZB] = jiancegongwei(matrix time,matrix x B);
        % 计算温育盘是否冲突
        [TB] = wenyushi(matrix time,matrix x B);
        % 计算清洗工位是否冲突
        [PB] = qingxihe(matrix time,matrix x B);

        % 考虑数量约束, 判断是否添加B

```

```

        if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
            matrix time = [matrix time; matrix x B];
            k2 = k2 + 1;
            break
        else
            ti = ti + 1;
        end
    end
elseif k1 < M && k2 == N && k3 == G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
        % 计算当前ti下A的冲突参数
        % 计算转轮是否冲突
        [YA] = zhuanpan(matrix time,matrix x A);
        % 计算磁珠是否冲突
        [XA] = cizhugongwei(matrix time,matrix x A);
        % 计算检测工位是否冲突
        [ZA] = jiancegongwei(matrix time,matrix x A);
        % 计算温育盘是否冲突
        [TA] = wenyushi(matrix time,matrix x A);
        % 计算清洗工位是否冲突
        [PA] = qingxihe(matrix time,matrix x A);

        % 考虑数量约束, 判断是否添加A
        if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
            matrix time = [matrix time; matrix x A];
            k1 = k1 + 1;
            break
        else
            ti = ti + 1;
        end
    end
else
    break
end
end
% ttttt = [ttttt max(matrix time(:,12))];
% end
% plot(ttttt)
% xlswrite('表4',matrix time,'Sheet1')

```

代码清单 11 问题 5 主函数

```

% ttttt = [];
% for iiii = 1:5

warm growth A1 = 600;
warm growth A2 = 300;
warm growth B1 = 2400;
warm growth B2 = 1200;
warm growth C1 = 1500;
warm growth C2 = 600;

M = 60; % A芯片数量
N = 70; % B芯片数量
G = 20; % C芯片数量
B = 24; % 磁珠工位时间
Wash = 325; % 清洗时间

matrix time = xlsread("问5芯片AB24中间输出结果.xls");
HHH = size(matrix time,1);
matrix time = matrix time(1:HHH-1,:);

```

```

% ti = 1764;
ti = matrix time(HHH-1,1);
k1 = 25; k2 = 24; k3 = 0; % A、B、C芯片的加入个数
% 遍历所有芯片
for i = 1:M+N+G-1
    ti = ti + 36;
    if k1 < M && k2 < N && k3 < G
        while 1
            % 写出两个芯片对应的新时间节点矩阵
            matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
            matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
            matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
            % 分别计算当前ti下A、B、C的冲突参数
            % 计算当前ti下A的冲突参数
            % 计算转轮是否冲突
            [YA] = zhuanpan(matrix time,matrix x A);
            % 计算磁珠是否冲突
            [XA] = cizhugongwei(matrix time,matrix x A);
            % 计算检测工位是否冲突
            [ZA] = jiancegongwei(matrix time,matrix x A);
            % 计算温育盘是否冲突
            [TA] = wenyushi(matrix time,matrix x A);
            % 计算清洗工位是否冲突
            [PA] = qingxihe(matrix time,matrix x A);

            % 计算当前ti下B的冲突参数
            % 计算转轮是否冲突
            [YB] = zhuanpan(matrix time,matrix x B);
            % 计算磁珠是否冲突
            [XB] = cizhugongwei(matrix time,matrix x B);
            % 计算检测工位是否冲突
            [ZB] = jiancegongwei(matrix time,matrix x B);
            % 计算温育盘是否冲突
            [TB] = wenyushi(matrix time,matrix x B);
            % 计算清洗工位是否冲突
            [PB] = qingxihe(matrix time,matrix x B);

            % 计算当前ti下C的冲突参数
            % 计算转轮是否冲突
            [YC] = zhuanpan(matrix time,matrix x C);
            % 计算磁珠是否冲突
            [XC] = cizhugongwei(matrix time,matrix x C);
            % 计算检测工位是否冲突
            [ZC] = jiancegongwei(matrix time,matrix x C);
            % 计算温育盘是否冲突
            [TC] = wenyushi(matrix time,matrix x C);
            % 计算清洗工位是否冲突
            [PC] = qingxihe(matrix time,matrix x C);

            % 考虑数量约束, 判断添加A、B还是C
            if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
                matrix time = [matrix time; matrix x A];
                k1 = k1 + 1;
                break
            end
            if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
                matrix time = [matrix time; matrix x C];
            end
        end
    end
end

```

```

        k3 = k3 + 1;
        break
    end
    if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
        matrix time = [matrix time; matrix x B];
        k2 = k2 + 1;
        break
    else
        ti = ti + 1;
    end
end
elseif k1 == M && k2 < N && k3 < G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
        matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
        % 分别计算当前ti下B、C的冲突参数

        % 计算当前ti下B的冲突参数
        % 计算转轮是否冲突
        [YB] = zhuanpan(matrix time,matrix x B);
        % 计算磁珠是否冲突
        [XB] = cizhugongwei(matrix time,matrix x B);
        % 计算检测工位是否冲突
        [ZB] = jiancegongwei(matrix time,matrix x B);
        % 计算温育盘是否冲突
        [TB] = wenyushi(matrix time,matrix x B);
        % 计算清洗工位是否冲突
        [PB] = qingxihe(matrix time,matrix x B);

        % 计算当前ti下C的冲突参数
        % 计算转轮是否冲突
        [YC] = zhuanpan(matrix time,matrix x C);
        % 计算磁珠是否冲突
        [XC] = cizhugongwei(matrix time,matrix x C);
        % 计算检测工位是否冲突
        [ZC] = jiancegongwei(matrix time,matrix x C);
        % 计算温育盘是否冲突
        [TC] = wenyushi(matrix time,matrix x C);
        % 计算清洗工位是否冲突
        [PC] = qingxihe(matrix time,matrix x C);

        % 考虑数量约束, 判断添加B还是C
        if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
            matrix time = [matrix time; matrix x C];
            k3 = k3 + 1;
            break
        end
        if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
            matrix time = [matrix time; matrix x B];
            k2 = k2 + 1;
            break
        else
            ti = ti + 1;
        end
    end
elseif k1 < M && k2 == N && k3 < G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8

```

```

+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+warm growth C2+16 ti+156+warm growth C1+8+
B+8+warm growth C2+16+Wash ti+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+warm growth C2+16+Wash+12+25 3];
% 分别计算当前ti下A、C的冲突参数
% 计算当前ti下A的冲突参数
% 计算转轮是否冲突
[YA] = zhuanpan(matrix time,matrix x A);
% 计算磁珠是否冲突
[XA] = cizhugongwei(matrix time,matrix x A);
% 计算检测工位是否冲突
[ZA] = jiancegongwei(matrix time,matrix x A);
% 计算温育盘是否冲突
[TA] = wenyushi(matrix time,matrix x A);
% 计算清洗工位是否冲突
[PA] = qingxihe(matrix time,matrix x A);

% 计算当前ti下C的冲突参数
% 计算转轮是否冲突
[YC] = zhuanpan(matrix time,matrix x C);
% 计算磁珠是否冲突
[XC] = cizhugongwei(matrix time,matrix x C);
% 计算检测工位是否冲突
[ZC] = jiancegongwei(matrix time,matrix x C);
% 计算温育盘是否冲突
[TC] = wenyushi(matrix time,matrix x C);
% 计算清洗工位是否冲突
[PC] = qingxihe(matrix time,matrix x C);

% 考虑数量约束, 判断添加A、B还是C
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
    matrix time = [matrix time; matrix x A];
    k1 = k1 + 1;
    break
end
if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
    matrix time = [matrix time; matrix x C];
    k3 = k3 + 1;
    break
else
    ti = ti + 1;
end
end
elseif k1 < M && k2 < N && k3 == G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
        matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+warm growth B2+16 ti+156+warm growth B1+8+
B+8+warm growth B2+16+Wash ti+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+warm growth B2+16+Wash+12+25 2];
        % 分别计算当前ti下A、B的冲突参数
        % 计算当前ti下A的冲突参数
        % 计算转轮是否冲突
        [YA] = zhuanpan(matrix time,matrix x A);
        % 计算磁珠是否冲突
        [XA] = cizhugongwei(matrix time,matrix x A);
        % 计算检测工位是否冲突
        [ZA] = jiancegongwei(matrix time,matrix x A);
        % 计算温育盘是否冲突
        [TA] = wenyushi(matrix time,matrix x A);

```

```

% 计算清洗工位是否冲突
[PA] = qingxihe(matrix time,matrix x A);

% 计算当前ti下B的冲突参数
% 计算转轮是否冲突
[YB] = zhuanpan(matrix time,matrix x B);
% 计算磁珠是否冲突
[XB] = cizhugongwei(matrix time,matrix x B);
% 计算检测工位是否冲突
[ZB] = jiancegongwei(matrix time,matrix x B);
% 计算温育盘是否冲突
[TB] = wenyushi(matrix time,matrix x B);
% 计算清洗工位是否冲突
[PB] = qingxihe(matrix time,matrix x B);

% 考虑数量约束, 判断添加A还是B
if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
    matrix time = [matrix time; matrix x A];
    k1 = k1 + 1;
    break
end
if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
    matrix time = [matrix time; matrix x B];
    k2 = k2 + 1;
    break
else
    ti = ti + 1;
end
end
elseif k1 == M && k2 == N && k3 < G
    while 1
        % 写出C芯片对应的新时间节点矩阵
        matrix x C = [ti ti+150 ti+156 ti+156+warm growth C1 ti+156+warm growth C1+8
ti+156+warm growth C1+8+B ti+156+warm growth C1+8+B+8 ti+156+warm growth C1+8+B+8+w
arm growth C2 ti+156+warm growth C1+8+B+8+arm growth C2+16 ti+156+warm growth C1+8+
B+8+arm growth C2+16+Wash ti+156+warm growth C1+8+B+8+arm growth C2+16+Wash+12 t
i+156+warm growth C1+8+B+8+arm growth C2+16+Wash+12+25 3];
        % 分别计算当前ti下C的冲突参数
        % 计算转轮是否冲突
        [YC] = zhuanpan(matrix time,matrix x C);
        % 计算磁珠是否冲突
        [XC] = cizhugongwei(matrix time,matrix x C);
        % 计算检测工位是否冲突
        [ZC] = jiancegongwei(matrix time,matrix x C);
        % 计算温育盘是否冲突
        [TC] = wenyushi(matrix time,matrix x C);
        % 计算清洗工位是否冲突
        [PC] = qingxihe(matrix time,matrix x C);

        % 考虑数量约束, 判断是否添加C

        if YC == 1 && XC == 1 && ZC == 1 && TC == 1 && PC == 1
            matrix time = [matrix time; matrix x C];
            k3 = k3 + 1;
            break
        else
            ti = ti + 1;
        end
    end
end
elseif k1 == M && k2 < N && k3 == G
    while 1
        % 写出B芯片对应的新时间节点矩阵
        matrix x B = [ti ti+150 ti+156 ti+156+warm growth B1 ti+156+warm growth B1+8
ti+156+warm growth B1+8+B ti+156+warm growth B1+8+B+8 ti+156+warm growth B1+8+B+8+w
arm growth B2 ti+156+warm growth B1+8+B+8+arm growth B2+16 ti+156+warm growth B1+8+
B+8+arm growth B2+16+Wash ti+156+warm growth B1+8+B+8+arm growth B2+16+Wash+12 t
i+156+warm growth B1+8+B+8+arm growth B2+16+Wash+12+25 2];
        % 计算当前ti下B的冲突参数
        % 计算转轮是否冲突

```

```

[YB] = zhuanpan(matrix time,matrix x B);
% 计算磁珠是否冲突
[XB] = cizhugongwei(matrix time,matrix x B);
% 计算检测工位是否冲突
[ZB] = jiancegongwei(matrix time,matrix x B);
% 计算温育盘是否冲突
[TB] = wenyushi(matrix time,matrix x B);
% 计算清洗工位是否冲突
[PB] = qingxihe(matrix time,matrix x B);

% 考虑数量约束, 判断是否添加B
if YB == 1 && XB == 1 && ZB == 1 && TB == 1 && PB == 1
    matrix time = [matrix time; matrix x B];
    k2 = k2 + 1;
    break
else
    ti = ti + 1;
end
end
elseif k1 < M && k2 == N && k3 == G
    while 1
        % 写出两个芯片对应的新时间节点矩阵
        matrix x A = [ti ti+150 ti+156 ti+156+warm growth A1 ti+156+warm growth A1+8
ti+156+warm growth A1+8+B ti+156+warm growth A1+8+B+8 ti+156+warm growth A1+8+B+8+w
arm growth A2 ti+156+warm growth A1+8+B+8+warm growth A2+16 ti+156+warm growth A1+8
+B+8+warm growth A2+16+Wash ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12
ti+156+warm growth A1+8+B+8+warm growth A2+16+Wash+12+25 1];
        % 计算当前ti下A的冲突参数
        % 计算转轮是否冲突
        [YA] = zhuanpan(matrix time,matrix x A);
        % 计算磁珠是否冲突
        [XA] = cizhugongwei(matrix time,matrix x A);
        % 计算检测工位是否冲突
        [ZA] = jiancegongwei(matrix time,matrix x A);
        % 计算温育盘是否冲突
        [TA] = wenyushi(matrix time,matrix x A);
        % 计算清洗工位是否冲突
        [PA] = qingxihe(matrix time,matrix x A);

        % 考虑数量约束, 判断是否添加A
        if YA == 1 && XA == 1 && ZA == 1 && TA == 1 && PA == 1
            matrix time = [matrix time; matrix x A];
            k1 = k1 + 1;
            break
        else
            ti = ti + 1;
        end
    end
end
else
    break
end
end
% tttttt = [ttttt max(matrix time(:,12'))];
% end
% plot(tttttt)
% % xlswrite('问1芯片B',matrix time,'Sheet1')

```