

基于改进遗传算法的化学发光免疫分析仪运行的调度优化

摘要

随着医学分析技术的快速发展进步,化学发光免疫分析仪在人体体液样本检测中发挥了重要作用,目前广泛应用于国内外的各大医院和医疗机构,用于临床检测人体血清、唾液等,实现对蛋白、核酸等指标的精准检测。本文通过仿真该分析仪全自动检测体液的前处理、第一次温育、磁珠加样、第二次温育、清洗和检测等六道重要工序,并建立数学模型和相应算法,给出了分析仪运行的最优调度方案。

针对问题一:在处理 A 和 B 芯片在 1 小时单位时间内最多能够完成的检测数量问题时,本文严格按照分析仪在前处理、转盘对位、第一次温育、转盘对位、磁珠加样、第二次温育、转盘对位、清洗、检测对位和检测等各工序所需时长,仿真分析仪运行过程。并且充分考虑转盘转动时间和各工序处理时间重叠的可能性,考虑分析仪卡槽数量、清洗盒数量、加样盒数量等设备因素,最终仿真结果为 A 芯片在 60 分钟内最多可以完成 88 块芯片的检测, B 芯片在 60 分钟内最多可以完成 71 块芯片的检测。

针对问题二和问题三:对于多种类型的芯片调度进行分析。该问题类似于混合流水车间调度问题,但其区别在于:多个芯片在检测是其相同的工序安排顺序具有强约束、一个芯片在同一处理模块上不止操作一次、一个芯片的某些工序可以选择多个处理模块进行加工(不同的温育卡槽和不同的清洗盒)。在考虑到上述区别的情况下,将该类问题转化为带有强时刻顺序约束及可折返加工的多选择混合流水车间调度优化问题。随后基于决策论和动态规划的思想,在除温育时间外其它工序时间既定的条件下,以最小化仪器最终的检测结束时刻为目标构建了带有强时刻顺序约束的多选择最优作业时间模型,并结合经过启发式改进的遗传算法对模型进行求解,求解结果如下:

- **问题 2 结果:** 优先安排 A 类芯片检测,再安排 B 类芯片检测,最优检测时间为 **11059s**;
- **问题 3 结果:** 按照 A、C、B 的顺序安排芯片检测,最优检测时间为 **10980s**;

针对问题四:本文对求解该问题的算法进行了推广验证,总结归纳了单种芯片排产的最小耗时递推公式,并设计了区别于题目设定的新算例来检验算法。

针对问题五:对于分批次提供待检测芯片的调度问题进行分析。该问题的解决是在问题二、三的基础上,进一步考虑不同批次的芯片的进仓时刻约束,即 **C 芯片的进仓时刻需不小于 1800s**。考虑到该约束,对已构建的算法进行微调然后求解,结果为 **11029s**。

在灵敏度分析部分,本文调整加磁时间和清洗盒清洗时间观察了问题 2、问题 3 和问题 5 的最优检测耗时变动情况,发现加磁时间在小范围内变动时对最优结果影响不大,而清洗盒清洗时间的小范围变动会很大程度上影响最终的总耗时。

最后,本文对提出的模型进行了综合评价:带有启发式改进的遗传算法能合理解决仪器检测的芯片调度问题,具有实用性强,算法在小规模芯片检测和处理上效率很高,并且在应对分批次处理芯片检测问题也仍然有效。

关键词: 分析仪运行调度 混合流水车间调度 启发式构造 遗传算法

目录

1 问题综述	1
1.1 问题背景	1
1.2 问题提出	1
1.3 资料条件	2
2 模型假设与符号说明	2
2.1 模型基本假设	2
2.2 符号说明	2
3 问题分析与模型建立	3
3.1 问题 1 分析与模型构建	3
3.2 问题 2 与问题 3: 分析与模型求解	4
3.2.1 问题 2 与问题 3: 多种芯片组合检测问题	4
3.2.2 模型构建	5
3.2.3 改进的遗传算法设计	6
3.2.4 问题 2 求解结果	12
3.2.5 问题 3 求解结果	13
3.3 问题 4: 分析与模型推广检验	14
3.3.1 问题 4: 模型推广问题	14
3.3.2 模型推广检验	15
3.4 问题 5: 分析与模型建立	15
3.4.1 问题 5: 两阶段多种芯片组合检测问题	15
3.4.2 改进的遗传算法调整	16
3.4.3 问题 5 求解结果	16
4 模型灵敏度检验	18
5 模型评价与推广	20
5.1 模型的优点	20
5.2 模型的不足	20
5.3 模型的推广	20
参考文献	21
附 录	22
附录 A: 问题 1 完整结果	22
附录 B: 问题 2 完整结果	28
附录 C: 问题 3 完整结果	32
附录 D: 问题 4 算例求解完整结果	38
附录 E: 问题 5 完整结果	46
附录 F: 支撑材料列表	50
附录 G: 主要程序/关键代码	50

1 问题综述

1.1 问题背景

体外诊断是一种通过化验人体体液及血液等组织进行疾病诊断的方法。该方法无需对人体本身进行检测，具有方法简便、降低对人体损害、能在疾病早期进行快速准确判定等优点^[1]。随着高新技术和检验医学的发展，全自动化学发光免疫分析仪作为医学上对人体体液样本进行检验的仪器，目前已被国内外各大医院和医疗机构广泛用于临床。该仪器使医学上免疫诊断的整个流程通过机器实现全自动化操作，从而克服了传统人工检测存在的效率低、结果精确度差、耗时费力等缺点。

化学发光免疫分析仪运行的调度优化问题，是指对一种或多种待检测的芯片，在保证各检测工序不发生冲突的前提下，以最小化最大检测时间为目标进行芯片检测。面对多任务项目检测任务，与经典的车间流水调度问题不同，免疫分析仪更多的具有可重入特点^[2]。因此，如何对多个功能子模块进行优化调度，使各个子模块高效配合，进一步提高设备的灵敏度和高效性是当前亟需解决的问题。

1.2 问题提出

化学发光免疫分析仪运行的调度优化问题，涉及前处理、第一次温育、磁珠加样、第二次温育、清洗和检测等六道工序。每道工序都有固定的处理工位，也需要一定的时间，并且由于检测需要，每片芯片的检测流程一旦开始便不能停止，直到整个检测过程全部完成。

该分析仪运行调度问题，为了解决一组待检测的芯片，按照上述的六个步骤进行处理时，如何分配芯片的检测顺序、检测时间，并且全局考虑、综合考量各道工序不会发送冲突的前提下，实现测量组的芯片总的完成时间尽可能的小。

为了给出免疫分析仪运行的最优调度方案，本文按照题目每道问题要求，逐步建立数学模型，并给出相应的算法，求解每片芯片在每道工序的开始时间和结束时间，并通盘考虑芯片进仓的顺序。具体待解决问题有如下 5 点：

(1) 问题 1：已知 A 类芯片，其第一次温育时间 10 分钟，第二次温育时间 5 分钟，则 1 小时的单位时间内，最多可以完成多芯片的检测？如果换成 B 芯片，其第一次温育时间为 40 分钟，第二次温育时间为 20 分钟，在单位时间内最多可以完成多少芯片检测？

(1) 问题 1：现有一组 A 类型芯片，其第一次温育时间 10 分钟，第二次温育时间 5 分钟，则求解单位时间（1 小时）内最多可以完成芯片检测的数量。随后，换成一组 B 类型芯片，其第一次温育时间 40 分钟，第二次温育时间 20 分钟，进行求解。

(2) 问题 2：如果有 A，B 两种类型的芯片各 80 片，求解它们全部检测完至少需要的时间。

(3) 问题 3：若还有 C 型芯片，其第一次温育时间 25 分钟，第二次温育时间 10 分钟。则 A 型芯片 60 片、B 型芯片 50 片、C 型芯片 50 片，求解全部检测完至少需要的时间。

(4) 问题 4：针对问题 2 和问题 3，将本文提出的模型和算法，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。最后，进一步推广到 N 种类型的芯片并举例验证。

(5) 问题 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，求解全部检测完至少需要多少时间。

1.3 资料条件

由题目已知条件，已有的设备（包括工作台、转盘、卡槽、清洗盒、磁珠加样盒以及检测盒等）数目，和每道工序所需的处理时间，经过整理后，如下表所示。

表 1 资料条件

处理步骤	仪器设备	设备数量	处理时间		
			建议处理时间 (单位/s)	A 芯片处理时间 (单位/s)	B 芯片处理时间 (单位/s)
前处理	前处理工作台	5 片	150	150	150
第一次温育	大转盘	40 个卡槽	0~3600	600	2400
磁珠加样	磁珠加样盒	1 个	[21,25]	21	21
第二次温育	大转盘	40 个卡槽	0~1800	300	1200
清洗	清洗盒	8 个	≥ 325	325	325
检测	检测盘	1 个	25	25	25

此外，化学发光免疫分析仪对人体体液样本进行检测的全部过程中，还包括转盘对位等时间，具体将整个流程表示如下：

前处理->转盘转动对位->第一次温育->转盘对位磁珠盘->磁珠工位->转盘转动对位->第二次温育->转盘对位清洗盒->清洗工位->检测盘对位->检测工位。

整个处理过程，细分为 11 个步骤，在后续问题求解调度优化问题时，本文将严格按照该处理流程仔细记录每个步骤结束后的处理时间变化。

2 模型假设与符号说明

2.1 模型基本假设

- (1) 假设仪器的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；
- (2) 假设磁珠工位上加磁时间为 21s，清洗工位上清洗时间为 325s；
- (3) 假设仪器的转盘或检测盘满载时也能够完成转动对位，即完成对位的芯片在转盘或检测盘一有空闲时就可以即可补位；
- (4) 假定前处理台有 5 个芯片槽，分别对应不同的前处理台步骤，内槽串联。
- (5) 在进行问题 4 的解决时，假定同一类型的第一次温育所需的时间不小于第二次温育所需的时间。

2.2 符号说明

本文定义了如下重要符号，其余符号在使用时注明。

表 2 符号说明

符号	含义	单位	取值
----	----	----	----

符号	含义	单位	取值
G	优化的目标函数	无	
M	仪器的处理模块个数	个	55
N_A	A 芯片的数量	个	≥ 0
N_B	B 芯片的数量	个	≥ 0
N_C	C 芯片的数量	个	≥ 0
N	芯片总数	片	所有芯片数量之和
P	芯片处理步骤数	步	10
P_{sum}	芯片总处理步骤	步	$P \times N$
(i, j)	第 i 个芯片的第 j 道工序	无	$i \in (1, N), j \in (1, P)$
E_m	第 m 个模块的作业结束时间	无	$m \in (1, M), E_m \geq 0$
S_{ij}	第 i 个芯片的第 j 道工序的开始时间	秒	$i \in (1, N), j \in (1, P),$ $S_{ij} \geq 0$
T_{ij}	第 i 个芯片的第 j 道工序的消耗时间	秒	$i \in (1, N), j \in (1, P),$ $T_{ij} \geq 0$
L_{ijm}	(i, j) 能否在第 m 个模块上加工	无	$\{0, 1\}$

3 问题分析与模型建立

3.1 问题 1 分析与模型构建

题目以化学发光免疫分析仪分析载有人体体液的芯片为背景, 介绍了化学发光免疫分析仪的设备构造和 workflow, 问题一要求我们根据已知的 A 和 B 芯片的第一次温育和第二次温育的时间, 以及化学发光免疫分析仪每个处理阶段的处理时间, 计算出在 1 小时单位时间内最多可以完成多少芯片的检测。

对于 A 芯片, 第一次温育时间为 10 分钟, 第二次温育时间为 5 分钟; 对于 B 芯片第一次温育时间为 40 分钟, 第二次温育时间为 20 分钟。在下表中列出 A 和 B 芯片在各个阶段的处理时间如下:

表 3 表 A 和 B 芯片在化学发光免疫分析仪各个阶段的处理时间 (单位/s)

芯片代 号	前 处 理	转盘 对位	第一次 温育	转盘 对位	磁珠 加样	转盘 对位	第二次 温育	转盘 对位	清 洗	转盘 对位	检测
A 芯片	50	6	600	8	21	8	300	16	25	12	5
B 芯片	50	6	2400	8	21	8	1200	16	25	12	5

在上述时间处理过程中, 由于问题一题目要求计算最多可以完成多少芯片的检测, 因此本文在考虑磁珠加样时间时, 取[21s,25s]时间区间内的最小值, 即本文按照磁珠加样时间为 21s 进行计算; 考虑在清洗盒的清洗时时, 取 $\geq 325s$ 的最小值 325s 作为清洗时间进行计算。

对于 A 芯片，我们首先按照第一块芯片的处理流程给出各个阶段的处理时间，由于在前处理阶段芯片依次经过进仓读码、激光超声、过滤、定量、R1R2 加样处理，并且由于芯槽位有限，每 30s 只能处理一块芯片，因此第二片芯片的进入时间将延后 30s，依次对于后续芯片也是如此。

本文模拟了 10000 块 A 芯片的处理流程，各阶段的处理结果如附件 x 所示。经过仔细计算每个处理阶段的设备数量和每个阶段结束后芯片在卡槽的处理时间是否重叠，本文发现在前 8 个阶段都不存在设备不足或芯片间处理时间冲突的问题。在清洗阶段，由于仅有 8 个清洗盒，并且每块芯片的清洗时间过长为 325s，导致存在芯片在上一阶段处理完成后，需要继续等待的情况。考虑到该细节后，本文采取在清洗阶段以 8 块芯片为一组的办法解决处理时间冲突的问题。

以第一、二、三组为例，第一组的 8 块芯片中最早结束清洗阶段的芯片结束时间是 1434s，然而当 8 个清洗位全部占满后，第 9 块芯片在 1349s 已经完成了前面 8 个步骤，此该块需要等到 1434s 时，才能够进入清洗盘开始工作。对于第二组的剩余芯片，等到第一组的芯片清洗完成后，依次进入清洗盘开始工作。第二组的第一块芯片从清洗盘离开时间为 1759s，然后第三组的第一块芯片完成前 8 步骤的结束时间为 1589s，此时需要第三组芯片等待到 1759s 后，才能进入清洗盘工作，第三组的剩余芯片后续依次进入清洗盘工作。

对于 B 芯片，同 A 芯片的处理步骤进行实验。经过每个芯片的仿真处理发现，在第一次温育时，由于 B 芯片在大转盘槽位的处理时间过长（2400s），导致当大转盘的 40 个槽位全部占满时（此时为 1356s），第一块进入大转盘工作的芯片依然工作，当第一块芯片在大转盘温育结束是为 2556s，因此将第 41 块芯片进入大转盘的时间调整为 2556s，后续芯片的进入时间依次调整为前一组芯片的温育结束时间；

对于 B 芯片的清洗过程，同样存在 A 芯片的问题。B 芯片同样以 8 片为一组，对比前阶段结束时间和当前清洗盘盘位占用情况。若前阶段的工作结束时间 < 清洗盘第一块芯片清洗结束时间，则按照清洗盘中清洗芯片结束时间作为新一轮清洗的开始时间；反之，则按照前阶段的工作结束时间作为新一轮清洗的开始时间。

综上所述分析，通过仿真 A 芯片和 B 芯片在分析仪上的全部处理过程，本文将 A 芯片调度结果表示为附件 1 所示，B 芯片调度结果表示为附件 2 所示。

通过延长仿真运行时间，本文观察到 A 芯片在 35 段测量 60 分钟内检测芯片完成数量在 [54,88] 块内递增，之后稳定运行，得到 A 芯片在 60 分钟内最多可以完成 88 块芯片的检测；B 芯片在首个 60 分钟测量时间内，并不能完成芯片检测，当延长仿真运行时间后，计算 60 分钟内可以完成的芯片检测数量在 [48,71] 块区间内变化，出现该现象原因是 B 芯片温育时间过长，在一轮温育全部结束的一段 60 分钟统计时间内，统计得到的检测完成芯片数量最多，得到 B 芯片在 60 分钟内最多可以完成 71 块芯片的检测。

3.2 问题 2 与问题 3：分析与模型求解

3.2.1 问题 2 与问题 3：多种芯片组合检测问题

(1) 问题分析

由问题 2 和问题 3 可知，分别给定了 A、B 和 C 三种芯片的两个阶段的培育时间和待检测数量，在已知各检测子流程所需时间的条件下，最小化检测完全部芯片所需的总时间。该问题可以归结为类混合流水车间调度优化问题。

(2) 解决思路

该问题类似于混合流水车间调度问题：可以将检测仪器的每个处理模块看成不同的机器（ m 个），而芯片（ n 个）就是每个工件，每个芯片需要经过一定的工序才能完成检验，芯片的检测工序已知（不同类的芯片检测工序一致），每个检测工序所需的时间也已知（不同芯片的温培时间虽然不同，但也已知），因此该问题能够抽象成一个 m 对 n 的类混合流水车间调度问题。值得注意的是，该问题与传统混合流水车间调度问题的区别如下：

- **芯片检测的某些工序的安排顺序具有强约束：**传统的混合流水车间调度问题对于一个工件的每个工序只强调了其先后顺序，并不强调接续性，例如一个 J 工件包含三道工序 (j_1, j_2, j_3) ，一个 L 工件包含两道工序 (l_1, l_2) ，那么在进行作业时，可以将上述工序按照工序开始时间由小到大顺序进行如下安排 $(j_1, l_1, l_2, j_2, j_3)$ ，即只要保证对于每个工件而言，其所需工序在排产加工时的相对位次符合加工要求即可。而对于问题 2 而言，因为检测仪器的前处理室是一个整体，尽管在进行前处理时实际上包含了 5 道工序，并且推送进培育盘也需要时间（一个是对齐时间，一个是等待所需），也就是说由于检测仪器的特殊性，要求一个芯片在开始第一次培育时，其下一个芯片才能进行 R1R21 加样操作，这可以解释为：一个芯片 A 的工序 $(a_1, a_2, a_3, a_4, a_5, a_6, \dots, a_{10})$ ，一个芯片 B 的工序 $(b_1, b_2, b_3, b_4, b_5, b_6, \dots, b_{10})$ ，如果芯片 B 紧随 A 后进仓，那么在 $n \leq 6$ 时，要求 b_n 的顺序必须排在 a_n 的后面。
- **一个芯片在同一处理模块上不止操作一次：**传统的混合流水车间调度问题要求“每个工件只能在 1 台机器上加工 1 次”，而在问题 2 中，每检测一枚芯片需要经历两次温培工序，两次温育均在带有卡槽的温育盘上进行，这就打破了该约束。
- **一个芯片的某些工序可以选择多个处理模块进行加工：**传统的混合流水车间调度问题要求指定每个工件的每个工序的加工机器，而对于问题 2 而言，芯片在温培和清洗的时候有不同的卡槽以及不同的清洗盒可以选择。

在考虑了上述三个主要区别的情况下，本文采用改良的遗传算法对该问题进行求解，并且该方法具有很强的拓展性。

3.2.2 模型构建

依据 4.2.1 节问题分析部分，问题 2 与问题 3 都可以抽象描述如下：

一个化学发光免疫分析仪有 M （常量）个模块，要求检测 N 个芯片，其中，第 i 个芯片包含的工序数为 P （常量），则 $P_{sum} = P \times N$ 芯片检测任务集的总工序数。其中，各工序的时间耗费已确定（见表 xx 所示），并且每个芯片必须按照工序的先后顺序加工。调度的任务是安排 A、B 芯片各 80 片的检测调度排序，约束条件被满足的同时，使检测总耗时最小化。该调度问题需要考虑如下约束：

- (1) 对于一个芯片而言，其检测的每一道工序必须在前一道工序完成后才能开始作业；
- (2) 某一时刻 1 个模块上只能加工 1 个作业；
- (3) 对于一个芯片而言，它的进仓次序决定了它的前 6 个工序（进仓读码、激光超声、过滤、定量、R1R2 加样以及第一次温育）始终要比它前一个进仓的芯片的对应工序的开始时间要晚，且始终要比它后一个进仓的芯片的对应的工序的开始时间要早；
- (4) 各芯片的工序顺序和加工时间已知，不随工序排序的改变而改变。

需要特殊说明的模型构建条件：

- (1) 根据 2.1 节模型假设 4，将前处理台视为 5 个模块构成的集成模块，内部模块串联；

(2) 温育转盘上的 40 个卡槽为独立模块，8 个清洗盒也为独立模块。

基于上述问题的抽象描述，令 (i, j) 表示芯片 i 的第 j 个工序。 S_{ij} 和 T_{ij} 分别表示 (i, j) 的作业开始时刻和作业时间，单位均统一为秒 (s)。 L_{ijm} 表示 (i, j) 是否可以在第 m 个模块上加工：如果 (i, j) 可以在第 m 个模块上加工，则 $L_{ijm} = 1$ ；否则， $L_{ijm} = 0$ ， E_m 为第 m 个模块的完工时间，则可以将转化为如下数学模型（注意多机器选择的作业）：

$$G = \min(\max(E_m), m = 1, 2, \dots, M) \quad (1)$$

$$s.t. \begin{cases} S_{ij} - S_{i(j+1)} + T_{ij} \leq 0; i = 1, 2, \dots, N; j = 1, 2, \dots, I_i - 1 \\ S_{i1} \geq 0; i = 1, 2, \dots, N \\ S_{ij} - S_{pq} + T_{ij} \leq 0; \text{if } L_{ijm} = L_{pqm} = 1 \text{ } i \neq p \\ S_{pq} - S_{ij} + T_{pq} \leq 0; \text{if } L_{ijm} = L_{pqm} = 1 \text{ } i \neq p \end{cases} \quad (2)$$

3.2.3 改进的遗传算法设计

随着遗传算法（genetic algorithm (GA)）在组合优化问题的广泛应用，许多人开始对遗传算法进行深入研究。已有研究表明，遗传算法对求解混合流水车间调度问题具有较好的效果，因此本文呢采用遗传算法来解该问题，遗传算法是计算数学中用于解决最优化的搜索算法，是进化算法的一种。进化算法最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。系统通过模拟生物进化，包括遗传、突变、选择等，来不断地产生新个体，并在算法终止时求得最优个体，即最优解。

遗传算法解决传统的混合流水车间调度问题基本步骤：

- Step 1** 初始化一定数量的种群（染色体编码）；
- Step 2** 计算个体适应度（染色体解码）；
- Step 3** 采用锦标赛法选择染色体并交叉产生新个体；
- Step 4** 个体（染色体）变异；
- Step 5** 达到遗传代数终止算法并从中选取适应度最优的个体作为混合流水车间调度问题的解；

由于问题 2 和 3 与传统的车间调度问题有一些区别，并且为了加快算法的收敛速度提升优化性能，本文在传统的遗传算法流程进行了适当更改：

- Step 1** 采用启发式初始化一定数量的合法种群（染色体编码）；
- Step 2** 计算个体适应度（染色体解码）；
- Step 3** 采用锦标赛法选择染色体并交叉产生通过合法性检验的新个体；
- Step 4** 个体（染色体）变异并进行合法性检验；
- Step 5** 达到遗传代数终止算法并从中选取适应度最优的个体作为混合流水车间调度问题的解。

流程图如下：

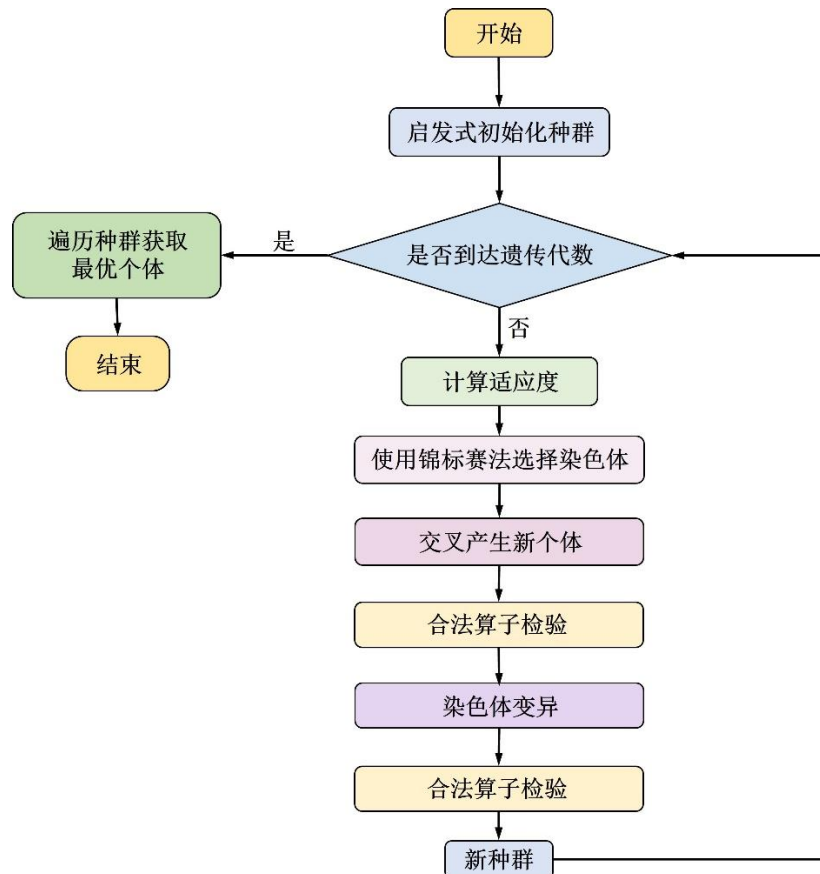


图 1 改进的遗传算法流程图

遗传算法所需参数：

- 1.种群规模：种群中个体的数量，用 `populationNumber` 表示
- 2.染色体长度：个体的染色体的长度，用 `chromosomeSize` 表示
- 3.交叉概率：控制交叉算子的使用频率，用 `crossProbability` 表示，并且值为 0.95
- 4.变异概率：控制变异算子的使用频率，用 `mutationProbability` 表示，并且值为 0.05
- 5.遗传代数：种群的遗传代数，用于控制遗传算法的终止，用 `times` 来表示

遗传算法实现基本步骤及伪代码：

1. 编码及初始化种群

采用工序实数编码来表示染色体，即 M 个模块， N 个芯片，每个工件的工序数为 `process_i`，则染色体长度为 `chromosome=process_1+process_2+...`，对染色体编码如下：

`chromosome=...,w_i,w_j,w_k,...`

其中 w_i 代表第 i 个工件编号，而出现的次数代表该工件的第几道工序。例如 $\{0, 1, 2, 1, 2, 0, 0, 1, 2\}$ ，中 0, 1, 2 表示工件的编号，第几次出现就代表第几道工序。

为了在种群初始化开始是就能够包含一条高质量的染色体，本文采用优先安排总温培时间短的芯片进行仪器的进仓，这启发于温育转盘上的卡槽有限，只有 40 个槽位，而且在本文的假设前提之下，只有不同种类的芯片只有温育时间有区别，因此温育时间的长短是影响总检测耗时的最关键因素。为了保证初期的种群多样性，除了利用启发式生成的高质量染色体外，算法也将随机安排每个芯片的检验顺序，最终将每一次随机生成的染色体个体也加入到种群集合中。

算法伪代码：

表 4 编码及初始化种群算法伪代码

算法伪代码
<pre> for i ← 0 to population_number do chromosome [-1, ..., -1] for j ← 0 to N for k ← 0 to M index = random(chromosome_size) if 第 j 个工件在第 k 台机器上有待加工工序 then chromosome[index] = j 过滤掉 chromosome 的 -1 gene ← Gene 对象 gene.chromosome ← chromosome gene.fitness ← chromosome 的适应度 将 gene 加入到种群集合中 </pre>

2. 解码及计算适应度

将优化目标定义为总加工时间最短，因此适应度定义为最短加工时间的倒数，设 $fitness$ 为对应个体的适应度， $fulfillTime$ 为最短加工时间，因此

$$fitness = \frac{1}{fulfillTime} \quad (3)$$

其中 $fulfillTime$ 的计算方法如下：

首先定义如下变量：

表 5 $fulfillTime$ 计算变量说明

变量	含义
$startTime_{ij}$	第 i 个工件第 j 个工序的最晚开始时间
$endTime_{ij}$	第 i 个工件第 j 个工序的结束时间
$machine_{ij}$	第 i 个工件第 j 个工序所使用的机器编号
$time_{ij}$	第 i 个工件第 j 个工序所用的加工时间
$process_{ij}$	第 i 个工件第 j 台机器上加工的工序编号
$machineWorkTime_i$	第 i 台机器的加工时间
$processIds_i$	第 i 个工件当前的工序编号

然后从左到右遍历个体的染色体序列，其中表示第 i 个工件的编号，则对应的当前工序为，设为 p 。当前工件当前工序所使用的机器编号为，设为 m 。当前工件当前工序对应的加工时间为，设为 t 。则工件的第 p 道工序的最晚开始时间为：

$$startTime_{w_i p} = \begin{cases} machineWorkTime_m & p = 0 \\ \max(endTime_{w_i p-1}, machineWorkTime_m) & p > 0 \end{cases} \quad (4)$$

而第 m 台机器的加工时间为

$$machineWorkTime_m = startTime_{w_i p} + t \quad (5)$$

工件的第 p 道工序的结束时间为

$$endTime_{w_i p} = startTime_{w_i p} + t \quad (6)$$

最后加工完所有工件的最短加工时间 $fulfillTime$ 为

$$fulfillTime = \max_{0 \leq i < m} machineWorkTime_i \quad (7)$$

从而计算出适应度 $fitness$ 。

伪代码如下：

表 6 解码及计算适应度算法伪代码

算法伪代码
<pre> fulfillTime $\leftarrow$ 0 for i $\leftarrow$ 0 to chromosomeSize do wId $\leftarrow$ gene.chromosome[i] pId $\leftarrow$ processIds[wId] ++ mId $\leftarrow$ machine[wId][pId] t $\leftarrow$ time[wId][pId] if pId = 0 then startTime[wId][pId] $\leftarrow$ machineWorkTime[mId] else startTime[wId][pId] $\leftarrow$ max(endTime[wId][pId-1], machineWorktime[mId]) machineWorkTime[mId] $\leftarrow$ startTime[wId][pId] + t endTime[wId][pId] $\leftarrow$ machineWorkTime[mId] fulfillTime $\leftarrow$ max(fulfillTime, machineWorkTime[mId]) fitness $\leftarrow$ 1 / fulfillTime 返回 fitness </pre>

3. 个体选择算子

个体的选择使用锦标赛法，其基本策略为从整个种群中随机抽取 n 个个体让它们竞争，选取其中最优的个体。该算子的选择过程如下：

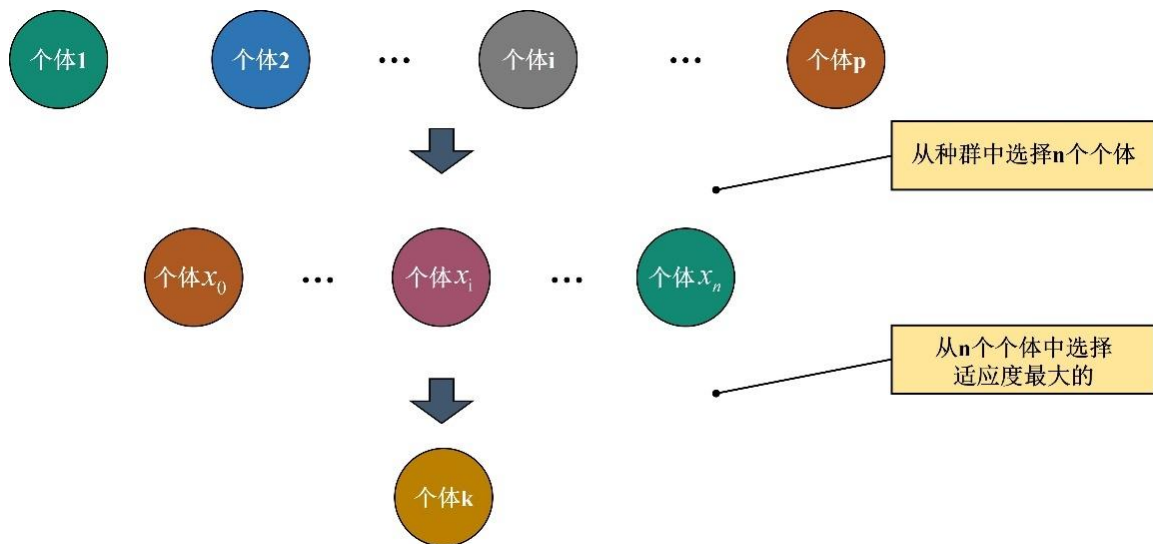


图 2 个体选择算子选择过程

伪代码如下：

表 7 个体选择算子算法伪代码

算法伪代码
indexList $\leftarrow$ 随机生成 n 个索引
bestGene $\leftarrow$ Gene 对象
for i $\leftarrow$ 0 to populationNumber
do if indexList 包含 i
then if bestGene.fitness < geneSet[i].fitness
then bestGene $\leftarrow$ geneSet[i]
返回 bestGene

4. 染色体交叉算子

使用 Order Crossover(OX)交叉算子，该算子的交叉步骤如下：

对于一对染色体 g1, g2，首先随机产生一个起始位置 start 和终止位置 end，并由从 g1 的染色体序列从 start 到 end 的序列中产生一个子代原型。

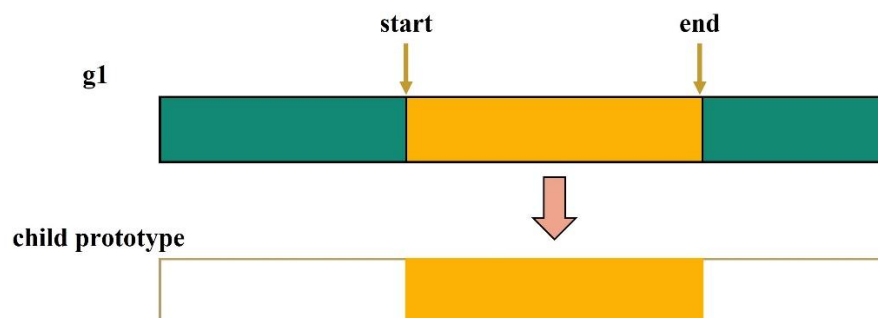


图 3 由 g1 产生一个子代原型

将 g2 中不包含在 child prototype 的其余编码加入到 child prototype 两侧。

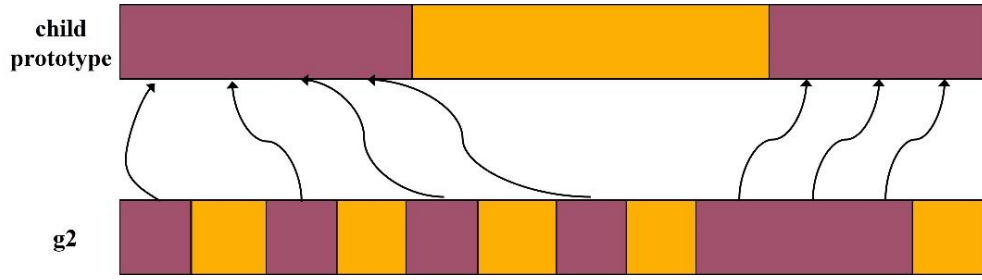


图 4 将 g2 其余编码加到 child prototype 两侧

上述步骤将产生一个 child，交换 g1, g2 即可产生另一个 child。

伪代码如下：

表 8 染色体交叉算子算法伪代码

算法伪代码

```

start ← random (chromosomeSize)
end ← random (chromosomeSize)
childProto ← g1.chromosome[start...end]
temp ← g2.chromosome.length

    for i ← 0 to childProto.length
        do for j ← 0 to temp
            do if childProto[i] = temp[j]
                then 删除 temp 的第 i 项
                break

child ← Gene 对象
child.chromosome ← 将 temp[0...start], childProto, temp[start...]拼接成一个数组
child.fitness ← 计算 child 的适应度
交换 g1, g2 以此产生第二个 child2
返回 child, child2

```

5. 染色体变异算子

变异的作用主要是使算法能跳出局部最优解，因此不同的变异方式对算法能否求得全局最优解有很大的影响。使用位置变异法作为变异算子，即从染色体中随机产生两个位置并交换这两个位置的值。

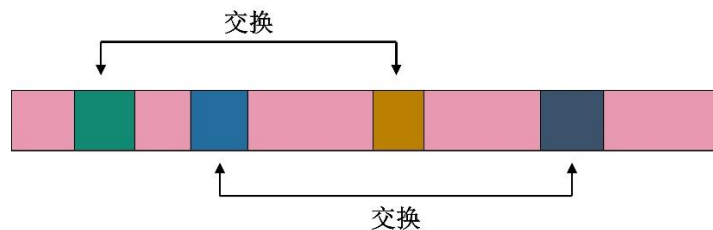


图 5 使用位置变异法作为变异算子

伪代码如下：

表 9 染色体变异算子算法伪代码

算法伪代码
for $i \leftarrow 0$ to n do $p1 \leftarrow \text{random}(\text{chromosomeSize})$ $p2 \leftarrow \text{random}(\text{chromosomeSize})$ 交换 $g.\text{chromosome}[p1]$, $g.\text{chromosome}[p2]$ 的值

6. 染色体检验算子:

因为部分芯片检测工序的安排顺序具有强约束: 从一系列前期处理直到第一次温培开始, 先进仓的芯片的上述工序的开始时间都应该小于之后的芯片的相应工序的开始时间。基于此, 无论是在初始种群生成还是染色体变异或者交叉结束后, 都应该据此检验染色体的合法性: 首先从染色体上计算得到每一个 w_j 的进仓顺序, 然后计算每一个 w_i 的前 6 道工序分别开始的时刻, 接着分别对每一个 w_j 的前 6 道工序按照开始时刻从小到大进行排序, 最后将排序结果与 w_j 的进仓顺序进行比较, 如果相同则认为该染色体是合法的, 可以保留和更新种群, 即该染色体可以生存。

7. 算法整体伪代码如下:

表 10 算法整体伪代码

算法整体伪代码
$\text{geneSet} \leftarrow$ 初始化后的种群集合 for $i \leftarrow 0$ to times do $p <$ 随机生成一个概率 if $p < \text{mutationProbability}$ then $g \leftarrow$ 从 geneSet 随机选取一个个体 对 g 执行变异操作 else $g1, g2 \leftarrow$ 使用锦标赛法从 geneSet 中选取两个个体 $\text{child1}, \text{child2} \leftarrow$ 对 $g1, g2$ 进行交叉, 并产生两个子代 将 $\text{child1}, \text{child2}$ 加入到 geneSet 中 $\text{bestGene} \leftarrow$ Gene 对象 for $i \leftarrow 0$ to geneSet.length do if $\text{bestGene.fitness} < \text{geneSet}[i].\text{fitness}$ $\text{bestGene} \leftarrow \text{geneSet}[i]$ 返回 bestGene

3.2.4 问题 2 求解结果

算法超参数设置如下表所示:

表 11 问题 2 求解时的超参数设置

超参数名称	含义	取值
<i>populationNumber</i>	种群规模	160
<i>chromosomeSize</i>	染色体长度	1600
<i>crossProbability</i>	染色体交叉概率	0.99

超参数名称	含义	取值
<i>mutationProbability</i>	染色体变异概率	0.1

在该参数组合下进行 35 次实验，每次实验的最优结果均为 11059s，约 184min。图 6 展示了全部实验的种群中个体的平均适应度变化情况：

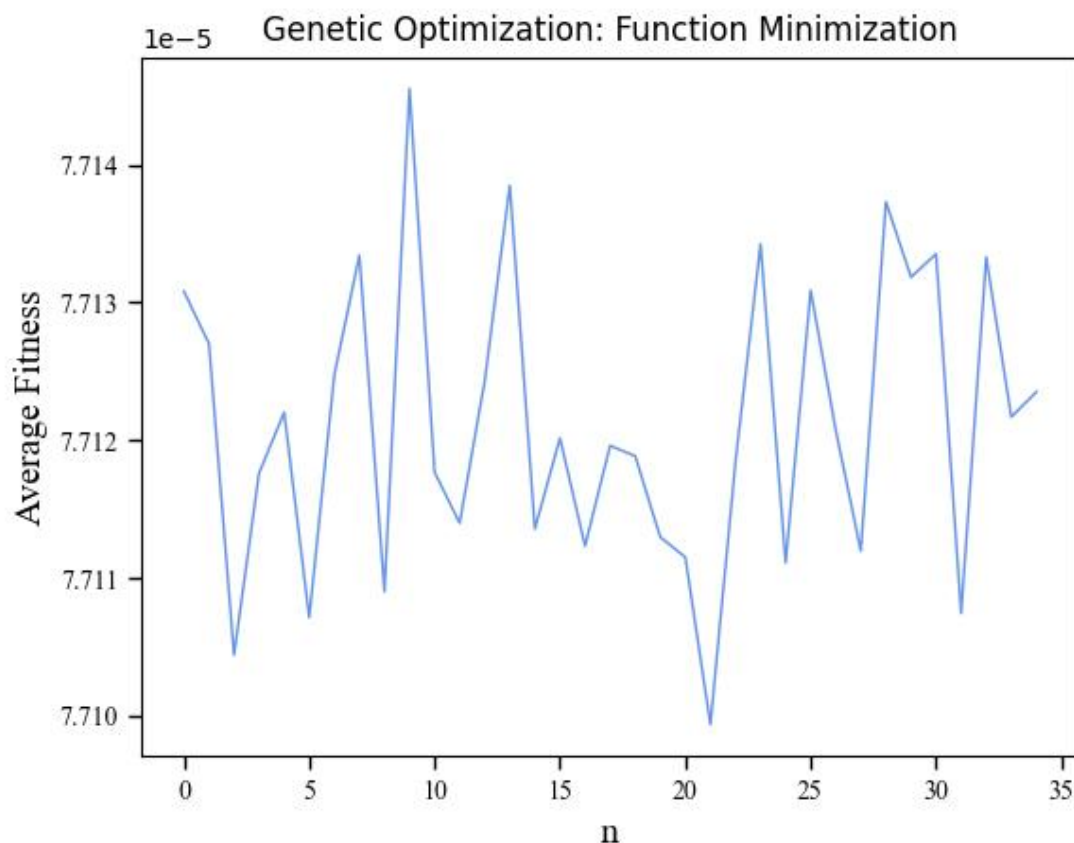


图 6

附录中展示了在本文设计的算法求解的问题 2 的 A、B 型芯片检测的最优调度结果。结果表明，在除温育时间不同，其他工序时间已知且固定不变的情况下，进行芯片调度时，将优先安排芯片 A 进仓，待芯片 A 全部安排完毕时再安排芯片 B 进仓，此时的调度结果达到最优。

3.2.5 问题 3 求解结果

算法参数设置如下表所示：

表 12 问题 3 求解时的超参数设置

超参数名称	含义	取值
<i>populationNumber</i>	种群规模	160
<i>chromosomeSize</i>	染色体长度	1600
<i>crossProbability</i>	染色体交叉概率	0.99
<i>mutationProbability</i>	染色体变异概率	0.1

在该参数组合下进行 35 次实验，每次实验的最优结果均为 10980s,为 183min。图 7 展示了全部实验的种群中个体的平均适应度变化情况。

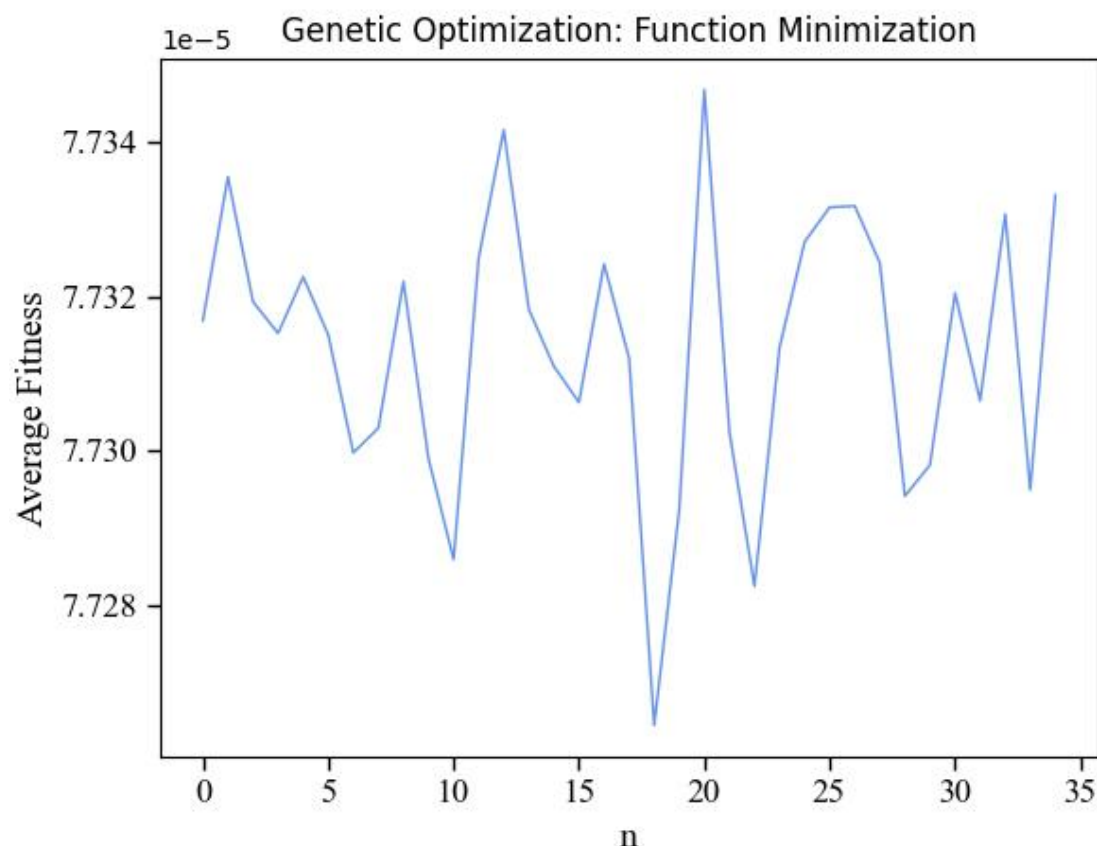


图 7

附录中展示了在本文设计的算法求解的问题 2 的 A、B 和 C 型芯片检测的最优调度结果。结果表明，在除温育时间不同，其他工序时间已知且固定不变的情况下，进行芯片调度时，将优先安排芯片 A 进仓，待芯片 A 全部安排完毕时再安排 C 芯片进仓，最后安排芯片 B 进仓，此时的调度结果达到最优。

3.3 问题 4：分析与模型推广检验

3.3.1 问题 4：模型推广问题

(1) 问题分析

由问题 4 可知，需要将第 2、第 3 个构建的模型进行合理推广，归纳推理出更一般的、对于该类问题的求解步骤以及普适性结论。

(2) 解决思路

根据问题 2 和问题 3 的最优排序结果可以看出，在模型的假设前提之下，芯片的总温育时间是影响仪器运行的调度的关键因素，但是由于该类问题是一个 N-P 难问题，很难求解出最优调度结果的递推公式，因此文本按照这一策略，该问题在此只尝试单一芯片排产时的最优递推。在算法设计方面，只需要按照启发式构造初始解的逻辑，将芯片按照其总温育时间的相对次序进行优良染色体的构造，就能很好的适应和解决多种类芯片不同温育时间不同数量的检测问题。

3.3.2 模型推广检验

单一芯片生产时，其最优递推公式可以分为两种情况讨论：

(3) 第一次温育时间不超过 1685s:

$$E_{M_1} = \sum_{j=1}^{10} S_{1j}$$

$$E_{M_i} = \begin{cases} E_{M_{i-1}} + 36 & \text{if } i \bmod 8 \neq 1 \\ E_{M_{i-1}} + d - 252 & \text{else} \end{cases} \quad (8)$$

其中 E_{M_i} 为第 i 个芯片完成检测的时间， d 为清洗盒清洗时间+检测盘对位时间。

(4) 第一次温育时间超过 1685s:

$$E_{M_1} = \sum_{j=1}^{10} S_{1j}$$

$$E_{M_i} = \begin{cases} E_{M_{i-1}} + 36 & \text{if } i \bmod 8 \neq 1 \\ E_{M_{i-1}} + d - 252 & \text{if } i \bmod 40 \neq 1 \\ E_{M_{i-1}} + a_1 - 1685 + d - 252 & \text{else} \end{cases} \quad (9)$$

其中 E_{M_i} 为第 i 个芯片完成检测的时间， d 为清洗盒清洗时间与检测盘对位时间之和。

多种芯片进行排产时，本文尝试了如下多个算例检验本文提出的求解算法，由于篇幅限制，仅展示其中一种，其最终最优结果为 **14164s**，算例的最终芯片调度排序见附录。此外，由于本文设计的遗传算法染色体的长度与芯片总数相同，因此当进行大规模的芯片检测时，本文的算法效率会大大降低。

表 13

芯片种类	第一次温育时间	第二次温育时间	数量
A	10min	6min	160
B	30min	15min	49
C	20min	10min	83

3.4 问题 5：分析与模型建立

3.4.1 问题 5：两阶段多种芯片组合检测问题

(5) 问题分析

由问题 5 可知，需要先进行 A 和 B 两种芯片的检测，待 30 分钟后，才能进行 C 芯片的检测。该问题与前 3 个问题的区别在于芯片是分批次提供的。

(6) 解决思路

可以将问题 5 转化为带有作业时刻限制的混合流水车间调度问题：即将芯片 C 的进仓时刻限制在 1800s 之后，那么之后的解决沉思录就与问题 2 和 3 的思路相似。

3.4.2 改进的遗传算法调整

就是 4.2.3 节的基础上将限制仪器检验的第一张 C 芯片的进仓时刻，即第一张 C 芯片的进仓时刻需不小于 1800s，并将该约束转化为遗传算法的检验算子。目标函数见式 (9)，而新的约束条件如下：

$$s.t. \begin{cases} S_{ij} - S_{i(j+1)} + T_{ij} \leq 0; i = 1, 2, \dots, N; j = 1, 2, \dots, I_i - 1 \\ S_{i1} \geq 0; i = 1, 2, \dots, N \\ S_{i1} \geq d; i = N_{A+B+1}, \dots, N_{A+B+C} \\ S_{ij} - S_{pq} + T_{ij} \leq 0; \text{when } L_{ijm} = L_{pqm} = 1 \ i \neq p \\ S_{pq} - S_{ij} + T_{pq} \leq 0; \text{when } L_{ijm} = L_{pqm} = 1 \ i \neq p \end{cases} \quad (10)$$

其中，在对芯片进行序号编码时采用“先编号 A 类再编号 B 类最后编号 C 类”的顺序，那么 $i = N_{A+B+1}, \dots, N_{A+B+C}$ 代表的都是 C 类芯片， d 代表 C 类芯片延迟进仓的秒数， $d = 1800$ 。

将算法中的检验算子调整为：在原有检验算子的基础上，从染色体上检索到第一张 C 芯片，并计算获取该片芯片的第一道工序的开始时刻与 1800s 进行比较，如果不小于 1800s 则该染色体合法，代码实现见附录。

3.4.3 问题 5 求解结果

算法参数设置如下表所示：

表 14 问题 5 求解时的超参数设置

超参数名称	含义	取值
<i>populationNumber</i>	种群规模	160
<i>chromosomeSize</i>	染色体长度	1600
<i>crossProbability</i>	染色体交叉概率	0.99
<i>mutationProbability</i>	染色体变异概率	0.1

在该参数组合下进行 35 次实验，每次实验的最优结果均为 11029s，约 184min。图 8 展示了全部实验的种群中个体的平均适应度变化情况。

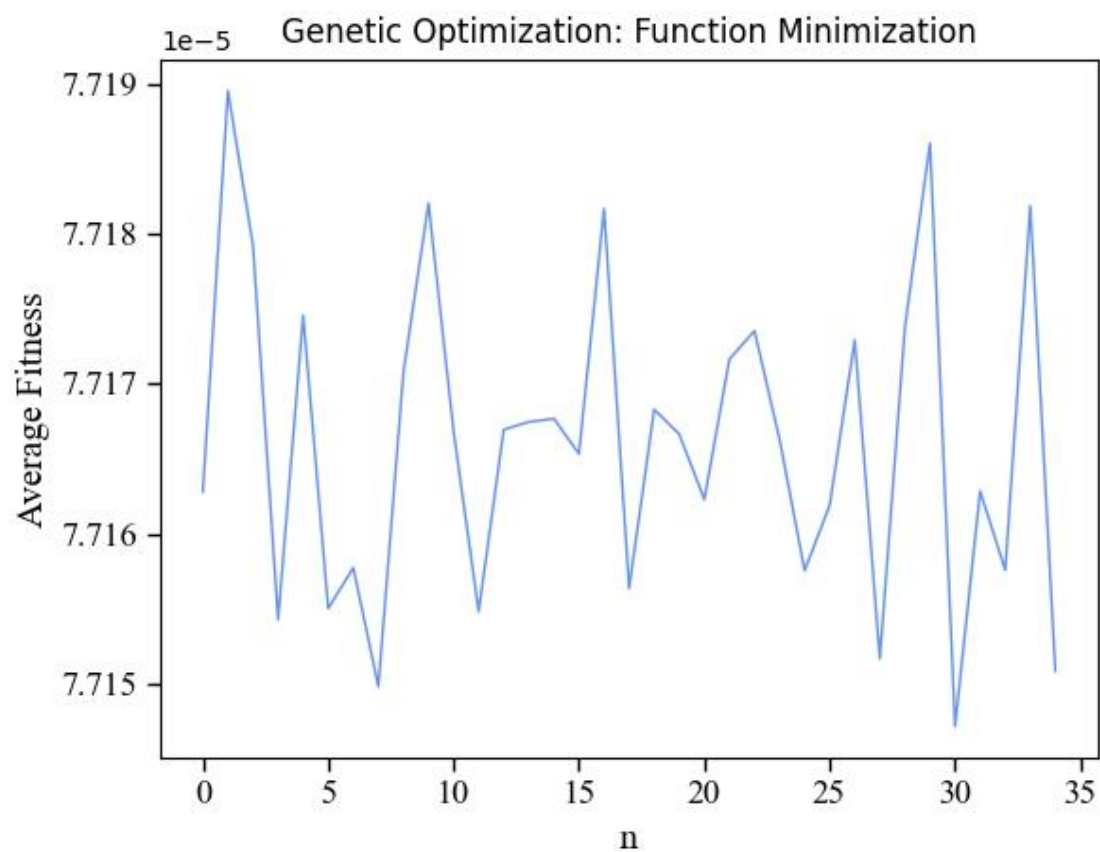


图 8

附录展示了在本文设计的算法求解的问题 2 的 A、B 和 C 型芯片检测的最优调度安排。结果表明，在除温育时间不同，其他工序时间已知且固定不变的情况下，进行芯片调度时，将优先安排芯片 A 进仓，30min 后芯片 A 并未全部安排完毕，待芯片 A 全部安排完毕时再安排 C 芯片进仓，最后安排芯片 B 进仓，此时的调度结果达到最优。

4 模型灵敏度检验

1、问题 1 灵敏度检验

本文模拟了每检验 200 件芯片的处理流程，分别调整模型的加磁时间或者清洗时间，进行灵敏度检验。

清洗时间设定为 325 秒，调整加磁时间为 21、22、23、24、25 秒。

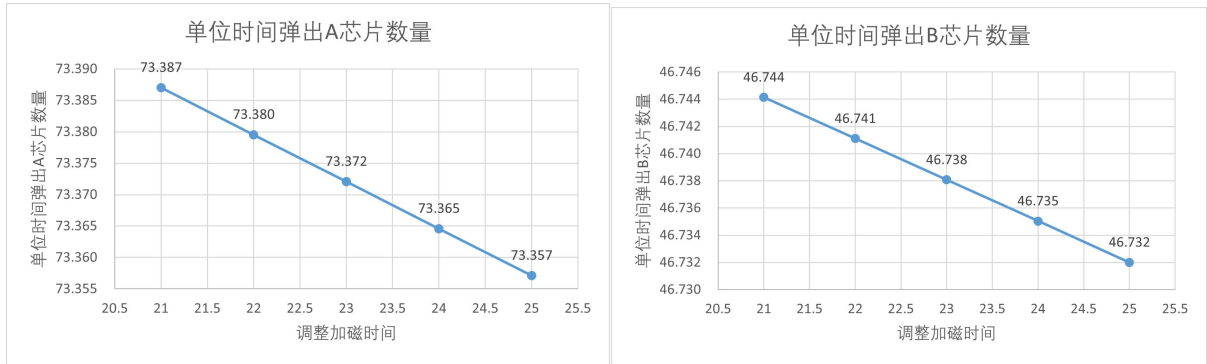


图 9 问题 1 调整加磁时间

加磁时间设定为 21 秒，调整清洗时间增幅分别为 10、20、30、36、72。

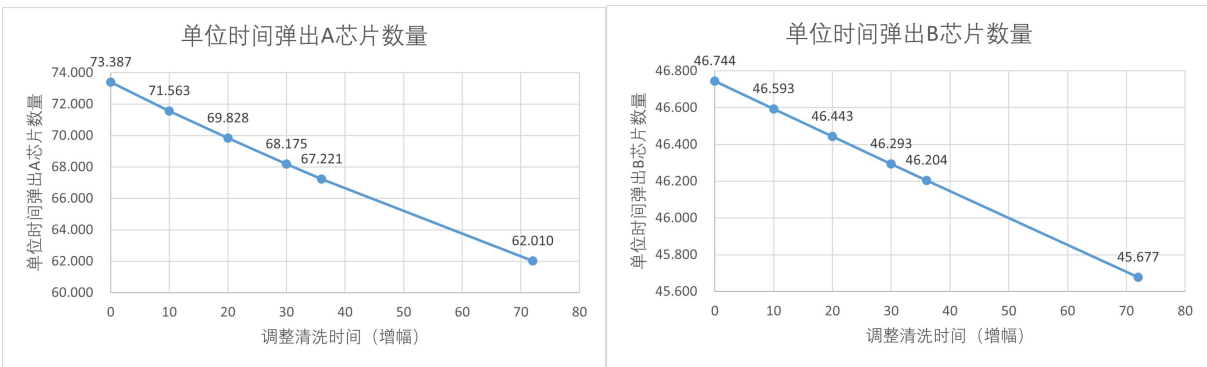


图 10 问题 1 调整清洗时间

2、问题 2 灵敏度检验

优先安排 A 类芯片检测，再安排 B 类芯片检测，计算最优检测时间。如下左图所示，清洗时间设定为 325 秒，调整加磁时间为 21、22、23、24、25 秒。如右图所示，加磁时间设定为 21 秒，调整清洗时间增幅分别为 10、20、30、40、72。

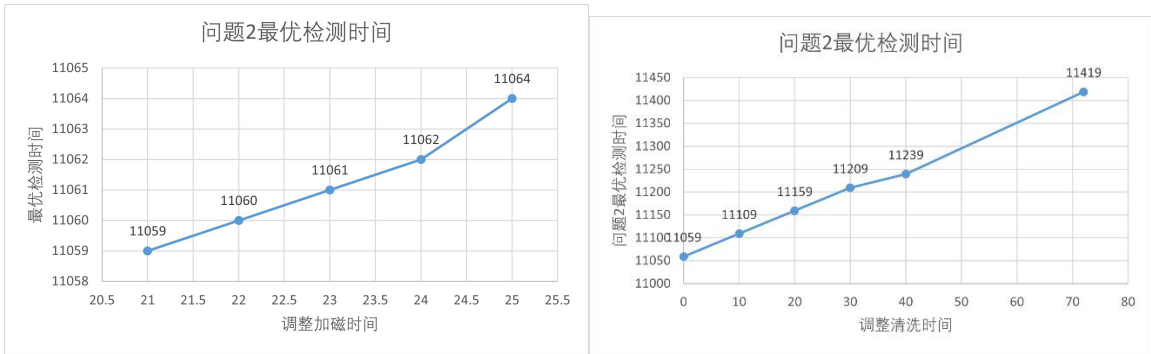


图 11 问题 2 调整加磁时间或调整清洗时间

3、问题 3 灵敏度检验

按照 A、C、B 的顺序安排芯片检测，计算最优检测时间。如下左图所示，清洗时间设定为 325 秒，调整加磁时间为 21、22、23、24、25 秒。如右图所示，加磁时间设定为 21 秒，调整清洗时间增幅分别为 10、20、30、36、40。

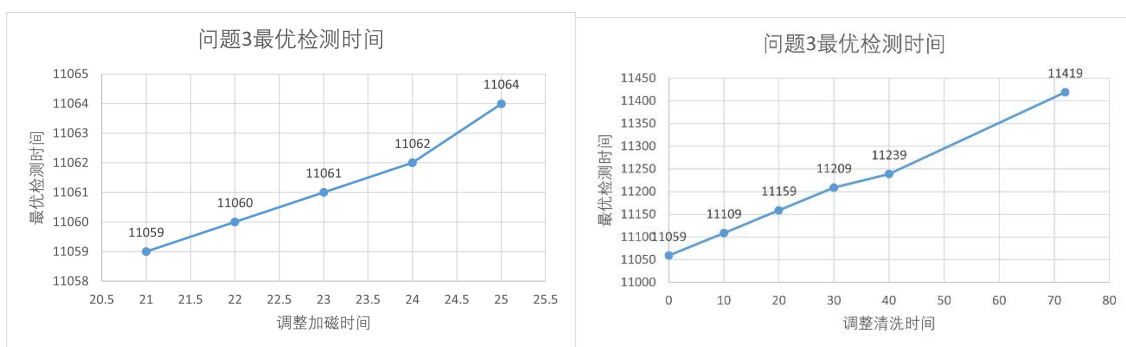


图 12 问题 3 调整加磁时间或调整清洗时间

4、问题 5 灵敏度检验

对分批次提供待检测芯片的调度问题进行分析。如下左图所示，清洗时间设定为 325 秒，调整加磁时间为 21、22、23、24、25 秒。如右图所示，加磁时间设定为 21 秒，调整清洗时间增幅分别为 10、20、30、36、72。

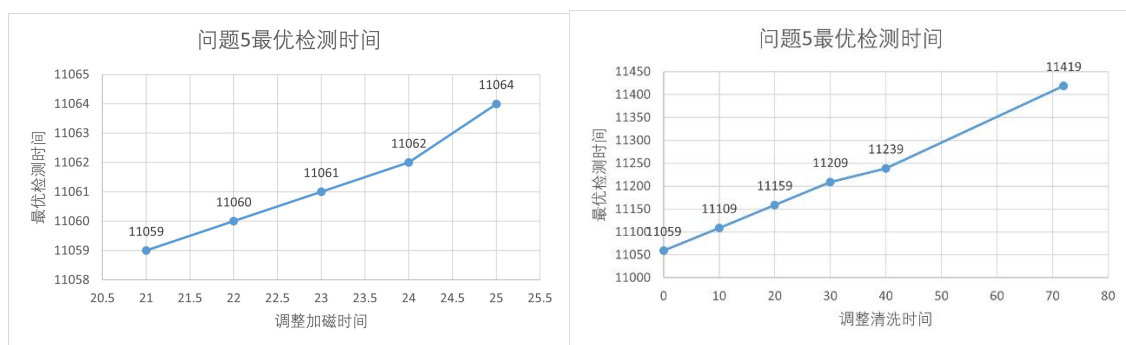


图 13 问题 5 调整加磁时间或调整清洗时间

通过观察模型对于两个输入变量的敏感程度可以看出，无论是调整模型加磁时间，还是调整清洗时间，最优检测时间均平稳变化，因此模型较为稳定。

5 模型评价与推广

5.1 模型的优点

(1) 模型充分结合实际, 简化分析盘的设备 and 并行运行条件, 考虑了诸多重要因素得到合理的模型, 如: 假设转动相同距离所需相同时长, 在计算最多清洗芯片数量时假定磁珠加珠时间和清洗时间都是最短, 假设前处理台有 5 各芯片槽分别对应不同的前处理台步骤. 这样得到的模型贴合实际, 具有较高的应用价值, 可以推广到流水线调度模型;

(2) 模型运用遗传学思想, 抓住影响分析盘调度处理最短、处理芯片数量最多问题的重要因素, 将复杂的分析盘芯片最优化调度问题转化为简单的遗传学最优解问题, 合理设置参数, 模型的输出结果符合题目要求, 能解决实际问题;

(3) 本文使用的遗传算法具有高效、并行、全局搜索等优点, 对于求解最优调度模型非常适用;

(4) 本文得到的改进遗传算法具有效率高、多方向全局搜索、快速找到最优解等特点, 在现有条件下能有效提高生产效率.

5.2 模型的不足

(5) 实际应用中, 第一次和第二次温育共用卡槽盘和前处理阶段与转盘并行可能也是重要的因素, 但本文未能考虑到这些因素的影响, 一定程度上影响了模型的准确性;

(6) 本文提出的模型对于现有条件使用效果较好, 由于时间问题没有对其他情况进行检验. 对于局部最优解的问题, 可能无法达到较好的效果;

(7) 本文提出的模型完备性和鲁棒性有待进一步验证.

5.3 模型的推广

本文提出的模型不仅适用于转盘调度问题, 还可以进一步推广到流水线调度问题中, 可以将各个工序处理参数替换成新问题流程处理参数, 将转盘转动过程替换为新问题的预处理流程, 从而解决更广泛的流水线调度问题; 此外, 该模型在自组织、自学习、群体自适应等问题上, 都能够从多方向搜索, 从点到面发现全局最优解。

参考文献

- [1] 张晓亮. 小型全自动化学发光免疫分析仪关键技术研究[D]. 中国科学技术大学, 2019.
- [2] 胡善德. 全自动生化免疫分析仪多任务优化调度研究及其软件系统实现[D]. 华南理工大学, 2014.

附 录

附录 A: 问题 1 完整结果

表 15 问题 1.A 型芯片的调度结果

芯片 序号	进仓 时间	第一次 育温起 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	30	186	794	823	1139	1476	1501
3	60	216	824	853	1169	1506	1531
4	90	246	854	883	1199	1536	1561
5	120	276	884	913	1229	1566	1591
6	150	306	914	943	1259	1596	1621
7	180	336	944	973	1289	1626	1651
8	210	366	974	1003	1319	1656	1681
9	240	396	1004	1033	1434	1771	1796
10	270	426	1034	1063	1464	1801	1826
11	300	456	1064	1093	1494	1831	1856
12	330	486	1094	1123	1524	1861	1886
13	360	516	1124	1153	1554	1891	1916
14	390	546	1154	1183	1584	1921	1946
15	420	576	1184	1213	1614	1951	1976
16	450	606	1214	1243	1644	1981	2006
17	480	636	1244	1273	1759	2096	2121
18	510	666	1274	1303	1789	2126	2151
19	540	696	1304	1333	1819	2156	2181
20	570	726	1334	1363	1849	2186	2211
21	600	756	1364	1393	1879	2216	2241
22	630	786	1394	1423	1909	2246	2271
23	660	816	1424	1453	1939	2276	2301
24	690	846	1454	1483	1969	2306	2331
25	720	876	1484	1513	2084	2421	2446
26	750	906	1514	1543	2114	2451	2476
27	780	936	1544	1573	2144	2481	2506
28	810	966	1574	1603	2174	2511	2536
29	840	996	1604	1633	2204	2541	2566
30	870	1026	1634	1663	2234	2571	2596
31	900	1056	1664	1693	2264	2601	2626

32	930	1086	1694	1723	2294	2631	2656
33	960	1116	1724	1753	2409	2746	2771
34	990	1146	1754	1783	2439	2776	2801
35	1020	1176	1784	1813	2469	2806	2831
36	1050	1206	1814	1843	2499	2836	2861
37	1080	1236	1844	1873	2529	2866	2891
38	1110	1266	1874	1903	2559	2896	2921
39	1140	1296	1904	1933	2589	2926	2951
40	1170	1326	1934	1963	2619	2956	2981
41	1200	1356	1964	1993	2734	3071	3096
42	1230	1386	1994	2023	2764	3101	3126
43	1260	1416	2024	2053	2794	3131	3156
44	1290	1446	2054	2083	2824	3161	3186
45	1320	1476	2084	2113	2854	3191	3216
46	1350	1506	2114	2143	2884	3221	3246
47	1380	1536	2144	2173	2914	3251	3276
48	1420	1576	2184	2213	2944	3281	3306
49	1450	1606	2214	2243	3059	3396	3421
50	1480	1636	2244	2273	3089	3426	3451
51	1510	1666	2274	2303	3119	3456	3481
52	1540	1696	2304	2333	3149	3486	3511
53	1570	1726	2334	2363	3179	3516	3541
54	1600	1756	2364	2393	3209	3546	3571
55	1630	1786	2394	2423	3239	3576	3601
56	1660	1816	2424	2453	3269	3606	3631
57	1690	1846	2454	2483	3384	3721	3746
58	1720	1876	2484	2513	3414	3751	3776
59	1750	1906	2514	2543	3444	3781	3806
60	1780	1936	2544	2573	3474	3811	3836
61	1810	1966	2574	2603	3504	3841	3866
62	1840	1996	2604	2633	3534	3871	3896
63	1870	2026	2634	2663	3564	3901	3926
64	1900	2056	2664	2693	3594	3931	3956
65	1930	2086	2694	2723	3709	4046	4071
66	1960	2116	2724	2753	3739	4076	4101
67	1990	2146	2754	2783	3769	4106	4131
68	2020	2176	2784	2813	3799	4136	4161

69	2050	2206	2814	2843	3829	4166	4191
70	2080	2236	2844	2873	3859	4196	4221
71	2110	2266	2874	2903	3889	4226	4251
72	2140	2296	2904	2933	3919	4256	4281
73	2170	2326	2934	2963	4034	4371	4396
74	2200	2356	2964	2993	4064	4401	4426
75	2230	2386	2994	3023	4094	4431	4456
76	2260	2416	3024	3053	4124	4461	4486
77	2290	2446	3054	3083	4154	4491	4516
78	2320	2476	3084	3113	4184	4521	4546
79	2350	2506	3114	3143	4214	4551	4576
80	2380	2536	3144	3173	4244	4581	4606
81	2410	2566	3174	3203	4359	4696	4721
82	2440	2596	3204	3233	4389	4726	4751
83	2470	2626	3234	3263	4419	4756	4781
84	2500	2656	3264	3293	4449	4786	4811
85	2530	2686	3294	3323	4479	4816	4841
86	2560	2716	3324	3353	4509	4846	4871
87	2590	2746	3354	3383	4539	4876	4901
88	2620	2776	3384	3413	4569	4906	4931
89	2650	2806	3414	3443	4684	5021	5046
90	2680	2836	3444	3473	4714	5051	5076
91	2710	2866	3474	3503	4744	5081	5106
92	2740	2896	3504	3533	4774	5111	5136
93	2770	2926	3534	3563	4804	5141	5166
94	2800	2956	3564	3593	4834	5171	5196
95	2830	2986	3594	3623	4864	5201	5226
96	2860	3016	3624	3653	4894	5231	5256

表 16 问题 1 B 芯片的调度结果

芯片序号	进仓时间	第一次育温起始时间	磁珠加样起始时间	第二次育温起始时间	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	30	186	2594	2623	3839	4176	4201
3	60	216	2624	2653	3869	4206	4231
4	90	246	2654	2683	3899	4236	4261
5	120	276	2684	2713	3929	4266	4291
6	150	306	2714	2743	3959	4296	4321
7	180	336	2744	2773	3989	4326	4351
8	210	366	2774	2803	4019	4356	4381
9	240	396	2804	2833	4134	4471	4496
10	270	426	2834	2863	4164	4501	4526
11	300	456	2864	2893	4194	4531	4556
12	330	486	2894	2923	4224	4561	4586
13	360	516	2924	2953	4254	4591	4616
14	390	546	2954	2983	4284	4621	4646
15	420	576	2984	3013	4314	4651	4676
16	450	606	3014	3043	4344	4681	4706
17	480	636	3044	3073	4459	4796	4821
18	510	666	3074	3103	4489	4826	4851
19	540	696	3104	3133	4519	4856	4881
20	570	726	3134	3163	4549	4886	4911
21	600	756	3164	3193	4579	4916	4941
22	630	786	3194	3223	4609	4946	4971
23	660	816	3224	3253	4639	4976	5001
24	690	846	3254	3283	4669	5006	5031
25	720	876	3284	3313	4784	5121	5146
26	750	906	3314	3343	4814	5151	5176
27	780	936	3344	3373	4844	5181	5206
28	810	966	3374	3403	4874	5211	5236
29	840	996	3404	3433	4904	5241	5266
30	870	1026	3434	3463	4934	5271	5296
31	900	1056	3464	3493	4964	5301	5326

32	930	1086	3494	3523	4994	5331	5356
33	960	1116	3524	3553	5109	5446	5471
34	990	1146	3554	3583	5139	5476	5501
35	1020	1176	3584	3613	5169	5506	5531
36	1050	1206	3614	3643	5199	5536	5561
37	1080	1236	3644	3673	5229	5566	5591
38	1110	1266	3674	3703	5259	5596	5621
39	1140	1296	3704	3733	5289	5626	5651
40	1170	1326	3734	3763	5319	5656	5681
41	1200	2556	4964	4993	6209	6546	6571
42	1230	2586	4994	5023	6239	6576	6601
43	1260	2616	5024	5053	6269	6606	6631
44	1290	2646	5054	5083	6299	6636	6661
45	1320	2676	5084	5113	6329	6666	6691
46	1350	2706	5114	5143	6359	6696	6721
47	1380	2736	5144	5173	6389	6726	6751
48	1410	2766	5174	5203	6419	6756	6781
49	1440	2796	5204	5233	6534	6871	6896
50	1470	2826	5234	5263	6564	6901	6926
51	1500	2856	5264	5293	6594	6931	6956
52	1530	2886	5294	5323	6624	6961	6986
53	1560	2916	5324	5353	6654	6991	7016
54	1590	2946	5354	5383	6684	7021	7046
55	1620	2976	5384	5413	6714	7051	7076
56	1650	3006	5414	5443	6744	7081	7106
57	1680	3036	5444	5473	6859	7196	7221
58	1710	3066	5474	5503	6889	7226	7251
59	1740	3096	5504	5533	6919	7256	7281
60	1770	3126	5534	5563	6949	7286	7311
61	1800	3156	5564	5593	6979	7316	7341
62	1830	3186	5594	5623	7009	7346	7371
63	1860	3216	5624	5653	7039	7376	7401
64	1890	3246	5654	5683	7069	7406	7431
65	1920	3276	5684	5713	7184	7521	7546
66	1950	3306	5714	5743	7214	7551	7576
67	1980	3336	5744	5773	7244	7581	7606
68	2010	3366	5774	5803	7274	7611	7636

69	2040	3396	5804	5833	7304	7641	7666
70	2070	3426	5834	5863	7334	7671	7696
71	2100	3456	5864	5893	7364	7701	7726
72	2130	3486	5894	5923	7394	7731	7756
73	2160	3516	5924	5953	7509	7846	7871
74	2190	3546	5954	5983	7539	7876	7901
75	2220	3576	5984	6013	7569	7906	7931
76	2250	3606	6014	6043	7599	7936	7961
77	2280	3636	6044	6073	7629	7966	7991
78	2310	3666	6074	6103	7659	7996	8021
79	2340	3696	6104	6133	7689	8026	8051
80	2370	3726	6134	6163	7719	8056	8081
81	2400	4956	7364	7393	8609	8946	8971
82	2430	4986	7394	7423	8639	8976	9001
83	2460	5016	7424	7453	8669	9006	9031
84	2490	5046	7454	7483	8699	9036	9061
85	2520	5076	7484	7513	8729	9066	9091
86	2550	5106	7514	7543	8759	9096	9121
87	2580	5136	7544	7573	8789	9126	9151
88	2610	5166	7574	7603	8819	9156	9181
89	2640	5196	7604	7633	8934	9271	9296
90	2670	5226	7634	7663	8964	9301	9326
91	2700	5256	7664	7693	8994	9331	9356
92	2730	5286	7694	7723	9024	9361	9386
93	2760	5316	7724	7753	9054	9391	9416
94	2790	5346	7754	7783	9084	9421	9446
95	2820	5376	7784	7813	9114	9451	9476
96	2850	5406	7814	7843	9144	9481	9506
97	2880	5436	7844	7873	9259	9596	9621
98	2910	5466	7874	7903	9289	9626	9651
99	2940	5496	7904	7933	9319	9656	9681
100	2970	5526	7934	7963	9349	9686	9711
101	3000	5556	7964	7993	9379	9716	9741
102	3030	5586	7994	8023	9409	9746	9771
103	3060	5616	8024	8053	9439	9776	9801
104	3090	5646	8054	8083	9469	9806	9831
105	3120	5676	8084	8113	9584	9921	9946

106	3150	5706	8114	8143	9614	9951	9976
107	3180	5736	8144	8173	9644	9981	10006
108	3210	5766	8174	8203	9674	10011	10036
109	3240	5796	8204	8233	9704	10041	10066
110	3270	5826	8234	8263	9734	10071	10096
111	3300	5856	8264	8293	9764	10101	10126
112	3330	5886	8294	8323	9794	10131	10156
113	3360	5916	8324	8353	9909	10246	10271
114	3390	5946	8354	8383	9939	10276	10301
115	3420	5976	8384	8413	9969	10306	10331
116	3450	6006	8414	8443	9999	10336	10361
117	3480	6036	8444	8473	10029	10366	10391
118	3510	6066	8474	8503	10059	10396	10421
119	3540	6096	8504	8533	10089	10426	10451
120	3570	6126	8534	8563	10119	10456	10481
121	3600	7356	9764	9793	11009	11346	11371

附录 B: 问题 2 完整结果

表 17 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	第一次育温起始时间	磁珠加样起始时间	第二次育温起始时间	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	A	30	192	800	829	1145	1482	1507
3	A	60	228	836	865	1181	1518	1543
4	A	90	264	872	901	1217	1554	1579
5	A	120	300	908	937	1253	1590	1615
6	A	156	336	944	973	1289	1626	1651
7	A	192	372	980	1009	1325	1662	1687
8	A	228	408	1016	1045	1361	1698	1723
9	A	264	444	1052	1081	1446	1783	1808
10	A	300	480	1088	1117	1482	1819	1844
11	A	336	516	1124	1153	1518	1855	1880
12	A	372	552	1160	1189	1554	1891	1916
13	A	408	588	1196	1225	1590	1927	1952
14	A	444	624	1232	1261	1626	1963	1988
15	A	480	660	1268	1297	1662	1999	2024
16	A	516	696	1304	1333	1698	2035	2060

17	A	552	732	1340	1369	1783	2120	2145
18	A	588	768	1376	1405	1819	2156	2181
19	A	624	804	1412	1441	1855	2192	2217
20	A	660	840	1448	1477	1891	2228	2253
21	A	696	876	1484	1513	1927	2264	2289
22	A	732	912	1520	1549	1963	2300	2325
23	A	768	948	1556	1585	1999	2336	2361
24	A	804	984	1592	1621	2035	2372	2397
25	A	840	1020	1628	1657	2120	2457	2482
26	A	876	1056	1664	1693	2156	2493	2518
27	A	912	1092	1700	1729	2192	2529	2554
28	A	948	1128	1736	1765	2228	2565	2590
29	A	984	1164	1772	1801	2264	2601	2626
30	A	1020	1200	1808	1837	2300	2637	2662
31	A	1056	1236	1844	1873	2336	2673	2698
32	A	1092	1272	1880	1909	2372	2709	2734
33	A	1128	1308	1916	1945	2457	2794	2819
34	A	1164	1344	1952	1981	2493	2830	2855
35	A	1200	1380	1988	2017	2529	2866	2891
36	A	1236	1416	2024	2053	2565	2902	2927
37	A	1272	1452	2060	2089	2601	2938	2963
38	A	1308	1488	2096	2125	2637	2974	2999
39	A	1344	1524	2132	2161	2673	3010	3035
40	A	1380	1560	2168	2197	2709	3046	3071
41	A	1416	1596	2204	2233	2794	3131	3156
42	A	1452	1632	2240	2269	2830	3167	3192
43	A	1488	1668	2276	2305	2866	3203	3228
44	A	1524	1704	2312	2341	2902	3239	3264
45	A	1560	1740	2348	2377	2938	3275	3300
46	A	1596	1776	2384	2413	2974	3311	3336
47	A	1632	1812	2420	2449	3010	3347	3372
48	A	1668	1848	2456	2485	3046	3383	3408
49	A	1704	1884	2492	2521	3131	3468	3493
50	A	1740	1920	2528	2557	3167	3504	3529
51	A	1776	1956	2564	2593	3203	3540	3565
52	A	1812	1992	2600	2629	3239	3576	3601
53	A	1848	2028	2636	2665	3275	3612	3637

54	A	1884	2064	2672	2701	3311	3648	3673
55	A	1920	2100	2708	2737	3347	3684	3709
56	A	1956	2136	2744	2773	3383	3720	3745
57	A	1992	2172	2780	2809	3468	3805	3830
58	A	2028	2208	2816	2845	3504	3841	3866
59	A	2064	2244	2852	2881	3540	3877	3902
60	A	2100	2280	2888	2917	3576	3913	3938
61	A	2136	2316	2924	2953	3612	3949	3974
62	A	2172	2352	2960	2989	3648	3985	4010
63	A	2208	2388	2996	3025	3684	4021	4046
64	A	2244	2424	3032	3061	3720	4057	4082
65	A	2280	2460	3068	3097	3805	4142	4167
66	A	2316	2496	3104	3133	3841	4178	4203
67	A	2352	2532	3140	3169	3877	4214	4239
68	A	2388	2568	3176	3205	3913	4250	4275
69	A	2424	2604	3212	3241	3949	4286	4311
70	A	2460	2640	3248	3277	3985	4322	4347
71	A	2496	2676	3284	3313	4021	4358	4383
72	A	2532	2712	3320	3349	4057	4394	4419
73	A	2568	2748	3356	3385	4142	4479	4504
74	A	2604	2784	3392	3421	4178	4515	4540
75	A	2640	2820	3428	3457	4214	4551	4576
76	A	2676	2856	3464	3493	4250	4587	4612
77	A	2712	2892	3500	3529	4286	4623	4648
78	A	2748	2928	3536	3565	4322	4659	4684
79	A	2784	2964	3572	3601	4358	4695	4720
80	A	2820	3000	3608	3637	4394	4731	4756
81	B	2856	3036	5444	5473	6689	7026	7051
82	B	2892	3072	5480	5509	6725	7062	7087
83	B	2928	3108	5516	5545	6761	7098	7123
84	B	2964	3144	5552	5581	6797	7134	7159
85	B	3000	3180	5588	5617	6833	7170	7195
86	B	3036	3216	5624	5653	6869	7206	7231
87	B	3072	3252	5660	5689	6905	7242	7267
88	B	3108	3288	5696	5725	6941	7278	7303
89	B	3144	3324	5732	5761	7026	7363	7388
90	B	3180	3360	5768	5797	7062	7399	7424

91	B	3216	3396	5804	5833	7098	7435	7460
92	B	3252	3432	5840	5869	7134	7471	7496
93	B	3288	3468	5876	5905	7170	7507	7532
94	B	3324	3504	5912	5941	7206	7543	7568
95	B	3360	3540	5948	5977	7242	7579	7604
96	B	3396	3576	5984	6013	7278	7615	7640
97	B	3432	3612	6020	6049	7363	7700	7725
98	B	3468	3648	6056	6085	7399	7736	7761
99	B	3504	3684	6092	6121	7435	7772	7797
100	B	3540	3720	6128	6157	7471	7808	7833
101	B	3576	3756	6164	6193	7507	7844	7869
102	B	3612	3792	6200	6229	7543	7880	7905
103	B	3648	3828	6236	6265	7579	7916	7941
104	B	3684	3864	6272	6301	7615	7952	7977
105	B	3720	3900	6308	6337	7700	8037	8062
106	B	3756	3936	6344	6373	7736	8073	8098
107	B	3792	3972	6380	6409	7772	8109	8134
108	B	3828	4008	6416	6445	7808	8145	8170
109	B	3864	4044	6452	6481	7844	8181	8206
110	B	3900	4080	6488	6517	7880	8217	8242
111	B	3936	4116	6524	6553	7916	8253	8278
112	B	3972	4152	6560	6589	7952	8289	8314
113	B	4008	4188	6596	6625	8037	8374	8399
114	B	4044	4224	6632	6661	8073	8410	8435
115	B	4080	4260	6668	6697	8109	8446	8471
116	B	4116	4296	6704	6733	8145	8482	8507
117	B	4152	4332	6740	6769	8181	8518	8543
118	B	4188	4368	6776	6805	8217	8554	8579
119	B	4224	4404	6812	6841	8253	8590	8615
120	B	4260	4440	6848	6877	8289	8626	8651
121	B	4296	5444	7852	7881	9097	9434	9459
122	B	4332	5480	7888	7917	9133	9470	9495
123	B	4368	5516	7924	7953	9169	9506	9531
124	B	4404	5552	7960	7989	9205	9542	9567
125	B	4440	5588	7996	8025	9241	9578	9603
126	B	5444	5624	8032	8061	9277	9614	9639
127	B	5480	5660	8068	8097	9313	9650	9675

128	B	5516	5696	8104	8133	9349	9686	9711
129	B	5552	5732	8140	8169	9434	9771	9796
130	B	5588	5768	8176	8205	9470	9807	9832
131	B	5624	5804	8212	8241	9506	9843	9868
132	B	5660	5840	8248	8277	9542	9879	9904
133	B	5696	5876	8284	8313	9578	9915	9940
134	B	5732	5912	8320	8349	9614	9951	9976
135	B	5768	5948	8356	8385	9650	9987	10012
136	B	5804	5984	8392	8421	9686	10023	10048
137	B	5840	6020	8428	8457	9771	10108	10133
138	B	5876	6056	8464	8493	9807	10144	10169
139	B	5912	6092	8500	8529	9843	10180	10205
140	B	5948	6128	8536	8565	9879	10216	10241
141	B	5984	6164	8572	8601	9915	10252	10277
142	B	6020	6200	8608	8637	9951	10288	10313
143	B	6056	6236	8644	8673	9987	10324	10349
144	B	6092	6272	8680	8709	10023	10360	10385
145	B	6128	6308	8716	8745	10108	10445	10470
146	B	6164	6344	8752	8781	10144	10481	10506
147	B	6200	6380	8788	8817	10180	10517	10542
148	B	6236	6416	8824	8853	10216	10553	10578
149	B	6272	6452	8860	8889	10252	10589	10614
150	B	6308	6488	8896	8925	10288	10625	10650
151	B	6344	6524	8932	8961	10324	10661	10686
152	B	6380	6560	8968	8997	10360	10697	10722
153	B	6416	6596	9004	9033	10445	10782	10807
154	B	6452	6632	9040	9069	10481	10818	10843
155	B	6488	6668	9076	9105	10517	10854	10879
156	B	6524	6704	9112	9141	10553	10890	10915
157	B	6560	6740	9148	9177	10589	10926	10951
158	B	6596	6776	9184	9213	10625	10962	10987
159	B	6632	6812	9220	9249	10661	10998	11023
160	B	6668	6848	9256	9285	10697	11034	11059

附录 C: 问题 3 完整结果

表 18 问题 2.A、B 型芯片的调度结果（单位：s）

芯片	芯片	进仓	第一次	磁珠加	第二次	清洗起	检测起	检测结
----	----	----	-----	-----	-----	-----	-----	-----

序号	类型	时间	温育起始时间	样起始时间	温育起始时间	始时间	始时间	束时间
1	A	0	156	764	793	1109	1446	1471
2	A	30	192	800	829	1145	1482	1507
3	A	60	228	836	865	1181	1518	1543
4	A	90	264	872	901	1217	1554	1579
5	A	120	300	908	937	1253	1590	1615
6	A	156	336	944	973	1289	1626	1651
7	A	192	372	980	1009	1325	1662	1687
8	A	228	408	1016	1045	1361	1698	1723
9	A	264	444	1052	1081	1446	1783	1808
10	A	300	480	1088	1117	1482	1819	1844
11	A	336	516	1124	1153	1518	1855	1880
12	A	372	552	1160	1189	1554	1891	1916
13	A	408	588	1196	1225	1590	1927	1952
14	A	444	624	1232	1261	1626	1963	1988
15	A	480	660	1268	1297	1662	1999	2024
16	A	516	696	1304	1333	1698	2035	2060
17	A	552	732	1340	1369	1783	2120	2145
18	A	588	768	1376	1405	1819	2156	2181
19	A	624	804	1412	1441	1855	2192	2217
20	A	660	840	1448	1477	1891	2228	2253
21	A	696	876	1484	1513	1927	2264	2289
22	A	732	912	1520	1549	1963	2300	2325
23	A	768	948	1556	1585	1999	2336	2361
24	A	804	984	1592	1621	2035	2372	2397
25	A	840	1020	1628	1657	2120	2457	2482
26	A	876	1056	1664	1693	2156	2493	2518
27	A	912	1092	1700	1729	2192	2529	2554
28	A	948	1128	1736	1765	2228	2565	2590
29	A	984	1164	1772	1801	2264	2601	2626
30	A	1020	1200	1808	1837	2300	2637	2662
31	A	1056	1236	1844	1873	2336	2673	2698
32	A	1092	1272	1880	1909	2372	2709	2734
33	A	1128	1308	1916	1945	2457	2794	2819
34	A	1164	1344	1952	1981	2493	2830	2855
35	A	1200	1380	1988	2017	2529	2866	2891
36	A	1236	1416	2024	2053	2565	2902	2927

37	A	1272	1452	2060	2089	2601	2938	2963
38	A	1308	1488	2096	2125	2637	2974	2999
39	A	1344	1524	2132	2161	2673	3010	3035
40	A	1380	1560	2168	2197	2709	3046	3071
41	A	1416	1596	2204	2233	2794	3131	3156
42	A	1452	1632	2240	2269	2830	3167	3192
43	A	1488	1668	2276	2305	2866	3203	3228
44	A	1524	1704	2312	2341	2902	3239	3264
45	A	1560	1740	2348	2377	2938	3275	3300
46	A	1596	1776	2384	2413	2974	3311	3336
47	A	1632	1812	2420	2449	3010	3347	3372
48	A	1668	1848	2456	2485	3046	3383	3408
49	A	1704	1884	2492	2521	3131	3468	3493
50	A	1740	1920	2528	2557	3167	3504	3529
51	A	1776	1956	2564	2593	3203	3540	3565
52	A	1812	1992	2600	2629	3239	3576	3601
53	A	1848	2028	2636	2665	3275	3612	3637
54	A	1884	2064	2672	2701	3311	3648	3673
55	A	1920	2100	2708	2737	3347	3684	3709
56	A	1956	2136	2744	2773	3383	3720	3745
57	A	1992	2172	2780	2809	3468	3805	3830
58	A	2028	2208	2816	2845	3504	3841	3866
59	A	2064	2244	2852	2881	3540	3877	3902
60	A	2100	2280	2888	2917	3576	3913	3938
61	B	4004	4184	6592	6621	7837	8174	8199
62	B	4040	4220	6628	6657	7873	8210	8235
63	B	4076	4256	6664	6693	7909	8246	8271
64	B	4112	4292	6700	6729	7945	8282	8307
65	B	4148	4328	6736	6765	7981	8318	8343
66	B	4184	4364	6772	6801	8017	8354	8379
67	B	4220	4400	6808	6837	8053	8390	8415
68	B	4256	4436	6844	6873	8089	8426	8451
69	B	4292	4472	6880	6909	8174	8511	8536
70	B	4328	4508	6916	6945	8210	8547	8572
71	B	4364	4544	6952	6981	8246	8583	8608
72	B	4400	4580	6988	7017	8282	8619	8644
73	B	4436	4616	7024	7053	8318	8655	8680

74	B	4472	4652	7060	7089	8354	8691	8716
75	B	4508	4688	7096	7125	8390	8727	8752
76	B	4544	4724	7132	7161	8426	8763	8788
77	B	4580	4760	7168	7197	8511	8848	8873
78	B	4616	4796	7204	7233	8547	8884	8909
79	B	4652	4832	7240	7269	8583	8920	8945
80	B	4688	4868	7276	7305	8619	8956	8981
81	B	4724	4904	7312	7341	8655	8992	9017
82	B	4760	4940	7348	7377	8691	9028	9053
83	B	4796	4976	7384	7413	8727	9064	9089
84	B	4832	5012	7420	7449	8763	9100	9125
85	B	4868	5048	7456	7485	8848	9185	9210
86	B	4904	5084	7492	7521	8884	9221	9246
87	B	4940	5120	7528	7557	8920	9257	9282
88	B	4976	5156	7564	7593	8956	9293	9318
89	B	5012	5192	7600	7629	8992	9329	9354
90	B	5048	5228	7636	7665	9028	9365	9390
91	B	5084	5332	7740	7769	9064	9401	9426
92	B	5120	5368	7776	7805	9100	9437	9462
93	B	5156	5404	7812	7841	9185	9522	9547
94	B	5192	5440	7848	7877	9221	9558	9583
95	B	5228	5476	7884	7913	9257	9594	9619
96	B	5332	5512	7920	7949	9293	9630	9655
97	B	5368	5548	7956	7985	9329	9666	9691
98	B	5404	5584	7992	8021	9365	9702	9727
99	B	5440	5620	8028	8057	9401	9738	9763
100	B	5476	5656	8064	8093	9437	9774	9799
101	B	5512	6592	9000	9029	10245	10582	10607
102	B	5548	6628	9036	9065	10281	10618	10643
103	B	5584	6664	9072	9101	10317	10654	10679
104	B	5620	6700	9108	9137	10353	10690	10715
105	B	5656	6736	9144	9173	10389	10726	10751
106	B	6592	6772	9180	9209	10425	10762	10787
107	B	6628	6808	9216	9245	10461	10798	10823
108	B	6664	6844	9252	9281	10497	10834	10859
109	B	6700	6880	9288	9317	10582	10919	10944
110	B	6736	6916	9324	9353	10618	10955	10980

111	C	2136	2316	3824	3853	4469	4806	4831
112	C	2172	2352	3860	3889	4505	4842	4867
113	C	2208	2388	3896	3925	4541	4878	4903
114	C	2244	2424	3932	3961	4577	4914	4939
115	C	2280	2460	3968	3997	4613	4950	4975
116	C	2316	2496	4004	4033	4649	4986	5011
117	C	2352	2532	4040	4069	4685	5022	5047
118	C	2388	2568	4076	4105	4721	5058	5083
119	C	2424	2604	4112	4141	4806	5143	5168
120	C	2460	2640	4148	4177	4842	5179	5204
121	C	2496	2676	4184	4213	4878	5215	5240
122	C	2532	2712	4220	4249	4914	5251	5276
123	C	2568	2748	4256	4285	4950	5287	5312
124	C	2604	2784	4292	4321	4986	5323	5348
125	C	2640	2820	4328	4357	5022	5359	5384
126	C	2676	2856	4364	4393	5058	5395	5420
127	C	2712	2892	4400	4429	5143	5480	5505
128	C	2748	2928	4436	4465	5179	5516	5541
129	C	2784	2964	4472	4501	5215	5552	5577
130	C	2820	3000	4508	4537	5251	5588	5613
131	C	2856	3036	4544	4573	5287	5624	5649
132	C	2892	3072	4580	4609	5323	5660	5685
133	C	2928	3108	4616	4645	5359	5696	5721
134	C	2964	3144	4652	4681	5395	5732	5757
135	C	3000	3180	4688	4717	5480	5817	5842
136	C	3036	3216	4724	4753	5516	5853	5878
137	C	3072	3252	4760	4789	5552	5889	5914
138	C	3108	3288	4796	4825	5588	5925	5950
139	C	3144	3324	4832	4861	5624	5961	5986
140	C	3180	3360	4868	4897	5660	5997	6022
141	C	3216	3396	4904	4933	5696	6033	6058
142	C	3252	3432	4940	4969	5732	6069	6094
143	C	3288	3468	4976	5005	5817	6154	6179
144	C	3324	3504	5012	5041	5853	6190	6215
145	C	3360	3540	5048	5077	5889	6226	6251
146	C	3396	3576	5084	5113	5925	6262	6287
147	C	3432	3612	5120	5149	5961	6298	6323

148	C	3468	3648	5156	5185	5997	6334	6359
149	C	3504	3684	5192	5221	6033	6370	6395
150	C	3540	3720	5228	5257	6069	6406	6431
151	C	3576	3824	5332	5361	6154	6491	6516
152	C	3612	3860	5368	5397	6190	6527	6552
153	C	3648	3896	5404	5433	6226	6563	6588
154	C	3684	3932	5440	5469	6262	6599	6624
155	C	3720	3968	5476	5505	6298	6635	6660
156	C	3824	4004	5512	5541	6334	6671	6696
157	C	3860	4040	5548	5577	6370	6707	6732
158	C	3896	4076	5584	5613	6406	6743	6768
159	C	3932	4112	5620	5649	6491	6828	6853
160	C	3968	4148	5656	5685	6527	6864	6889

附录 D: 问题 4 算例求解完整结果

表 19 问题 4 算例求解完整结果

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
A	0	156	764	793	1169	1506	1531
A	30	192	800	829	1205	1542	1567
A	60	228	836	865	1241	1578	1603
A	90	264	872	901	1277	1614	1639
A	120	300	908	937	1313	1650	1675
A	156	336	944	973	1349	1686	1711
A	192	372	980	1009	1385	1722	1747
A	228	408	1016	1045	1421	1758	1783
A	264	444	1052	1081	1506	1843	1868
A	300	480	1088	1117	1542	1879	1904
A	336	516	1124	1153	1578	1915	1940
A	372	552	1160	1189	1614	1951	1976
A	408	588	1196	1225	1650	1987	2012
A	444	624	1232	1261	1686	2023	2048
A	480	660	1268	1297	1722	2059	2084
A	516	696	1304	1333	1758	2095	2120
A	552	732	1340	1369	1843	2180	2205
A	588	768	1376	1405	1879	2216	2241
A	624	804	1412	1441	1915	2252	2277
A	660	840	1448	1477	1951	2288	2313
A	696	876	1484	1513	1987	2324	2349
A	732	912	1520	1549	2023	2360	2385
A	768	948	1556	1585	2059	2396	2421
A	804	984	1592	1621	2095	2432	2457
A	840	1020	1628	1657	2180	2517	2542
A	876	1056	1664	1693	2216	2553	2578
A	912	1092	1700	1729	2252	2589	2614
A	948	1128	1736	1765	2288	2625	2650
A	984	1164	1772	1801	2324	2661	2686
A	1020	1200	1808	1837	2360	2697	2722
A	1056	1236	1844	1873	2396	2733	2758
A	1092	1272	1880	1909	2432	2769	2794
A	1128	1308	1916	1945	2517	2854	2879

A	1164	1344	1952	1981	2553	2890	2915
A	1200	1380	1988	2017	2589	2926	2951
A	1236	1416	2024	2053	2625	2962	2987
A	1272	1452	2060	2089	2661	2998	3023
A	1308	1488	2096	2125	2697	3034	3059
A	1344	1524	2132	2161	2733	3070	3095
A	1380	1560	2168	2197	2769	3106	3131
A	1416	1596	2204	2233	2854	3191	3216
A	1452	1632	2240	2269	2890	3227	3252
A	1488	1668	2276	2305	2926	3263	3288
A	1524	1704	2312	2341	2962	3299	3324
A	1560	1740	2348	2377	2998	3335	3360
A	1596	1776	2384	2413	3034	3371	3396
A	1632	1812	2420	2449	3070	3407	3432
A	1668	1848	2456	2485	3106	3443	3468
A	1704	1884	2492	2521	3191	3528	3553
A	1740	1920	2528	2557	3227	3564	3589
A	1776	1956	2564	2593	3263	3600	3625
A	1812	1992	2600	2629	3299	3636	3661
A	1848	2028	2636	2665	3335	3672	3697
A	1884	2064	2672	2701	3371	3708	3733
A	1920	2100	2708	2737	3407	3744	3769
A	1956	2136	2744	2773	3443	3780	3805
A	1992	2172	2780	2809	3528	3865	3890
A	2028	2208	2816	2845	3564	3901	3926
A	2064	2244	2852	2881	3600	3937	3962
A	2100	2280	2888	2917	3636	3973	3998
A	2136	2316	2924	2953	3672	4009	4034
A	2172	2352	2960	2989	3708	4045	4070
A	2208	2388	2996	3025	3744	4081	4106
A	2244	2424	3032	3061	3780	4117	4142
A	2280	2460	3068	3097	3865	4202	4227
A	2316	2496	3104	3133	3901	4238	4263
A	2352	2532	3140	3169	3937	4274	4299
A	2388	2568	3176	3205	3973	4310	4335
A	2424	2604	3212	3241	4009	4346	4371

A	2460	2640	3248	3277	4045	4382	4407
A	2496	2676	3284	3313	4081	4418	4443
A	2532	2712	3320	3349	4117	4454	4479
A	2568	2748	3356	3385	4202	4539	4564
A	2604	2784	3392	3421	4238	4575	4600
A	2640	2820	3428	3457	4274	4611	4636
A	2676	2856	3464	3493	4310	4647	4672
A	2712	2892	3500	3529	4346	4683	4708
A	2748	2928	3536	3565	4382	4719	4744
A	2784	2964	3572	3601	4418	4755	4780
A	2820	3000	3608	3637	4454	4791	4816
A	2856	3036	3644	3673	4539	4876	4901
A	2892	3072	3680	3709	4575	4912	4937
A	2928	3108	3716	3745	4611	4948	4973
A	2964	3144	3752	3781	4647	4984	5009
A	3000	3180	3788	3817	4683	5020	5045
A	3036	3216	3824	3853	4719	5056	5081
A	3072	3252	3860	3889	4755	5092	5117
A	3108	3288	3896	3925	4791	5128	5153
A	3144	3324	3932	3961	4876	5213	5238
A	3180	3360	3968	3997	4912	5249	5274
A	3216	3396	4004	4033	4948	5285	5310
A	3252	3432	4040	4069	4984	5321	5346
A	3288	3468	4076	4105	5020	5357	5382
A	3324	3504	4112	4141	5056	5393	5418
A	3360	3540	4148	4177	5092	5429	5454
A	3396	3576	4184	4213	5128	5465	5490
A	3432	3612	4220	4249	5213	5550	5575
A	3468	3648	4256	4285	5249	5586	5611
A	3504	3684	4292	4321	5285	5622	5647
A	3540	3720	4328	4357	5321	5658	5683
A	3576	3756	4364	4393	5357	5694	5719
A	3612	3792	4400	4429	5393	5730	5755
A	3648	3828	4436	4465	5429	5766	5791
A	3684	3864	4472	4501	5465	5802	5827
A	3720	3900	4508	4537	5550	5887	5912

A	3756	3936	4544	4573	5586	5923	5948
A	3792	3972	4580	4609	5622	5959	5984
A	3828	4008	4616	4645	5658	5995	6020
A	3864	4044	4652	4681	5694	6031	6056
A	3900	4080	4688	4717	5730	6067	6092
A	3936	4116	4724	4753	5766	6103	6128
A	3972	4152	4760	4789	5802	6139	6164
A	4008	4188	4796	4825	5887	6224	6249
A	4044	4224	4832	4861	5923	6260	6285
A	4080	4260	4868	4897	5959	6296	6321
A	4116	4296	4904	4933	5995	6332	6357
A	4152	4332	4940	4969	6031	6368	6393
A	4188	4368	4976	5005	6067	6404	6429
A	4224	4404	5012	5041	6103	6440	6465
A	4260	4440	5048	5077	6139	6476	6501
A	4296	4476	5084	5113	6224	6561	6586
A	4332	4512	5120	5149	6260	6597	6622
A	4368	4548	5156	5185	6296	6633	6658
A	4404	4584	5192	5221	6332	6669	6694
A	4440	4620	5228	5257	6368	6705	6730
A	4476	4656	5264	5293	6404	6741	6766
A	4512	4692	5300	5329	6440	6777	6802
A	4548	4728	5336	5365	6476	6813	6838
A	4584	4764	5372	5401	6561	6898	6923
A	4620	4800	5408	5437	6597	6934	6959
A	4656	4836	5444	5473	6633	6970	6995
A	4692	4872	5480	5509	6669	7006	7031
A	4728	4908	5516	5545	6705	7042	7067
A	4764	4944	5552	5581	6741	7078	7103
A	4800	4980	5588	5617	6777	7114	7139
A	4836	5016	5624	5653	6813	7150	7175
A	4872	5052	5660	5689	6898	7235	7260
A	4908	5088	5696	5725	6934	7271	7296
A	4944	5124	5732	5761	6970	7307	7332
A	4980	5160	5768	5797	7006	7343	7368
A	5016	5196	5804	5833	7042	7379	7404
A	5052	5232	5840	5869	7078	7415	7440

A	5088	5268	5876	5905	7114	7451	7476
A	5124	5304	5912	5941	7150	7487	7512
A	5160	5340	5948	5977	7235	7572	7597
A	5196	5376	5984	6013	7271	7608	7633
A	5232	5412	6020	6049	7307	7644	7669
A	5268	5448	6056	6085	7343	7680	7705
A	5304	5484	6092	6121	7379	7716	7741
A	5340	5520	6128	6157	7415	7752	7777
A	5376	5556	6164	6193	7451	7788	7813
A	5412	5592	6200	6229	7487	7824	7849
A	5448	5628	6236	6265	7572	7909	7934
A	5484	5664	6272	6301	7608	7945	7970
A	5520	5700	6308	6337	7644	7981	8006
A	5556	5736	6344	6373	7680	8017	8042
A	5592	5772	6380	6409	7716	8053	8078
A	5628	5808	6416	6445	7752	8089	8114
A	5664	5844	6452	6481	7788	8125	8150
A	5700	5880	6488	6517	7824	8161	8186
B	8724	8904	10712	10741	11657	11994	12019
B	8760	8940	10748	10777	11693	12030	12055
B	8796	8976	10784	10813	11729	12066	12091
B	8832	9012	10820	10849	11765	12102	12127
B	8868	9048	10856	10885	11801	12138	12163
B	8904	9084	10892	10921	11837	12174	12199
B	8940	9120	10928	10957	11873	12210	12235
B	8976	9156	10964	10993	11909	12246	12271
B	9012	9192	11000	11029	11994	12331	12356
B	9048	9228	11036	11065	12030	12367	12392
B	9084	9264	11072	11101	12066	12403	12428
B	9120	9300	11108	11137	12102	12439	12464
B	9156	9336	11144	11173	12138	12475	12500
B	9192	9372	11180	11209	12174	12511	12536
B	9228	9408	11216	11245	12210	12547	12572
B	9264	9444	11252	11281	12246	12583	12608
B	9300	9480	11288	11317	12331	12668	12693
B	9336	9516	11324	11353	12367	12704	12729
B	9372	9552	11360	11389	12403	12740	12765

B	9408	9588	11396	11425	12439	12776	12801
B	9444	9624	11432	11461	12475	12812	12837
B	9480	9660	11468	11497	12511	12848	12873
B	9516	9696	11504	11533	12547	12884	12909
B	9552	9732	11540	11569	12583	12920	12945
B	9588	9768	11576	11605	12668	13005	13030
B	9624	9804	11612	11641	12704	13041	13066
B	9660	9840	11648	11677	12740	13077	13102
B	9696	9876	11684	11713	12776	13113	13138
B	9732	9912	11720	11749	12812	13149	13174
B	9768	9948	11756	11785	12848	13185	13210
B	9804	9984	11792	11821	12884	13221	13246
B	9840	10020	11828	11857	12920	13257	13282
B	9876	10056	11864	11893	13005	13342	13367
B	9912	10092	11900	11929	13041	13378	13403
B	9948	10128	11936	11965	13077	13414	13439
B	9984	10164	11972	12001	13113	13450	13475
B	10020	10200	12008	12037	13149	13486	13511
B	10056	10236	12044	12073	13185	13522	13547
B	10092	10272	12080	12109	13221	13558	13583
B	10128	10308	12116	12145	13257	13594	13619
B	10164	10712	12520	12549	13465	13802	13827
B	10200	10748	12556	12585	13501	13838	13863
B	10236	10784	12592	12621	13537	13874	13899
B	10272	10820	12628	12657	13573	13910	13935
B	10308	10856	12664	12693	13609	13946	13971
B	10712	10892	12700	12729	13645	13982	14007
B	10748	10928	12736	12765	13681	14018	14043
B	10784	10964	12772	12801	13717	14054	14079
B	10820	11000	12808	12837	13802	14139	14164
C	5736	5916	7124	7153	7909	8246	8271
C	5772	5952	7160	7189	7945	8282	8307
C	5808	5988	7196	7225	7981	8318	8343
C	5844	6024	7232	7261	8017	8354	8379
C	5880	6060	7268	7297	8053	8390	8415
C	5916	6096	7304	7333	8089	8426	8451
C	5952	6132	7340	7369	8125	8462	8487

C	5988	6168	7376	7405	8161	8498	8523
C	6024	6204	7412	7441	8246	8583	8608
C	6060	6240	7448	7477	8282	8619	8644
C	6096	6276	7484	7513	8318	8655	8680
C	6132	6312	7520	7549	8354	8691	8716
C	6168	6348	7556	7585	8390	8727	8752
C	6204	6384	7592	7621	8426	8763	8788
C	6240	6420	7628	7657	8462	8799	8824
C	6276	6456	7664	7693	8498	8835	8860
C	6312	6492	7700	7729	8583	8920	8945
C	6348	6528	7736	7765	8619	8956	8981
C	6384	6564	7772	7801	8655	8992	9017
C	6420	6600	7808	7837	8691	9028	9053
C	6456	6636	7844	7873	8727	9064	9089
C	6492	6672	7880	7909	8763	9100	9125
C	6528	6708	7916	7945	8799	9136	9161
C	6564	6744	7952	7981	8835	9172	9197
C	6600	6780	7988	8017	8920	9257	9282
C	6636	6816	8024	8053	8956	9293	9318
C	6672	6852	8060	8089	8992	9329	9354
C	6708	6888	8096	8125	9028	9365	9390
C	6744	6924	8132	8161	9064	9401	9426
C	6780	6960	8168	8197	9100	9437	9462
C	6816	6996	8204	8233	9136	9473	9498
C	6852	7032	8240	8269	9172	9509	9534
C	6888	7068	8276	8305	9257	9594	9619
C	6924	7104	8312	8341	9293	9630	9655
C	6960	7140	8348	8377	9329	9666	9691
C	6996	7176	8384	8413	9365	9702	9727
C	7032	7212	8420	8449	9401	9738	9763
C	7068	7248	8456	8485	9437	9774	9799
C	7104	7284	8492	8521	9473	9810	9835
C	7140	7320	8528	8557	9509	9846	9871
C	7176	7356	8564	8593	9594	9931	9956
C	7212	7392	8600	8629	9630	9967	9992
C	7248	7428	8636	8665	9666	10003	10028
C	7284	7464	8672	8701	9702	10039	10064

C	7320	7500	8708	8737	9738	10075	10100
C	7356	7536	8744	8773	9774	10111	10136
C	7392	7572	8780	8809	9810	10147	10172
C	7428	7608	8816	8845	9846	10183	10208
C	7464	7644	8852	8881	9931	10268	10293
C	7500	7680	8888	8917	9967	10304	10329
C	7536	7716	8924	8953	10003	10340	10365
C	7572	7752	8960	8989	10039	10376	10401
C	7608	7788	8996	9025	10075	10412	10437
C	7644	7824	9032	9061	10111	10448	10473
C	7680	7860	9068	9097	10147	10484	10509
C	7716	7896	9104	9133	10183	10520	10545
C	7752	7932	9140	9169	10268	10605	10630
C	7788	7968	9176	9205	10304	10641	10666
C	7824	8004	9212	9241	10340	10677	10702
C	7860	8040	9248	9277	10376	10713	10738
C	7896	8076	9284	9313	10412	10749	10774
C	7932	8112	9320	9349	10448	10785	10810
C	7968	8148	9356	9385	10484	10821	10846
C	8004	8184	9392	9421	10520	10857	10882
C	8040	8220	9428	9457	10605	10942	10967
C	8076	8256	9464	9493	10641	10978	11003
C	8112	8292	9500	9529	10677	11014	11039
C	8148	8328	9536	9565	10713	11050	11075
C	8184	8364	9572	9601	10749	11086	11111
C	8220	8400	9608	9637	10785	11122	11147
C	8256	8436	9644	9673	10821	11158	11183
C	8292	8472	9680	9709	10857	11194	11219
C	8328	8508	9716	9745	10942	11279	11304
C	8364	8544	9752	9781	10978	11315	11340
C	8400	8580	9788	9817	11014	11351	11376
C	8436	8616	9824	9853	11050	11387	11412
C	8472	8652	9860	9889	11086	11423	11448
C	8508	8688	9896	9925	11122	11459	11484
C	8544	8724	9932	9961	11158	11495	11520
C	8580	8760	9968	9997	11194	11531	11556
C	8616	8796	10004	10033	11279	11616	11641

C	8652	8832	10040	10069	11315	11652	11677
C	8688	8868	10076	10105	11351	11688	11713

附录 E: 问题 5 完整结果

表 20 问题 5.先有 A、B 型芯片后有 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	A	0	156	764	793	1109	1446	1471
2	A	30	192	800	829	1145	1482	1507
3	A	60	228	836	865	1181	1518	1543
4	A	90	264	872	901	1217	1554	1579
5	A	120	300	908	937	1253	1590	1615
6	A	156	336	944	973	1289	1626	1651
7	A	192	372	980	1009	1325	1662	1687
8	A	228	408	1016	1045	1361	1698	1723
9	A	264	444	1052	1081	1446	1783	1808
10	A	300	480	1088	1117	1482	1819	1844
11	A	336	516	1124	1153	1518	1855	1880
12	A	372	552	1160	1189	1554	1891	1916
13	A	408	588	1196	1225	1590	1927	1952
14	A	444	624	1232	1261	1626	1963	1988
15	A	480	660	1268	1297	1662	1999	2024
16	A	516	696	1304	1333	1698	2035	2060
17	A	552	732	1340	1369	1783	2120	2145
18	A	588	768	1376	1405	1819	2156	2181
19	A	624	804	1412	1441	1855	2192	2217
20	A	660	840	1448	1477	1891	2228	2253
21	A	696	876	1484	1513	1927	2264	2289
22	A	732	912	1520	1549	1963	2300	2325
23	A	768	948	1556	1585	1999	2336	2361
24	A	804	984	1592	1621	2035	2372	2397
25	A	840	1020	1628	1657	2120	2457	2482
26	A	876	1056	1664	1693	2156	2493	2518
27	A	912	1092	1700	1729	2192	2529	2554
28	A	948	1128	1736	1765	2228	2565	2590
29	A	984	1164	1772	1801	2264	2601	2626

30	A	1020	1200	1808	1837	2300	2637	2662
31	A	1056	1236	1844	1873	2336	2673	2698
32	A	1092	1272	1880	1909	2372	2709	2734
33	A	1128	1308	1916	1945	2457	2794	2819
34	A	1164	1344	1952	1981	2493	2830	2855
35	A	1200	1380	1988	2017	2529	2866	2891
36	A	1236	1416	2024	2053	2565	2902	2927
37	A	1272	1452	2060	2089	2601	2938	2963
38	A	1308	1488	2096	2125	2637	2974	2999
39	A	1344	1524	2132	2161	2673	3010	3035
40	A	1380	1560	2168	2197	2709	3046	3071
41	A	1416	1596	2204	2233	2794	3131	3156
42	A	1452	1632	2240	2269	2830	3167	3192
43	A	1488	1668	2276	2305	2866	3203	3228
44	A	1524	1704	2312	2341	2902	3239	3264
45	A	1560	1740	2348	2377	2938	3275	3300
46	A	1596	1776	2384	2413	2974	3311	3336
47	A	1632	1812	2420	2449	3010	3347	3372
48	A	1668	1848	2456	2485	3046	3383	3408
49	A	1704	1884	2492	2521	3131	3468	3493
50	A	1740	1920	2528	2557	3167	3504	3529
51	A	1776	1956	2564	2593	3203	3540	3565
52	A	1812	1992	2600	2629	3239	3576	3601
53	A	1848	2028	2636	2665	3275	3612	3637
54	A	1884	2064	2672	2701	3311	3648	3673
55	A	1920	2100	2708	2737	3347	3684	3709
56	A	1956	2136	2744	2773	3383	3720	3745
57	A	1992	2172	2780	2809	3468	3805	3830
58	A	2028	2208	2816	2845	3504	3841	3866
59	A	2064	2244	2852	2881	3540	3877	3902
60	A	2100	2280	2888	2917	3576	3913	3938
61	B	3216	3396	5804	5833	7049	7386	7411
62	B	3252	3432	5840	5869	7085	7422	7447
63	B	3288	3468	5876	5905	7121	7458	7483
64	B	3324	3504	5912	5941	7157	7494	7519
65	B	3360	3540	5948	5977	7193	7530	7555
66	B	3396	3576	5984	6013	7229	7566	7591

67	B	3432	3612	6020	6049	7265	7602	7627
68	B	3468	3648	6056	6085	7301	7638	7663
69	B	3504	3684	6092	6121	7386	7723	7748
70	B	3540	3720	6128	6157	7422	7759	7784
71	B	3576	3824	6232	6261	7477	7814	7839
72	B	3612	3860	6268	6297	7513	7850	7875
73	B	3648	3896	6304	6333	7549	7886	7911
74	B	3684	3932	6340	6369	7585	7922	7947
75	B	3720	3968	6376	6405	7621	7958	7983
76	B	3824	4004	6412	6441	7657	7994	8019
77	B	3860	4040	6448	6477	7723	8060	8085
78	B	3896	4076	6484	6513	7759	8096	8121
79	B	3932	4112	6520	6549	7814	8151	8176
80	B	3968	4148	6556	6585	7850	8187	8212
81	B	4004	4184	6592	6621	7886	8223	8248
82	B	4040	4220	6628	6657	7922	8259	8284
83	B	4076	4256	6664	6693	7958	8295	8320
84	B	4112	4292	6700	6729	7994	8331	8356
85	B	4148	4328	6736	6765	8060	8397	8422
86	B	4184	4364	6772	6801	8096	8433	8458
87	B	4220	4400	6808	6837	8151	8488	8513
88	B	4256	4436	6844	6873	8187	8524	8549
89	B	4292	4472	6880	6909	8223	8560	8585
90	B	4328	4508	6916	6945	8259	8596	8621
91	B	4364	4544	6952	6981	8295	8632	8657
92	B	4400	4580	6988	7017	8331	8668	8693
93	B	4436	4616	7024	7053	8397	8734	8759
94	B	4472	4652	7060	7089	8433	8770	8795
95	B	4508	4688	7096	7125	8488	8825	8850
96	B	4544	4724	7132	7161	8524	8861	8886
97	B	4580	4760	7168	7197	8560	8897	8922
98	B	4616	4796	7204	7233	8596	8933	8958
99	B	4652	4832	7240	7269	8632	8969	8994
100	B	4688	4868	7276	7305	8668	9005	9030
101	B	4724	5804	8212	8241	9457	9794	9819
102	B	4760	5840	8248	8277	9493	9830	9855
103	B	4796	5876	8284	8313	9529	9866	9891

104	B	4832	5912	8320	8349	9565	9902	9927
105	B	4868	5948	8356	8385	9601	9938	9963
106	B	5804	5984	8392	8421	9637	9974	9999
107	B	5840	6020	8428	8457	9673	10010	10035
108	B	5876	6056	8464	8493	9709	10046	10071
109	B	5912	6092	8500	8529	9794	10131	10156
110	B	5948	6128	8536	8565	9830	10167	10192
111	B	5984	6232	8640	8669	9885	10222	10247
112	B	6020	6268	8676	8705	9921	10258	10283
113	B	6056	6304	8712	8741	9957	10294	10319
114	B	6092	6340	8748	8777	9993	10330	10355
115	B	6128	6376	8784	8813	10029	10366	10391
116	B	6232	6412	8820	8849	10065	10402	10427
117	B	6268	6448	8856	8885	10131	10468	10493
118	B	6304	6484	8892	8921	10167	10504	10529
119	B	6340	6520	8928	8957	10222	10559	10584
120	B	6376	6556	8964	8993	10258	10595	10620
121	B	6412	6592	9000	9029	10294	10631	10656
122	B	6448	6628	9036	9065	10330	10667	10692
123	B	6484	6664	9072	9101	10366	10703	10728
124	B	6520	6700	9108	9137	10402	10739	10764
125	B	6556	6736	9144	9173	10468	10805	10830
126	B	6592	6772	9180	9209	10504	10841	10866
127	B	6628	6808	9216	9245	10559	10896	10921
128	B	6664	6844	9252	9281	10595	10932	10957
129	B	6700	6880	9288	9317	10631	10968	10993
130	B	6736	6916	9324	9353	10667	11004	11029
131	C	2136	2316	3824	3853	4469	4806	4831
132	C	2172	2352	3860	3889	4505	4842	4867
133	C	2208	2388	3896	3925	4541	4878	4903
134	C	2244	2424	3932	3961	4577	4914	4939
135	C	2280	2460	3968	3997	4613	4950	4975
136	C	2316	2496	4004	4033	4649	4986	5011
137	C	2352	2532	4040	4069	4685	5022	5047
138	C	2388	2568	4076	4105	4721	5058	5083
139	C	2424	2604	4112	4141	4806	5143	5168
140	C	2460	2640	4148	4177	4842	5179	5204

141	C	2496	2676	4184	4213	4878	5215	5240
142	C	2532	2712	4220	4249	4914	5251	5276
143	C	2568	2748	4256	4285	4950	5287	5312
144	C	2604	2784	4292	4321	4986	5323	5348
145	C	2640	2820	4328	4357	5022	5359	5384
146	C	2676	2856	4364	4393	5058	5395	5420
147	C	2712	2892	4400	4429	5143	5480	5505
148	C	2748	2928	4436	4465	5179	5516	5541
149	C	2784	2964	4472	4501	5215	5552	5577
150	C	2820	3000	4508	4537	5251	5588	5613
151	C	2856	3036	4544	4573	5287	5624	5649
152	C	2892	3072	4580	4609	5323	5660	5685
153	C	2928	3108	4616	4645	5359	5696	5721
154	C	2964	3144	4652	4681	5395	5732	5757
155	C	3000	3180	4688	4717	5480	5817	5842
156	C	3036	3216	4724	4753	5516	5853	5878
157	C	3072	3252	4760	4789	5552	5889	5914
158	C	3108	3288	4796	4825	5588	5925	5950
159	C	3144	3324	4832	4861	5624	5961	5986
160	C	3180	3360	4868	4897	5660	5997	6022

附录 F: 支撑材料列表

支撑材料列表

序号	文件名	材料说明
1	GA for JSP python	源代码
2	Result	问题的全部求解结果

附录 G: 主要程序/关键代码

代 码 环 境	操作系统: Windows 11 编程语言: Python 3.11.3 编辑器: Visual Code 代码详见: GA for JSP python 文件夹
------------------	--

代码清单 1 GA_for_JSP_1.py

```

from random import (randint)
from typing import (List, Tuple, Set)
from read_data import read_data
from reshape_data import reshape_data
from collections import namedtuple
import matplotlib.pyplot as plt
from write_result_list import write_data_to_excel
import numpy as np
## 保证 aa+bb>0
aa =100
bb = 0
MATRIX_SIZE_I = aa+bb
MATRIX_SIZE_J = 10
ROW_MACHINE_NUM = 40
WASH_MACHINE_NUM = 8

# 个体对象，染色体和适应度
class Gene(object):
    def __init__(self, fitness: float = 0, chromosome = None):
        self.fitness = fitness
        self.chromosome: list = chromosome

    def __eq__(self, other):
        if isinstance(other, Gene):
            return other.fitness == self.fitness and other.chromosome ==
self.chromosome
        return False

    def __hash__(self):
        return hash("".join(map(lambda x: str(x), self.chromosome)))

    def __str__(self):
        return "{} => {}".format(self.chromosome, self.fitness)

# 存储解码结果
class GeneEvaluation:
    def __init__(self):
        self.fulfill_time = 0 # 总检测时间
        self.machine_work_time = [0 for _ in range(MATRIX_SIZE_J)] #第 i
台机器的作业时间
        self.process_ids = [0 for _ in range(MATRIX_SIZE_I)] #第 i
个芯片当前的工序编号

```

代码清单 2 GA_for_JSP_2.py

```
from random import (randint, shuffle)
from typing import (List, Tuple, Set)
from read_data import read_data
from reshape_data import reshape_data
from collections import namedtuple
import matplotlib.pyplot as plt
from write_result_list import write_data_to_excel
import numpy as np
import json

##aa>0
##bb>0
aa = 80
bb = 80
MATRIX_SIZE_I = aa+bb
MATRIX_SIZE_J = 10
ROW_MACHINE_NUM = 40
WASH_MACHINE_NUM = 8

# 个体对象，染色体和适应度
class Gene(object):
    def __init__(self, fitness: float = 0, chromosome = None):
        self.fitness = fitness
        self.chromosome: list = chromosome

    def __eq__(self, other):
        if isinstance(other, Gene):
            return other.fitness == self.fitness and other.chromosome == self.chromosome
        return False

    def __hash__(self):
        return hash("".join(map(lambda x: str(x), self.chromosome)))

    def __str__(self):
        return "{} => {}".format(self.chromosome, self.fitness)

# 存储解码结果
class GeneEvaluation:
    def __init__(self):
        self.fulfill_time = 0 # 总检测时间
        self.machine_work_time = [0 for _ in range(MATRIX_SIZE_J)] #第 i
        台机器的作业时间
```

代码清单 3 GA_for_JSP_3.py

```
from random import (randint, shuffle)
from typing import (List, Tuple, Set, Dict, Any)
from read_data import read_data
from reshape_data import reshape_data
from collections import namedtuple
import matplotlib.pyplot as plt
from write_result_list import write_data_to_excel
import numpy as np

##aa>0
##bb>0
##cc>0
aa = 60
bb = 50
cc = 50
MATRIX_SIZE_I = aa+bb+cc
MATRIX_SIZE_J = 10
ROW_MACHINE_NUM = 40
WASH_MACHINE_NUM = 8
# 个体对象，染色体和适应度
class Gene(object):
    def __init__(self, fitness: float = 0, chromosome = None):
        self.fitness = fitness
        self.chromosome: list = chromosome

    def __eq__(self, other):
        if isinstance(other, Gene):
            return other.fitness == self.fitness and other.chromosome
            == self.chromosome
        return False

    def __hash__(self):
        return hash("".join(map(lambda x: str(x), self.chromosome)))

    def __str__(self):
        return "{} => {}".format(self.chromosome, self.fitness)

# 存储解码结果
class GeneEvaluation:
    def __init__(self):
        self.fulfill_time = 0 # 总检测时间
        self.machine_work_time = [0 for _ in range(MATRIX_SIZE_J)] #第 i
        台机器的作业时间
```


代码清单 4 融合 GA_for_JSP_4.py

```
from random import (randint, shuffle)
from typing import (List, Tuple, Set, Dict, Any)
from read_data import read_data
from reshape_data import reshape_data
from collections import namedtuple
import matplotlib.pyplot as plt
from write_result_list import write_data_to_excel
import numpy as np

##aa>0
##bb>0
##cc>0
aa = 60
bb = 50
cc = 50
MATRIX_SIZE_I = aa+bb+cc
MATRIX_SIZE_J = 10
ROW_MACHINE_NUM = 40
WASH_MACHINE_NUM = 8
# 个体对象，染色体和适应度
class Gene(object):
    def __init__(self, fitness: float = 0, chromosome = None):
        self.fitness = fitness
        self.chromosome: list = chromosome

    def __eq__(self, other):
        if isinstance(other, Gene):
            return other.fitness == self.fitness and other.chromosome
            == self.chromosome
        return False

    def __hash__(self):
        return hash(''.join(map(lambda x: str(x), self.chromosome)))

    def __str__(self):
        return "{} => {}".format(self.chromosome, self.fitness)

# 存储解码结果
class GeneEvaluation:
    def __init__(self):
        self.fulfill_time = 0 # 总检测时间
        self.machine_work_time = [0 for _ in range(MATRIX_SIZE_J)] #第 i
        台机器的作业时间
```


代码清单 5 GA_for_JSP_5.py

```
from random import (randint, shuffle)
from typing import (List, Tuple, Set, Dict, Any)
from read_data import read_data
from reshape_data import reshape_data
from collections import namedtuple
import matplotlib.pyplot as plt
from write_result_list import write_data_to_excel
import numpy as np

aa = 60
bb = 70
cc = 30
MATRIX_SIZE_I = aa+bb+cc
SIZE_f = aa+bb
MATRIX_SIZE_J = 10
ROW_MACHINE_NUM = 40
WASH_MACHINE_NUM = 8
# 个体对象，染色体和适应度
class Gene(object):
    def __init__(self, fitness: float = 0, chromosome = None):
        self.fitness = fitness
        self.chromosome: list = chromosome

    def __eq__(self, other):
        if isinstance(other, Gene):
            return other.fitness == self.fitness and other.chromosome
            == self.chromosome
        return False

    def __hash__(self):
        return hash("".join(map(lambda x: str(x), self.chromosome)))

    def __str__(self):
        return "{} => {}".format(self.chromosome, self.fitness)

# 存储解码结果
class GeneEvaluation:
    def __init__(self):
        self.fulfill_time = 0 # 总检测时间
        self.machine_work_time = [0 for _ in range(MATRIX_SIZE_J)] #第 i
        台机器的作业时间
        self.process_ids = [0 for _ in range(MATRIX_SIZE_I)] #第 i
        个芯片当前的工序编号
```

代码清单 6 read_data.py

```
from openpyxl import load_workbook

def read_data(path, sheetname=['芯片 A', '芯片 B', '芯片 C'], aa=0, bb=0, cc=0):
    book = load_workbook(path)
    sheeta = book[sheetname[0]]
    sheetb = book[sheetname[1]]
    sheetc = book[sheetname[2]]
    arows = sheeta.rows
    brows = sheetb.rows
    crows = sheetc.rows
    aheaders = [cell.value for cell in next(arows)]
    bheaders = [cell.value for cell in next(brows)]
    cheaders = [cell.value for cell in next(crows)]
    all_rows = []
    workpieces = []
    for row in arows:
        for j in range(1, aa+1):
            data = {}
            for title, cell in zip(ahheaders, row):
                data[title] = cell.value
            data['workpiece'] = str(j)+'_'+data['workpiece']
            all_rows.append(data)
            if not(data['workpiece'] in workpieces):
                workpieces.append(data['workpiece'])
            # print(data)
    for row in brows:
        for j in range(aa+1, aa+bb+1):
            data = {}
            for title, cell in zip(bheaders, row):
                data[title] = cell.value
            data['workpiece'] = str(j)+'_'+data['workpiece']
            all_rows.append(data)
            if not(data['workpiece'] in workpieces):
                workpieces.append(data['workpiece'])
    for row in crows:
        for j in range(aa+bb+1, aa+bb+cc+1):
            data = {}
            for title, cell in zip(chheaders, row):
                data[title] = cell.value
            data['workpiece'] = str(j)+'_'+data['workpiece']
            all_rows.append(data)
            if not(data['workpiece'] in workpieces):
                workpieces.append(data['workpiece'])
    return all_rows, workpieces
```

代码清单 7 reshape_data.py

```
from typing import (List, Dict)
from collections import namedtuple
from read_data import read_data

ReshapeData = namedtuple("ReshapeData",
                        ["result", "workpiece", "machine", "process",
                        "reverse_workpiece", "reverse_machine"])

def make_reverse_index(arr: list) -> dict:
    result = {}
    for i in range(len(arr)):
        result[arr[i]] = i
    return result

def filter_value(origin: list, except_value: int) -> list:
    return list(filter(lambda v: v != except_value, origin))

def reshape_data(data: List[Dict], workpieces) -> ReshapeData:
    def make_array(r: dict, workpieces) -> ReshapeData:
        workpieces = workpieces
        machines = ['前处理台1', '前处理台2', '前处理台3', '前处理台4', '前处理台5', '转盘1', '磁珠盒', '转盘2', '清洗盒', '检测盘']
        # machines = list(set(map(lambda v: v["machine"], data)))
        process = [-1 for _ in workpieces]
        # print(machines)
        reverse_workpieces = make_reverse_index(workpieces)
        reverse_machines = make_reverse_index(machines)
        ans = [-1 for _ in r.keys()]
        for key, val in r.items():
            # print(val, type(val))
            m = max(*map(lambda v: v["order"], val)) + 1 if len(val) >
1 else val[0]["order"]
            # print(m)
            t = [-1 for _ in range(m + 1)]
            x = [-1 for _ in range(m + 1)]
            for p in val:
                t[p["order"]] = (reverse_machines[p["machine"]],
p["time"])
                x[p["order"]] = p["process"]
            x = filter_value(x, -1)
            t = filter_value(t, -1)
            if ans[reverse_workpieces[key]] == -1:
                ans[reverse_workpieces[key]] = t
            else:
                ans[reverse_workpieces[key]].append(t)
```

代码清单 8 write_result_list.py

```
import pandas as pd

def write_data_to_excel( filename, data, fullfill):

    # 写入字典的键作为表头
    headers = ['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']

    data_w = []
    data_item = []
    for item in data:
        if item['工序名称'] == '前处理 1':
            data_item.append(item['芯片'][-1])
            data_item.append(item['开始时间'])
        elif item['工序名称'] == '第一次培育' or item['工序名称'] == '加磁'
        or \
            item['工序名称'] == '第二次培育' or item['工序名称'] == '清洗':

            data_item.append(item['开始时间'])
        elif item['工序名称'] == '检测':
            data_item.append(item['开始时间'])
            data_item.append(item['结束时间'])
            data_w.append(data_item)
            data_item = []

    df = pd.DataFrame(data_w,columns=headers)
    # df1=df.sort_values(by='进仓时间',ascending=True)

    # 写入字典的值
    df.to_excel(filename,sheet_name=str(fullfill))
```