

化学发光免疫分析仪运行的调度优化

摘要

可重入混合 Flow Shop 调度问题 (RHFSP) 是典型的 NP-hard 问题,为提高芯片生产力. 本文针对具有局部可重入特性的工艺流程, 基于决策论与动态规划的思想, 以最小化最大生产流程为目标, 运用基于时间与空间约束的线性规划模型结合改进后的遗传算法对若干单类或多类芯片的检测序列进行规划求解.

针对问题一: 对于单类芯片进入各工序的时间进行分析, 提高单类芯片生产力的关键在于求解单类芯片最快全部入仓的序列. 考虑到芯片处理的时间约束和工区容量的空间约束的影响, 本文建立了最小化最大流程时间等指标, 将动态调度问题转化为线性规划问题, 随后基于决策论和动态规划的思想, 建立了线性规划模型. 该模型充分考虑了芯片在两道温育工序中重复访问同一工区温育盘的影响, 对该模型进行求解, 求解结果如下:

- 情形 1: 单位时间 (1 小时) 内, 免疫分析仪能够检测 A 类型芯片 53 枚, 所需时间为 59.5 分钟. 检测 54 枚 A 类型芯片则需要 60.1 分钟, 芯片序列见附表 1.
- 情形 2: 免疫分析仪无法在 1 小时内完成对 1 枚 B 类型芯片的检测, 本题结果为 0 枚. 使用免疫分析仪检测 1 枚 B 类型芯片至少需要 69.5 分钟, 芯片序列见附表 2.

针对问题二至四: 在问题一的基础上, 芯片由单一类型转变为多种类型, 时间与空间约束的基本事实不变, 但由于温育时间存在差异, 芯片序列在加工过程中发生更改, 规划模型复杂度大大增加, 且由于技术限制, 多类芯片的动态调度改用基于智能搜索的遗传算法, 该模型充分考虑了芯片检测的等待时间约束, 并将温育盘按照一次温育时间在总温育时间中的占比进行分盘, 消除了“可重入”特性. 该模型依旧以最小化最大流程时间为指标, 迭代求解, 结果收敛至:

- 情形 1: A, B 两种类型的芯片各 80 片, 全部检测完至少需要 18105s, 芯片序列见附表 3.
- 情形 2: A 型芯片 60 片、B 型芯片 50 片、C 型芯片 50 片, 全部检测完至少需要 18205s, 芯片序列见附表 4.

此外, 对问题四, 模型不限于芯片类型、数量、温育时间, 可以修改参数后直接运用模型求解. 本文通过自定义四类芯片, 运用上述模型, 求得收敛解.

针对问题五: 在问题二的基础上, 在某一批次多类芯片检测的优化调度过程中发生一突发情况, 即有一新批次芯片的加入. 考虑到待检芯片集合发生变化, 本文以该突发情况的时刻开始对更新后的待检芯片集合再次运用上述模型, 求解新的最优序列和总最小化流程时间, 此时整个检测过程仍需满足时间和空间约束. 结果如下:

- 情形 1: 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 18939s, 芯片序列见附表 5.

最后, 我们对提出的模型进行全面的评价: 本文的模型贴合实际, 所使用的适用于柔性 Flow-Shop 调度问题的改进遗传算法具有很好的鲁棒性, 并行效率高, 该模型在其他连续化自动化作业仪器中也能使用. 未来可以进一步考虑可重入工区工位的柔性分配问题, 使得工位资源利用率和生产力进一步提高.

关键词: Flow Shop 调度问题 动态规划 遗传算法 可重入

目录

化学发光免疫分析仪运行的调度优化.....	I
摘要.....	I
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.....	1
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 模型建立.....	3
3.1 模型一: 单类线性规划模型.....	3
3.2 模型二: FFSP 的遗传算法优化.....	7
3.3 模型三: 多类型线性规划模型.....	11
3.4 模型四: 检测过程中的方案调整.....	13
4 模型求解.....	14
4.1 参数设置.....	14
4.2 问题一: 单类芯片生产力线性规划模型求解.....	14
4.3 问题二: A、B 类型芯片已知条件下的 FFSP 遗传算法模型求解.....	16
4.4 问题三: A、B、C 类型芯片已知条件下的 FFSP 遗传算法模型求解.....	17
4.5 问题四: 未知类型芯片条件下的 FFSP 遗传算法模型求解.....	18
4.6 问题五: 中途插入检测条件下的 FFSP 遗传算法模型求解.....	19
5 模型推广、评价与改进.....	22
5.1 模型推广.....	22
5.2 模型的优点.....	22
5.3 模型的不足.....	22
5.4 模型的改进.....	22
参考文献.....	23
附 录.....	24
附表 1: 问题 1.A 型芯片的调度结果 (单位: s).....	24
附表 2: 问题 1.B 型芯片的调度结果 (单位: s).....	26
附表 3: 问题 2.A、B 型芯片的调度结果 (单位: s).....	28
附表 4: 问题 3.A、B、C 型芯片的调度结果 (单位: s).....	32
附表 5: 问题 5. 先有 A、B 型芯片后有 C 型芯片的调度结果 (单位: s).....	36
附录 6: 主要程序/关键代码.....	40

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一种对血清、唾液等人体体液样本进行检验的仪器,通过对诸如蛋白、核酸等指标的检测,判断被检者是否患有某种疾病.该仪器使免疫诊断的整个流程实现全自动化,克服了传统人工检测效率低、精确度差等缺点,已被广泛地应用在临床上.

本文中,该分析仪以生物芯片为人体体液样本的载体.芯片的读码、激光超声、过滤、定量和 R1R2 加样归为前处理阶段,芯片的核心流程主要包括前处理、第一次温育、磁珠加样、第二次温育、清洗和检测六道工序.每道工序有其固定的处理工区,需要一定的时间.相邻两个工区间均存在一定时间的对位过程用于传输芯片.值得注意的是,温育和清洗工区存在并行工位,特别地,其中两次温育工序位于同一工区,共用一组并行工位.芯片的检测流程一旦开始便不能停止,直到检测完成.对于不同种类的芯片,其工艺流程相同,差异在于两次温育的所需时长.综上所述,本文可以视作可重入混合 Flow Shop 调度问题 (RHFSP).目前,已经有很多学者对可重入混合 Flow Shop 调度问题 (RHFSP) 进行了研究,也提出了很多算法来解决此类问题,求解 RHFSP 常用的方法有基于运筹学的数学规划法,基于启发式规则的调度方法,基于专家系统、人工神经网络等人工智能的方法,以及基于遗传算法、蚁群算法、粒子群算法等智能优化算法的方法等^[1].虽然这些算法可以快速得到调度方案,但其一般是针对多线程多机器的优化问题,对于同一个仪器分步骤进行的动态调度问题则需要针对性地进行解决,这也促使我们运用更好的方法建立更加合理的模型.

1.2 问题提出

基于本文背景的可重入混合 Flow Shop 调度问题 (RHFSP) 涉及多方面的问题,由芯片,工区,工位这几个要素构成.若干单类或多类芯片顺次通过前处理工作台、温床、磁珠加样盒、温育盒、清洗盒、检测盒共五个工区(前后两温育盒为同一工区),在各工区的工位上按照给定的加工时间进行加工.在满足下列 4 点约束下:

- 芯片的检测流程一旦开始便不能停止,直到检测完成;
- 芯片顺次进入各工序的起始时间存在特定制约关系;
- 相邻两个工序之间的对位传输有某一有限等待时间;
- 各工区基于自身容量即并行工位数,具有负荷上限,尤其地,温育工区将同时负荷处于两个不同温育工序的芯片.

需要从单位时间产量或最小化最大流程时间的角度考虑,解决以下 3 个问题:

- (1) 问题 1: 对给定温育时间的单类芯片进行生产力优化;
- (2) 问题 2: 对给定温育时间和数量的多类芯片进行最优序列求解和生产力优化;
- (3) 问题 3: 对两个批次给定温育时间和数量的多类芯片进行最优序列求解和生产力优化.

1.3 资料条件

【2023B.pdf】提供了化学发光免疫分析仪的工作原理和最优调度的方案需求,具体的各个工序对应所需时间和各个工序对应的卡槽位置总结如下:

表1 各个工序对应所需时间和各个工序对应的卡槽位置

工序	前处理	转盘转动对位	第一次温育	转盘对位磁珠盒	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
所需时间	150s	6s	0~60 min	8s	[21s, 25s]	8s	0~30 min	16s	>= 325s	12s	25s
个数	1		与第二次温育共计 40		1		与第一次温育共计 40		8		1

2 模型假设与符号说明

2.1 模型基本假设

- (1) 假设生物芯片在化学发光免疫分析仪的温育盘与其他工位 (前处理工作台、磁珠加样盒、清洗盒和检测盘) 之间的传递是瞬间完成的, 即转盘在题中所给时间内完成转动对位后, 芯片瞬间进入下一工位 (或从上一工位进入转盘), 而下一工序也随之立即启动.
- (2) 假定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.
- (3) 假设磁珠加样盒足够智能, 在 21~25s 的范围内, 对每枚芯片的加样时间是计算出的最优结果.
- (4) 假设转动对位过程必须在上一工序完成后才能开始, 无法提前对位, 即对每一芯片, 转动对位时间与各工序进行时间无法重合.
- (5) 假设前处理工作台、磁珠加样盒及清洗盒中均存在一个大容量“待机”区域, 芯片在完成前处理工作/加样/清洗工序后, 立即进入该“待机”区域等待转动对位完成, 不占用原工位而影响下一芯片的进入.

2.2 符号说明

本文定义了如下 26 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
T_0	芯片进仓时间	秒
T_1	第一次温育起始时间	秒
T_2	磁珠加样起始时间	秒

符号	含义	单位
T_3	第二次温育起始时间	秒
T_4	清洗起始时间	秒
T_5	检测起始时间	秒
T_6	检测结束时间	秒
a	第一次温育时间	分钟
b	第二次温育时间	分钟
N	某类型芯片数量	个
p	芯片序号	无
q	工序序号	无
M_i	并行工位数	个
A	FFSP 的编码矩阵	无
$Chrom_i$	第 i 条染色体	条
T	时间矩阵	无
S	机器矩阵	无
TT	相应芯片在机器上加工时间矩阵	无
ZT	完工时间矩阵	无
GT	重构矩阵	无
WT	回溯矩阵	无
U	工时向量重构矩阵	无
DT	非法等待矩阵	无
f	适值函数	无
P_i	第 i 条染色体被选择的概率	无
q_k	第 k 个个体的累积概率	无

3 模型建立

3.1 模型一：单类线性规划模型

3.1.1 问题分析

调度优化问题是在资源有限的情况下，将任务分配给可用资源以最大程度地提高效率和满足约束条件的问题。题目以化学发光免疫分析仪运行为背景，要求实现在保证各检测工序不发生冲突的前提下，使得特定数量、类型的芯片检测总时长尽可能缩短，或在特定时长内使得检测某类型芯片的数量尽可能增多。

问题一的特点在于一次仅对一种芯片进行检测，要求在单位时间（1 小时）内检测尽可能多的 A 或 B 类芯片。不同类型芯片之间的区别在于第一次温育、第二次温育的时长，如表 2 示。

显然，待检测芯片数为正整数，且越多的待检测芯片对应着越长的检测时间。未知的芯片数不利于编程实现，因此，将决策变量由最大芯片个数转换为最短检测时间，合理的假设芯片数量，能够使得求出的最短检测时间小于 1 小时的最大芯片个数即为所求。

表3 各类芯片温育时间 (单位: 分钟)

芯片类型	第一次温育时间	第二次温育时间
A	10	5
B	40	20
C	25	10

依据题意、结合文献得知, 免疫分析仪启动后, 首先对各个电机初始化, 如果发现某个电机阻塞则停止运行, 并给出具体的错误信息; 紧接着对仪器的管路及阀门系统初始化; 随后按照设定的程序对反应杯进行加样以及加试剂; 加样完毕后反应杯进入温育系统; 温育时间到达后将未参与反应的残留物洗脱^[1]. 基于以上信息, 可以认为仪器足够智能, 能够按照事先计算出的最优方案在最佳时间瞬间完成芯片的吞吐.

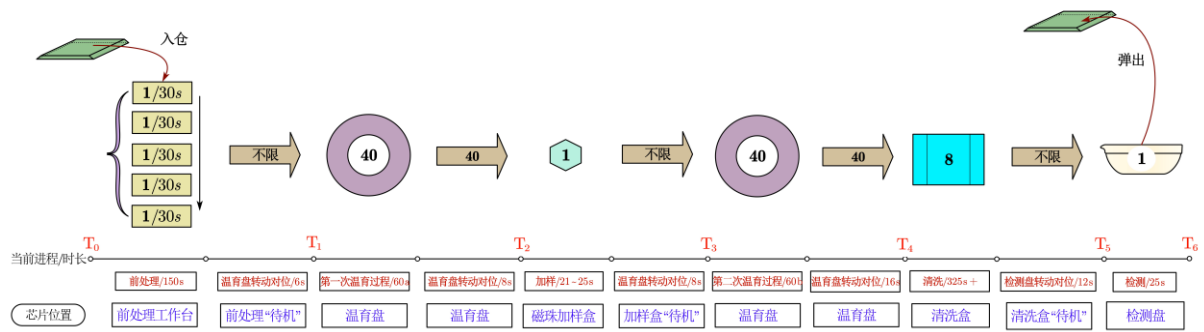


图1 化学发光免疫分析仪工作流程图, 其中黑色加粗数字为芯片当前位置的芯片容量

因此基于线性规划思想, 设置最后一枚芯片检测结束时间最小为目标, 基于免疫分析仪各工位容量和所需时间, 确定时间、空间约束条件, 从而建立规划模型, 求解多约束条件下的最优化问题.

3.1.2 模型指标定义

设芯片总数为 N . 为了方便约束条件的数学表述, 结合题目待求附表, 将芯片进入各工作区域的时间设置为变量, 其中序号为 p 的芯片进仓时间为 T_0^p ($p = 1, 2, \dots, N$), 检测结束时间为 T_6^p ($p = 1, 2, \dots, N$), 其余按步骤分别为 T_1 至 T_5 (见表 1).

3.1.3 目标函数

为了方便求解, 将决策变量由芯片总数 N 调整为总加工时间. 当第 N 枚芯片完成检测弹出时, 检测流程结束. 根据指标定义, 容易得到目标函数:

$$\min = T_6^N \quad (1)$$

即, 使得最后一枚芯片弹出时间最早.

3.1.4 时间约束条件

根据生产流程, 第 p 枚芯片进入各工位的时间之间有相互制约关系.

首先, 将第 1 枚芯片的进仓时间设置为 0 时刻, 即时间边界条件:

$$T_0^1 = 0 \quad (2)$$

芯片进入各个工作区的时间由进仓时间、上一工序持续时间和转动对位时间共同决定. 设第一、二次温育时间分别为 a 、 b , 根据流程图, 芯片 p 进入各流程的时间之间的关系可数学表示如下:

$$T_1^p = T_0^p + 150 + 6 \quad (3)$$

$$T_2^p = T_1^p + 60a + 8 \quad (4)$$

$$T_2^p + 21 + 8 \leq T_3^p \leq T_2^p + 25 + 8 \quad (5)$$

$$T_4^p = T_3^p + 60b + 16 \quad (6)$$

$$T_5^p \geq T_4^p + 325 + 12 \quad (7)$$

$$T_6^p = T_5^p + 25 \quad (8)$$

3.1.5 单个空间约束条件

根据所给条件, 免疫分析仪中各个工位容量是有限的, 即空间上存在互相制约关系. 本小节将六个流程分开, 分别讨论同一流程中, 由于工位容量而导致的芯片进入制约.

(1)考虑前处理过程. 芯片前期处理共 5 个步骤, 按顺序进行, 每个步骤一次只能处理一片芯片, 时长 30 秒, 于是 150 秒后芯片完成前处理工序便可被推送进入转盘. 由此可知, 上一芯片入仓至少 30s 后, 才能继续发出下一芯片, 即:

$$T_0^{p+1} - T_0^p \geq 30(p = 1, \dots, N - 1) \quad (9)$$

(2)考虑第一次温育过程. 前期处理完成后, 转盘转动对位, 芯片被推入转盘上一个空着的卡槽, 进行第一次温育过程. 过程结束后, 转盘再次转动对位, 芯片进入下一流程. 值得注意的是, 在第一次温育过程至加样过程中间的转动对位时间内, 芯片仍然位于转盘内, 占据 1 个卡槽. 因此, 芯片在第一次温育过程起始时间进入转盘, 直至温育结束且转动对位完成 (加样开始) 才离开转盘. 这与前处理过程是有区别的. 在前处理过程结束后的转动对位过程中, 根据假设, 芯片位于前处理工作台中一处大容量“待机”区域. 转盘上均匀分布 40 个卡槽, 那么, 必须满足当第 P 枚芯片结束第一次温育过程和转动对位而离开转盘时, 第 $p + 40$ 枚芯片尚未进入或正好进入转盘, 即:

$$T_1^{p+40} \geq T_1^p + 60a + 8(p = 1, \dots, N - 40) \quad (10)$$

当然, 这一条件是必要而不充分的, 因为两次温育过程共用同一温育盘, 这一约束将在下一小节进行讨论.

(3)考虑磁珠加样过程. 第一次温育完成后, 转盘转动对位到磁珠工位, 转动期间芯片仍位于温育盘中的卡槽. 加样完毕后, 芯片进入加样台“待机”区域等待转动对位, 不影响后一个芯片的进入. 免疫分析仪中仅有 1 个磁珠加样盒, 则当前一个芯片加样完毕后, 后一个芯片才能进入加样台, 而这一过程需要至少 21 秒时间, 即:

$$T_2^{p+1} - T_2^p \geq 21(p = 1, \dots, N - 1) \quad (11)$$

(4)考虑第二次温育过程. 与第一次温育过程类似, 第二次温育过程结束且转动对位到清洗盒后, 芯片才离开温育盘, 即:

$$T_3^{p+40} \geq T_3^p + 60b + 16(p = 1, \dots, N - 40) \quad (12)$$

这同样是一个必要而不充分条件.

(5)考虑清洗过程. 与加样过程类似, 芯片在清洗开始时进入清洗盒, 清洗结束后进入清洗盒“待机”区域等待转动对位, 不影响后一芯片进入. 转盘上方固定有 8 个清洗

盒, 则必须满足当第 p 枚芯片结束清洗后, 第 $p+8$ 枚芯片尚未进入或正好进入清洗盒, 即:

$$T_4^{p+8} - T_4^p \geq 325 (p = 1, \dots, N - 8) \quad (13)$$

(6)考虑检测过程. 该过程的特点在于其自身可转动, 且仅有 1 个检测盘. 上一枚芯片进入并完成检测后, 检测盘才可开始向下一清洗盒转动, 且转动期间检测盘一定是空置的, 芯片无法进入, 这是检测过程与加样、清洗过程的区别所在. 即:

$$T_5^{p+1} - T_5^p \geq 25 + 12 (p = 1, \dots, N - 1) \quad (14)$$

3.1.6 多重入空间约束条件

多重入 (re-entrant) 制造系统是指零件多次进入同一个加工中心等设备, 这是因为产品加工工艺的需要以及设备数量、经济因素等的约束, 使得在制品必须多次、重复的访问某些设备^[2]. 在本问题中, 由于工艺需要, 免疫分析仪中的芯片需要两次进入温育盘, 正是一类多重入问题.

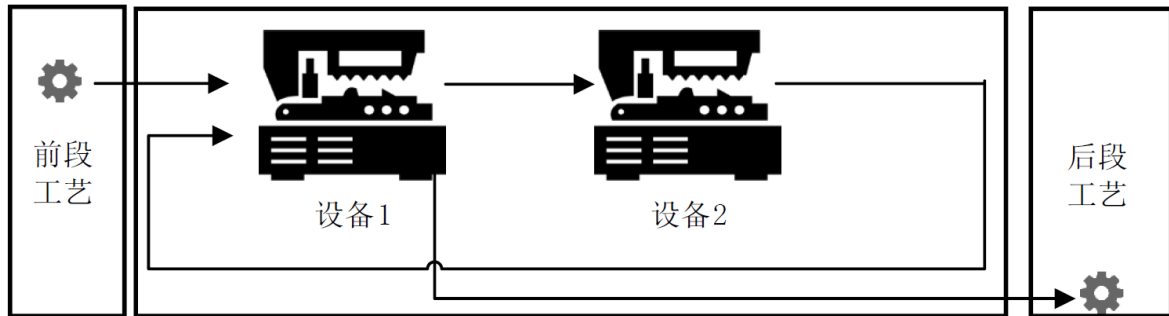


图2 多重入制造系统示意图^[2]

考虑入仓的第 1 枚芯片. 为了使检测流程顺利进行, 必须保证当第 1 枚芯片进入第二次温育过程时, 温育盘仍有空槽, 即以下两种可能性:

- (1)第 41 枚芯片尚未或刚好进入第一次温育流程;
- (2)第 2 枚芯片恰好位于加样工位, 且第 42 枚芯片尚未或刚好进入第一次温育流程.

以上两点满足其一, 第 1 枚芯片即可顺利进入第二次温育过程. 将第 1 枚芯片向后推广到第 p 枚, 即可得到多重入空间约束的充要条件:

$$T_1^{p+40} \geq T_3^p (p = 1, \dots, N - 40) \quad (15)$$

或

$$T_1^{p+41} \geq T_3^p, \quad T_2^{p+1} \leq T_3^p \leq T_2^{p+1} + 25 + 8 \quad (p = 1, \dots, N - 41) \quad (16)$$

联立公式(1)-(16), 就得到了单类芯片生产力线性规划方程组.

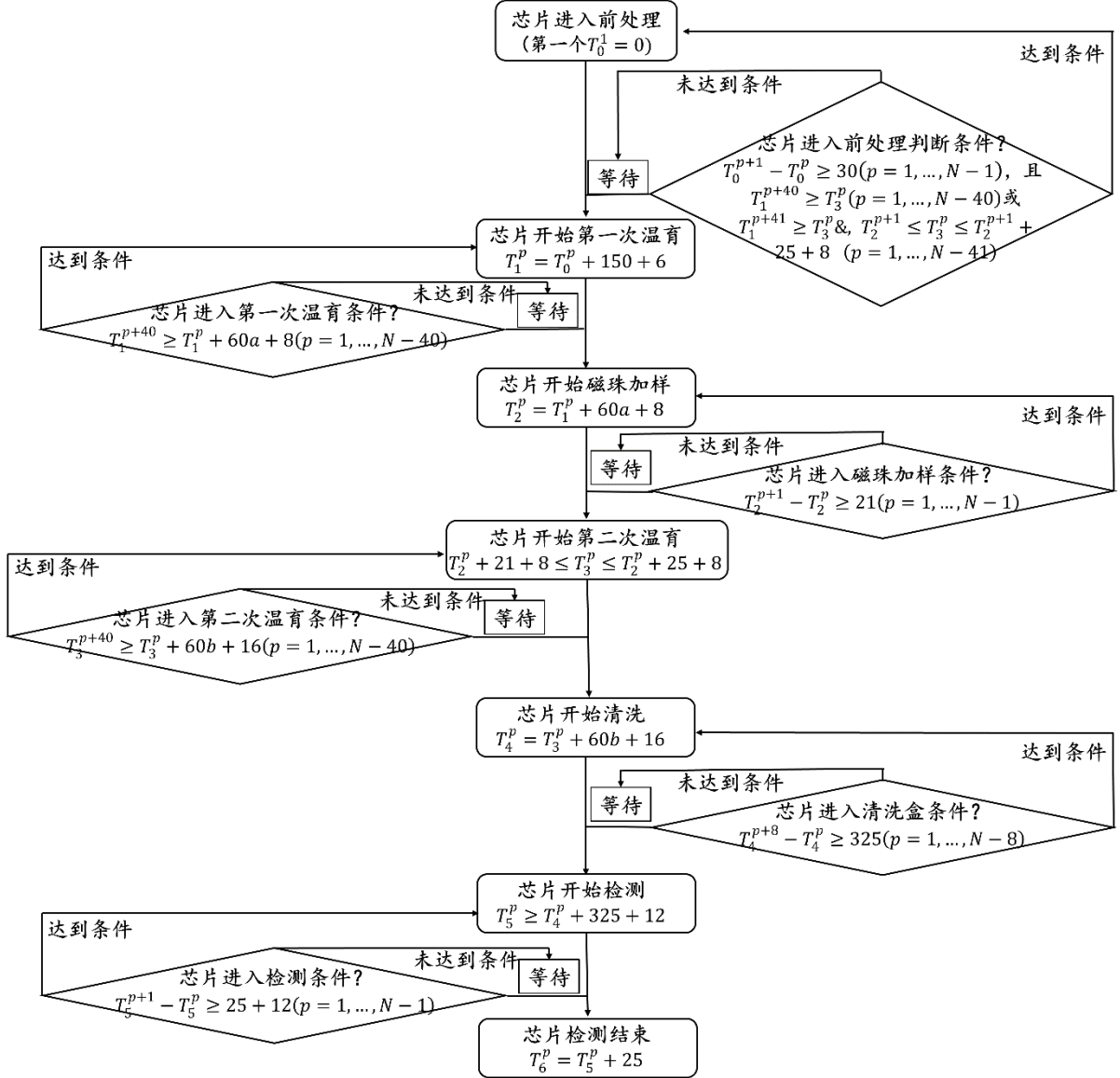


图3 问题 1 处理的具体流程

3.2 模型二: FFSP 的遗传算法优化

处理多种芯片进入工作区求其最短时间实际上是对所需检测芯片排序的问题. 而不同种类的芯片仅第一次温育时间 a 、第二次温育时间 b 不同. 不同种类的第一、二次温育时间分别用 a_z 、 b_z 来表示, 其中 $z = 1, 2, \dots, n$. 在这里, 我们应用柔性 Flow-Shop 调度的优化遗传算法^[3] 进行序列的选择. 将遗传算法应用于柔性 Flow-Shop 调度问题(Flexible Flow-Shop Scheduling Problem, FFSP)中的关键是采用有效的编码和解码方式以及适当的交叉、变异操作. 遗传算法对种群重复地进行选择、交叉、变异等基本遗传操作, 不断产生出比父代更适应环境的新一代种群, 直到满足要求条件为止. 其具体步骤如下:

3.2.1 个体编码

我们采用改进后的编码矩阵, 即不需要再在染色体编码中加分隔符, 改进后的编码矩阵意义更为直观, 从而使染色体编码的长度相应缩短, 在染色体交叉、变异操作运算过程中直接对整条染色体进行, 而不需要专门分段进行操作. 这里, FFSP 的编码方法同

样巧妙地利用了矩阵的元素和位置信息表示 FFSP 的工序之间的约束关系,使得产生的每个染色体对应一个可行的调度而且在进行遗传操作时也不会产生非法解。

这里为使得模型更具有普适性,我们考虑一般的情况.假设共需要检测 N 个芯片(芯片的种类 A 、 B 、 C 的比例完全随机),每个芯片都要依次经过前处理(5个)、第一次温育、磁珠加样、第二次温育、清洗、检测共10个工序,每个工序的并行工位数为 $M_i(i=1, \dots, 10)$.其中,每个工序允许并行的工位已知.首先构造一个 $10 \times N$ 维的 FFSP 的编码矩阵:

$$A_{10 \times N} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{10,1} & \cdots & a_{10,N} \end{bmatrix} \quad (17)$$

其中,编码矩阵第一行的元素 $a_{1,j}$ 为区间 $(1, M_1 + 1)$ 上的一个实数,表示芯片 j 的第一个工序 $Int(a_{1,j})$ 个并行工位上加工,函数 $Int(x)$ 表示对实数 x 取整;编码矩阵第 $m(1 < m \leq 10)$ 行的元素 $a_{m,j}$ 为区间 $(1 + \sum_{i=1}^{m-1} M_i, 1 + \sum_{i=1}^m M_i)$ 上的一个实数,表示芯片 j 的第 m 个工序 $Int(a_{m,j})$ 个并行工位上加工.显然,可能会出现 $Int(a_{i,j}) = Int(a_{i,k}), j \neq k$,这表明多个芯片在进行同一个工序.这时,对于前五个工序($1 \leq i \leq 5$),由于前处理整个过程每个工序仅允许一枚芯片进入,且芯片一旦开始检测则不能停止,因此按照 $a_{1,j}$ 的升序来进行检测.假如不是前5道工序($i > 5$),则根据每个工件的前一个工序的完成时间来确定其加工顺序,前一个工序先完成的先加工.假如完成时间相同,则也按照 a_{ij} 的升序来加工.

根据上述编码矩阵(17)可以确定染色体.染色体由10个小段组成,每个小段包括 N 个基因.即由编码矩阵的每一行组成个小段,表示不同的工序.因此染色体的长度为 $10 \times N$.染色体可表示为:

$$Chrom_i = [a_{11}, a_{12}, \dots, a_{1N}, a_{21}, a_{22}, \dots, a_{2N}, \dots, a_{10,1}, a_{10,2}, \dots, a_{10,N}] \quad (18)$$

3.2.2 个体解码

在这个问题中,我们结合各个工序的顺序以及工位(机器)的数量将各工序与机器对应,得到如下对应情况:

表4 各工序对应的机器情况

工序	1	2	3	4	5	6	7	8	9	10	总机器数
机器数	1	1	1	1	1	s	1	t	8	1	55

为了便于问题求解,下面对温育盘进行划分:由于两次温育共用一个温育盘,即工序6和8中共用一组并行机器,数量为 $s + t = 40$.因此为了降低该重入特性引起的问题复杂度,根据待检测芯片两次温育总时长的占比分配机器即空槽数量,即:

$$s = [(NA * a(A) + NB * a(B) + NC * a(C)) /$$

$$(NA * a(A) + NB * a(B) + NC * a(C) + NA * b(A) + NB * b(B) + NC * b(C))] \quad (19)$$

$$t = 40 - s \quad (20)$$

在利用计算机编程具体进行解码时,算法步骤如下:

(1) 根据表1生成芯片在各个机器上的加工时间矩阵 T .其中,矩阵的列分别表示芯片,行分别表示机器.

(2) 根据得到的各芯片处理的工序与机器的对应关系,生成工序选取的机器矩阵 S .其中,元素为“1”表示相应工序选取该机器,元素为“0”表示相应工序未选取该机器.

(3) 将加工时间矩阵 T 和机器矩阵 S 点乘得到矩阵 TT . 矩阵表示了相应芯片的工序选择的机器及其在该机器上的加工时间.

(4) 根据编码矩阵 A 的含义和矩阵 TT 生成相应工序在所选机器上的完工时间矩阵 ZT . 由于矩阵中元素不为 0 的数值表示的是相应工件每个工序在所选机器上的最早完成时间, 所以矩阵中元素最大的数即为该条染色体所表示的柔性 Flow-Shop 调度问题的最大完成时间.

(5) 由上一步骤生成的 $55 \times N$ 的完工时间矩阵 ZT 如下, 由于本文中工序间的等待时间存在特殊约束, 因此, 我们还要对 ZT 矩阵进行修正, 从而调整 N 枚芯片的初始序列和进入时间, 使得满足芯片的检测流程一旦开始便不能停止, 直到检测完成的基本假设.

$$ZT_{55 \times N} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{55,1} & \cdots & a_{55,N} \end{bmatrix} \quad (21)$$

其中非零项 a_{ij} 表示芯片 j 在完成第 i 道工序时的时刻, i 为 a_{ij} 在所在列中非零元的序数. 由于温育和清洗工序存在大量并行工位, 因此可以想到该矩阵是一个稀疏矩阵. 首先将该矩阵进行去零压缩, 压缩的方法是: 对 ZT 矩阵的 N 个列向量分别剔除零元素, 将向量维数由 55 降到 10, 即从工位总数降到工序总数, 再重新由这 N 个列向量构成一个 $10 \times N$ 的新矩阵, 命名为 GT , 如下:

$$GT_{10 \times N} = \begin{bmatrix} b_{1,1} & \cdots & b_{1,N} \\ \vdots & \ddots & \vdots \\ b_{10,1} & \cdots & b_{10,N} \end{bmatrix} \quad (22)$$

其中 b_{ij} 表示芯片 j 在完成第 i 道工序时的时刻, 显然 GT 矩阵只含有非零元.

对 GT 矩阵 $[b_1 \ b_2 \ \dots \ b_{10}]^T$ 的 10 个行向量由底部进行回溯相减, 即令 $c_{ij} = b_{ij} - b_{i-1,j}$, $c_1 = b_1$, 得到一个 $10 \times N$ 的新矩阵, 命名为 WT 矩阵, 如下:

$$WT_{10 \times N} = \begin{bmatrix} c_{1,1} & \cdots & c_{1,N} \\ \vdots & \ddots & \vdots \\ c_{10,1} & \cdots & c_{10,N} \end{bmatrix} \quad (23)$$

构造一个 10×1 的工时向量 $u[j] = [30 \ 30 \ 30 \ 30 \ 30 \ 30 \ a(j) \ m \ b(j) \ n \ 25]^T$, 该向量给出了 j 芯片在各工序中的时间, 其中 $m \in [21, 25]$, $n \geq 325$, $a(j)$ 为第一次温育时间, $b(j)$ 为第二次温育时间, $1 \leq j \leq N$. 由 N 个工时向量构成一个 $10 \times N$ 的新矩阵 U , 如下:

$$U_{10 \times N} = [u[1], u[2] \ \dots \ u[N]] \quad (24)$$

由 $WT-U$ 即可得到非法等待矩阵 DT , 如下.

$$DT_{10 \times N} = \begin{bmatrix} d_{1,1} & \cdots & d_{1,N} \\ \vdots & \ddots & \vdots \\ d_{10,1} & \cdots & d_{10,N} \end{bmatrix} \quad (25)$$

值得注意的是: 1) 对于加样工序, 芯片在加样盒中的时间至少为 25s 时才开始发生非法等待; 2) 对于清洗工序, 芯片不存在非法等待. 即:

1) 考虑加样工序, $d_{7j} = c_{ij} - \max\{0, c_{ij} - 25\}$;

2) 考虑清洗工序, $d_{9j} = 0$;

由于芯片超出等待时间约束的序列是不被允许的, 因此有必要根据 DT 矩阵对芯片的入仓时间进行调整, 以期消除上述的非法等待. 入仓时间调整后, 输入芯片的顺序也相应发生变化, 具体方法如下:

$$X = \{T_0^1 + \sum_{1 \leq i \leq 10} d_{i1}, T_0^2 + \sum_{1 \leq i \leq 10} d_{i2}, \dots, T_0^N + \sum_{1 \leq i \leq 10} d_{iN}\} \quad (26)$$

$$Y = X - \min X \quad (27)$$

最后对Y中元素进行升序排序,即可得到以消除非法等待时间为目的芯片入仓序列和时间,其中第一个元素为零,即第一枚入仓时间为零.

由于柔性 Flow-Shop 调度问题是求解最小化最大完成时间,因此将所有染色体按上述步骤解码后所求得数值的最小数值为该次迭代的最小化最大完成时间,染色体通过不断选择、交叉和变异操作,最终求得最优调度.

3.2.3 群体规模选择

合适的群体规模对遗传算法的收敛具有重要意义. 群体太小难以求得满意的结果,群体太大则计算复杂.

3.2.4 适值函数

遗传算法进行选择操作时会出现欺骗问题:在遗传进化的初期,通常会产生一些超常的个体,若按照比例选择法,这些异常个体因竞争力太突出而控制了选择过程,影响算法的全局优化性能;在遗传进化的后期,即算法接近收敛时,由于种群中个体适应度差异较小时,继续优化的潜能降低,可能获得某个局部最优解. 适值函数设计不当有可能造成这种问题的出现.

由于优化目标为最小化最大完成时间,因此令目标函数作指数变换得到适值函数:

$$f = \alpha \exp(-\beta * C_i) \quad (28)$$

其中, α, β 为正实数,在本文中均选取 1; C_i 为最大完成时间.

3.2.5 选择

选择是用来确定重组或交叉个体,以及备选个体将产生多少个子代个体. 选择的第一步是计算适值,采用按比例适应度分配,是利用比例于各个个体适应度的概率决定其子孙的遗留可能性. 若有 M 个个体,其中某个个体 i ,其适值为 f ,则其被选择的概率表示为:

$$P_i = \frac{f_i}{\sum_{k=1}^M f_k} \quad (29)$$

然后对各个染色体计算它们的累积概率,如第 k 个个体的累积概率为:

$$q_k = \sum_{i=1}^M p_i \quad (30)$$

第二步用轮盘赌选择法进行选择. 为了选择交配个体,需要进行多轮选择,每一轮产生一个 $[0,1]$ 均匀随机数,将该随机数作为选择指针来确定备选个体.

3.2.6 交叉与变异

交叉在遗传操作中起核心作用,交叉概率较大可增强遗传算法开辟新搜索空间的能力,但性能好的基因串遭到破坏的可能性较大,算法收敛速度降低且不稳定;若交叉概率较小,则遗传算法搜索可能陷入迟钝状态. 由于这里采用实值编码,为保证交叉后产生新的参数值,并开辟出新的搜索空间,这里的交叉操作采用离散重组方式四,即在个体间交换变量值,对每个变量,贡献给子代变量值的父代是随机的以相同的概率挑选的.

变异在遗传操作中属于辅助性的搜索操作，主要目的是维持群体的多样性。这里采用偏置变异，以一定的概率给变异位基因加一个从偏置区域中随机选取的数值，并保证变异后的基因在自己的取值范围内。

以上具体流程如下图所示：

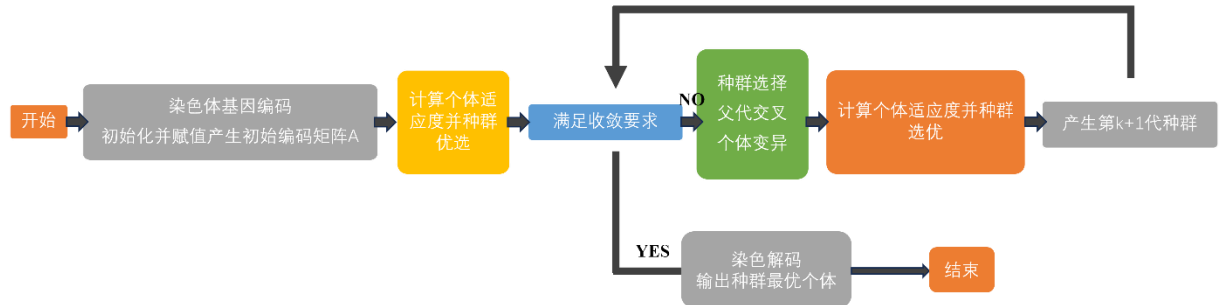


图4 解决 FFSP 问题的遗传算法模型流程图

3.3 模型三：多类型线性规划模型

对于多类型芯片的调度问题，除模型二所述的 FFSP 的遗传算法优化外，也可以从线性规划的角度进行建模求解。

3.3.1 问题分析

问题二至三基于问题一的模型基本假设，芯片由单一类型转变为多种类型，每种类型的芯片有已知若干，求解芯片的最优检测序列和对应的最小化最大流程时间。

3.3.2 模型指标定义

设芯片总数为 N 。各类芯片的数量分别为 N_a 、 N_b 、 N_c ($N_a+N_b+N_c=N$)，如表 5 所示。芯片进入各工区的时间变量仍参照表 2。

表5 各类芯片数量 (单位: 个)

芯片类型	第一次温育时间	第二次温育时间
A	10	5
B	40	20
C	25	10

3.3.3 目标函数

当最后一枚芯片完成检测弹出时，检测流程结束。由于各类芯片的温育时间存在差异，对从入仓到弹出的整个检测流程来说，芯片并非遵循先进先出原则，因此根据指标定义，目标函数：

$$\min = \max_{1 \leq p \leq N} \{T_6^p\} \quad (31)$$

即，最后一枚弹出的芯片的弹出时间最早。

3.3.4 多类芯片时间约束条件

与单类芯片类似地，有：

$$T_0^1 = 0 \quad (32)$$

$$T_1^p = T_0^p + 150 + 6 \quad (33)$$

$$T_2^p = T_1^p + 60a(p) + 8 \quad (34)$$

$$T_2^p + 21 + 8 \leq T_3^p \leq T_2^p + 25 + 8 \quad (35)$$

$$T_4^p = T_3^p + 60b(p) + 16 \quad (36)$$

$$T_5^p \geq T_4^p + 325 + 12 \quad (37)$$

$$T_6^p = T_5^p + 25 \quad (38)$$

3.3.5 多类芯片基本空间约束条件

本小节仍将六个工序分开，分别讨论同一工序中，由于工区容量而导致的芯片进入制约。值得注意的是，不同类型芯片的温育时间存在差异，芯片共经历两次温育，每次温育都是一次洗牌，温育结束芯片序列受到更改，共发生两次更改，这将对温育后的空间约束条件表达产生影响。

(1)考虑前处理过程. 与问题一相同地，芯片入仓间隔至少为 30s，即：

$$T_0^{p+1} - T_0^p \geq 30(p = 1, \dots, N - 1) \quad (39)$$

(2)考虑第一次温育过程. 当第 p 枚芯片结束第一次温育过程和转动对位而离开转盘时，第 $p+40$ 枚芯片尚未进入或正好进入转盘，即：

$$T_1^{p+40} \geq T_1^p + 60a(p) + 8(p = 1, \dots, N - 40) \quad (40)$$

由于两次温育过程共用同一温育盘，这一条件是必要而不充分的，由可重入特性引起的空间约束将在下一小节进行讨论。

(3)考虑磁珠加样过程. 在问题一的基础上，芯片序列在结束第一次温育后发生改变，而在新序列中任意相邻两枚芯片开始加样的时间间隔至少为 21 秒，即原序列中任意两枚芯片开始加样的时间间隔至少为 21 秒，即：

$$|T_2^p - T_2^i| \geq 21(p = 2, \dots, N; 1 \leq i \leq p - 1) \quad (41)$$

(4)考虑第二次温育过程.

与第一次温育过类似地，当第 p 枚芯片离开温育盘，即结束第二次温育过程且转动对位到清洗盒后，第 $p+40$ 枚芯片才从加样区重回温育盘，即：

$$T_3^{p+40} \geq T_3^p + 60b(p) + 16(p = 1, \dots, N - 40) \quad (42)$$

这同样是一个必要而不充分条件。

(5)考虑清洗过程. 与温床类似地，第 p 个芯片开始清洗时，清洗盒中最多有 7 枚芯片正在清洗，即至少 $p-1-7$ 枚不位于清洗盒。分析这 $p-8$ 枚芯片的位置，存在两种情况：1) 已结束清洗；2) 还未进入清洗。即：

$$\begin{aligned} T_4^p - T_4^i &\geq 325(p = 9, \dots, N; 1 \leq i \leq p - 1) \\ &\text{or} \\ T_4^p &< T_4^i(p = 9, \dots, N; 1 \leq i \leq p - 1) \end{aligned} \quad (43)$$

对任意的 p ，存在至少 $p - 8$ 个解 i 。

(6)考虑检测过程. 与磁珠加样过程类似地，两次温育后，新序列中任意相邻两枚芯片开始检测的时间，即原序列中任意两枚芯片开始检测的时间至少为检测所需时长与检测盘对位时间之和，即：

$$T_5^p - T_5^i \geq 25 + 12(p = 2, \dots, N; 1 \leq i \leq p - 1) \quad (44)$$

3.3.6 多重入空间约束条件

(1)考虑第一次温育过程. 当第 p 枚芯片刚好要进入第一次温育的时候, 温床上至少有 1 个空槽, 即温床至多正承载了 39 枚芯片. 换言之, 原序列中先于第 p 枚芯片入仓, 同时也是先于第 p 枚芯片进入第一次温育的 $(p-1)$ 枚芯片中至少有 $p-1-39$ 枚芯片不位于温床. 分析这 $p-40$ 枚芯片的位置, 有且仅存在两种情况: 1) 芯片已结束第二次温育并离开温床; 2) 芯片正在加样且还未回到温床. 即:

$$\begin{aligned} T_1^p &\geq T_3^i + b(i) + 16(p = 41, \dots, N; 1 \leq i \leq p-1) \\ &\quad \text{or} \\ T_2^i &\leq T_1^p \leq T_2^i + 25 + 8(p = 41, \dots, N; 1 \leq i \leq p-1) \end{aligned} \quad (45)$$

对任意的 p , 存在至少 $p-40$ 个解 i .

(2)考虑第二次温育过程. 与第一次温育过程类似地, 当第 p 枚芯片刚好要进入第二次温育的时候, 温床上至少有 1 个空槽, 即温床至多正承载了 39 枚芯片. 因此, 至少原序列中先于第 p 枚芯片入仓和开始第一次温育的 $(p-1)$ 枚芯片中至少有 $p-1-39$ 枚芯片不位于温床. 同样分析这 $p-40$ 枚芯片的位置, 仍旧有且仅存在上述的两种情况: 1) 已结束第二次温育并离开温床; 2) 正在加样且还未回到温床. 即:

$$\begin{aligned} T_3^p &\geq T_3^i + b(i) + 16(p = 41, \dots, N; 1 \leq i \leq p-1) \\ &\quad \text{or} \\ T_2^i &\leq T_3^p \leq T_2^i + 25 + 8(p = 41, \dots, N; 1 \leq i \leq p-1) \end{aligned} \quad (46)$$

对任意的 p , 存在至少 $p-40$ 个解 i .

由于 LINGO 对条件语句的限制, 清洗工序的基本空间约束条件和基于温床的多重入空间约束条件的写入存在困难, 本文对多类芯片生产力线性规划模型不进行求解. 并考虑到后续问题欲将芯片种类推广到 N , 且各类芯片数量未知, 线性规划模型面临数据规模大, 复杂度高的难题, 因此后续问题将改用基于智能搜索的遗传算法进行建模和求解.

3.4 模型四: 检测过程中的方案调整

在前面四个问题的求解中, 待检测芯片的类型、数量均为已知. 本问题中, 运行 30 分钟后新到 C 型芯片 20 片这一情况可考虑为突发情况, 即在 0 时刻, 仅考虑现有 A 型芯片与 B 型芯片检测时间最短的优化方案已经开始执行; 30 分钟后, 对尚未入仓的 A、B 型芯片及新到的 C 型芯片重新规划最优方案.

因此, 可按如下思路求解问题五:

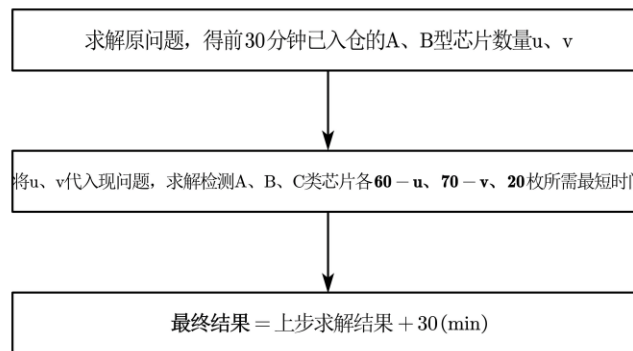


图5 检测过程中有方案调整的求解流程

注: 对于各阶段的求解仍采用模型二.

4 模型求解

4.1 参数设置

基于前述建模，使用 python 编程实现对应算法。结合收敛速度、算力与前人工作，经调试，将遗传算法相关各参数设置如下：

表6 遗传算法参数设置

参数	值
初始种群大小	6000
迭代次数	60
交叉概率	0.7
变异概率	0.1

表 7 展示了问题 2、3、5 的最短检测时间求解结果，完整表格附于文后。

表7 问题 2、3、5 的最短检测时间求解结果

问题	芯片 A	芯片 B	芯片 C	最短检测时间(s)
2	80	80	0	18105
3	60	50	50	18205
5	60	70	20(后置入)	18939

4.2 问题一：单类芯片生产力线性规划模型求解

本题使用模型一求解。以式(1)为目标函数，式(2)-(16)为约束条件，使用 Lingo 编程求解单类芯片生产力线性规划模型，并将结果填入附表 1、附表 2。简略结果如表 8 所示。

单位时间（1 小时）内，免疫分析仪能够检测 A 类型芯片 53 枚，所需时间为 59.5 分钟。检测 54 枚 A 类型芯片则需要 60.1 分钟。然而，免疫分析仪无法在 1 小时内完成对 1 枚 B 类型芯片的检测，本题结果为 0 枚。使用免疫分析仪检测 1 枚 B 类型芯片至少需要 69.5 分钟。

表8 各类芯片检测时间（单位：秒）

芯片类型	1 小时内可检测芯片数	具体时长	检测 1 枚芯片所需时间
A	53	3569	1471
B	0	\	4171

如图 6、7，随着芯片增加，检测时间整体上升。A 类芯片相对于 B 类芯片较为平缓。在第 8 枚芯片处，出现了小突变，容易想到这是清洗盒限制的具体体现。第 40 枚芯片处则出现了更明显的突变，且 B 类芯片尤其显著，则是温育盘卡槽数带来的现象，而 B 类芯片温育时间长，因此对空间约束也更加敏感。

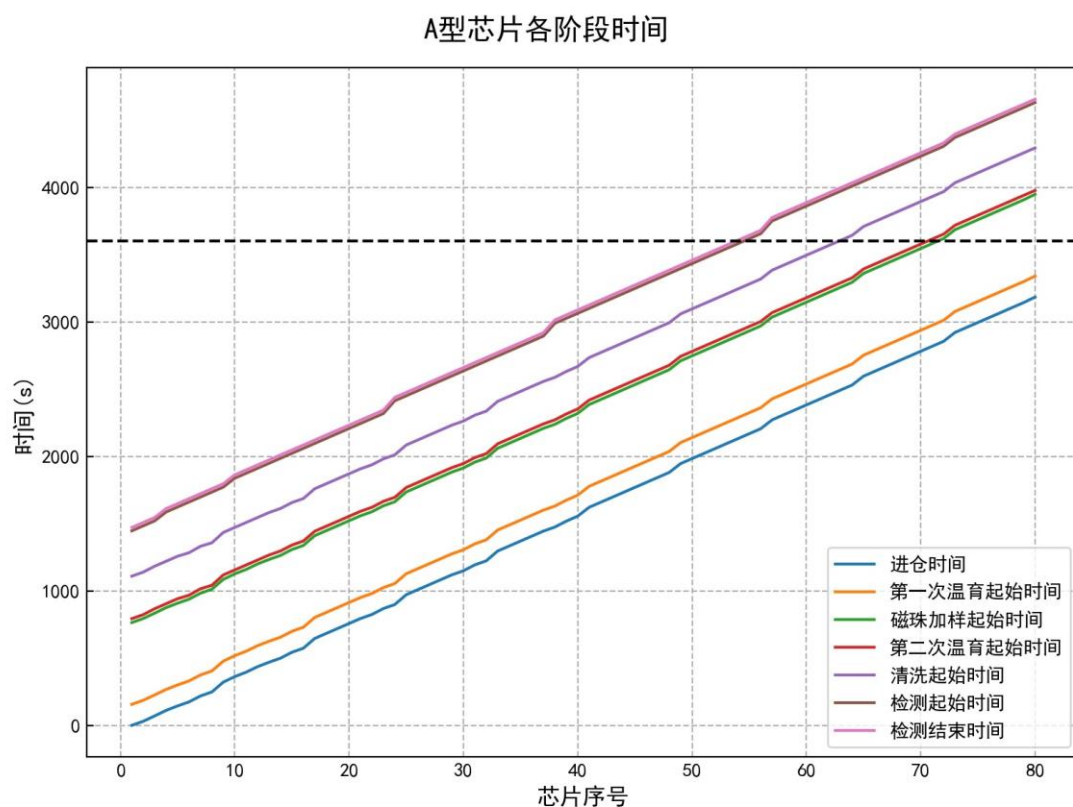


图6 A 型芯片处理各阶段起始时间及检测结束时间，黑色粗虚线表示单位时间 (1 小时)

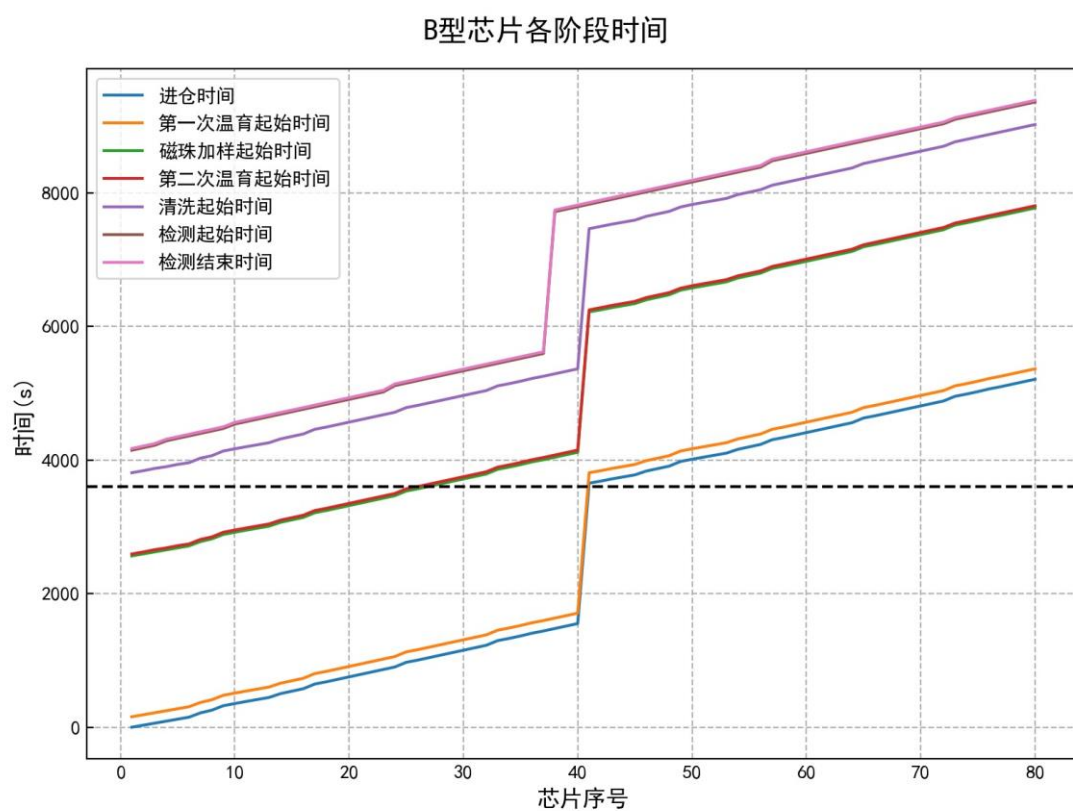


图7 B 型芯片处理各阶段起始时间及检测结束时间，黑色粗虚线表示单位时间 (1 小时)

4.3 问题二: A、B 类型芯片已知条件下的 FFSP 遗传算法模型求解

本题使用模型二求解. 图 8 展示基于问题 2 所给数据求解模型优化过程. 经计算, 温育盘中 27 个芯片槽分配为第一次温育槽, 13 个芯片槽分配为第二次温育槽. 60 次迭代过后, 模型尚未完全收敛, 但最优目标函数值已趋于稳定. 求得的最短检测时间为 18105 秒.

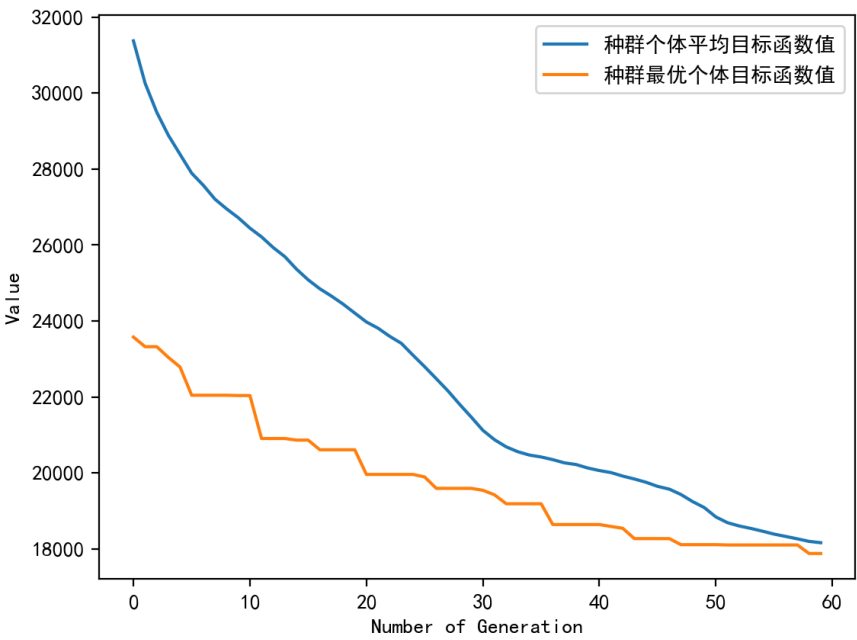


图8 问题 2 优化过程图像

如下图, 芯片序列末端大多数为温育时间较短的 A 型芯片, 而温育时间较长的 B 型芯片在 A 芯片前检测, 符合一般认知.

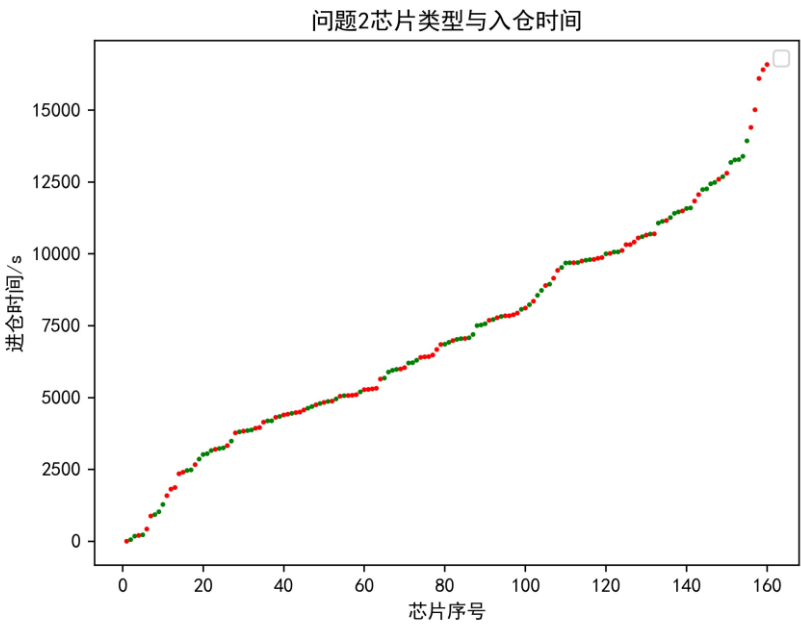


图9 芯片类型与入仓时间, 其中颜色表示芯片类型不同, 红色为 A 芯片, 绿色为 B 芯片.

4.4 问题三: A、B、C 类型芯片已知条件下的 FFSP 遗传算法模型求解

本题使用模型二求解. 图 10 展示了问题 3 所给数据求解模型优化过程. 经计算, 温育盘中 27 个芯片槽分配为第一次温育槽, 13 个芯片槽分配为第二次温育槽. 最优目标函数值在第 33 次迭代已趋于收敛, 平均目标函数则在 40 次迭代后逐渐收敛. 求得最短检测时间为 18205 秒.

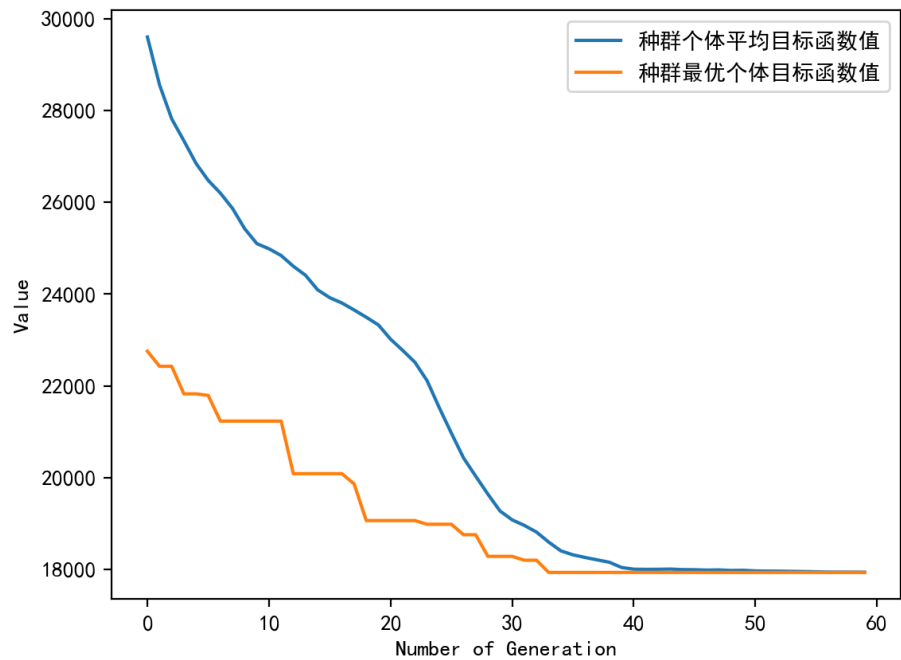


图10 问题 3 优化过程图像

如下图, 芯片序列末端大多数为温育时间较短的 A、C 型芯片, 而温育时间较长的 B 型芯片相对靠前检测. 进仓时间或出现一些断点, 体现了对芯片加工的连续性要求.

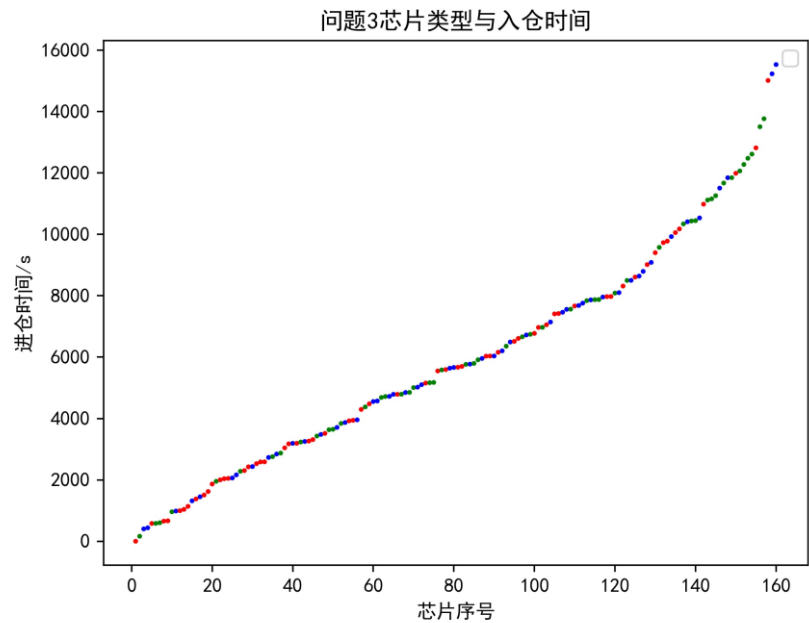


图11 芯片类型与入仓时间, 其中颜色表示芯片类型不同, 红色为 A 芯片, 绿色为 B 芯片, 蓝色为 C 芯片.

4.5 问题四: 未知类型芯片条件下的 FFSP 遗传算法模型求解

本题使用模型二求解. 将模型推广到一般情形. 模型二求解过程与芯片数量、类型、温育时间无直接联系, 因此, 可直接修改代码中的参数设置, 再行求解即可. 引入芯片 D, 修改、自定义芯片参数如下:

表9 自定义多类芯片参数表 (单位: 分钟)

芯片类型	芯片数量	第一次温育时间	第二次温育时间
A	30	5	10
B	40	20	40
C	35	10	25
D	10	50	15

算法中参数设置与问题 2、3 相同. 经计算, 温育盘中 14 个芯片槽分配为第一次温育槽, 26 个芯片槽分配为第二次温育槽. 求解模型, 得图 12. 最优目标函数值和平均目标函数值分别在第 30、38 次迭代后逐渐收敛, 最终得到的最短检测时间为 15972 秒.

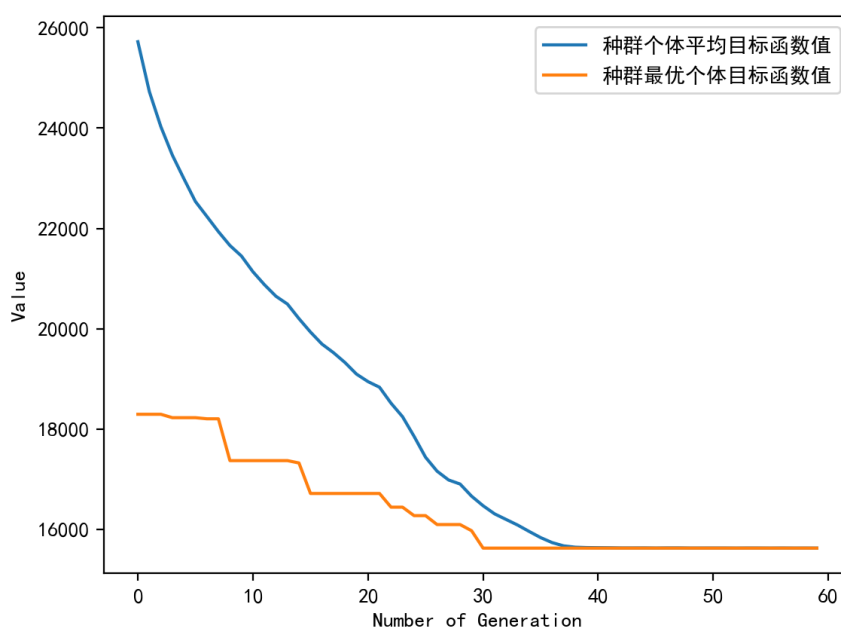


图12 问题 4-自定义芯片组优化过程图像

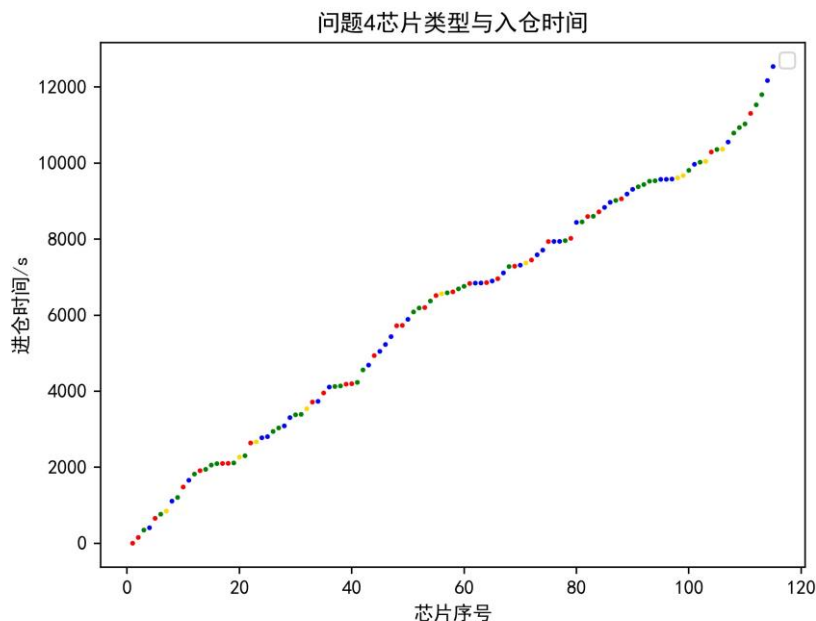


图13 芯片类型与入仓时间，其中颜色表示芯片类型不同，红色为 A 芯片，绿色为 B 芯片，蓝色为 C 芯片，黄色为 D 芯片。

4.6 问题五：中途插入检测条件下的 FFSP 遗传算法模型求解

4.6.1 原问题求解

本题使用模型四求解. 参数设置参考表 6. 要求前 30 分钟内进入免疫分析仪的芯片数量，即满足：

$$T_0^p \leq 1800 \quad (30)$$

经计算，温育盘中 27 个芯片槽分配为第一次温育槽，13 个芯片槽分配为第二次温育槽. 最终得到 $u = 10, v = 6$ ，其中 u 和 v 分别为前 30 分钟已经进入检测的 A, B 芯片数量. 其进仓时间如下表 (完整表格附于支撑材料)：

表10 原问题中前 30 分钟进入检测仪的芯片

芯片序号	芯片类型	进仓时间
1	B	0
2	A	137
3	A	382
4	A	483
5	B	486
6	B	596
7	B	660
8	A	807
9	B	983
10	A	1017

11	A	1286
12	A	1294
13	A	1393
14	A	1518
15	B	1729
16	A	1731
17	B	1769

如图所示，模型在约 60 次迭代中几乎恰好收敛，求得最短检测时间为 15728s.

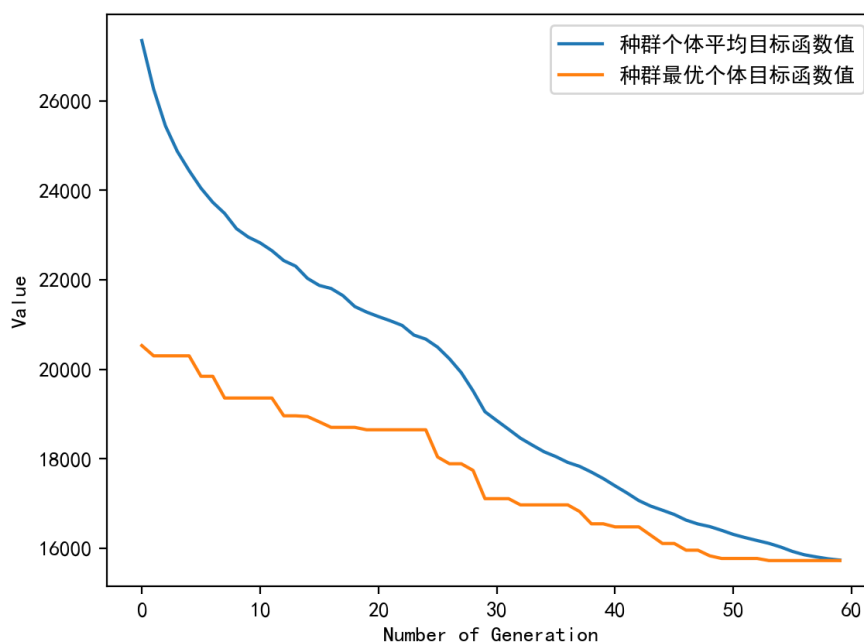


图14 问题 5-原问题优化过程图像

4.6.2 现问题求解

除去上一小节中求出的在 30 分钟内处理完毕的 10 枚 A 类芯片、6 枚 B 类芯片，本小节求解剩余 50 枚 A 类芯片、64 枚 B 类芯片以及新到 20 枚 C 类芯片的最小检测时间，参数设置不变。经计算，温育盘中 27 个芯片槽分配为第一次温育槽，13 个芯片槽分配为第二次温育槽。如图 15 所示，目标函数最优值在第 47 次迭代中收敛，平均值则在第 53 次迭代中收敛，结果为 17139s. 完整表格附于支撑材料中。

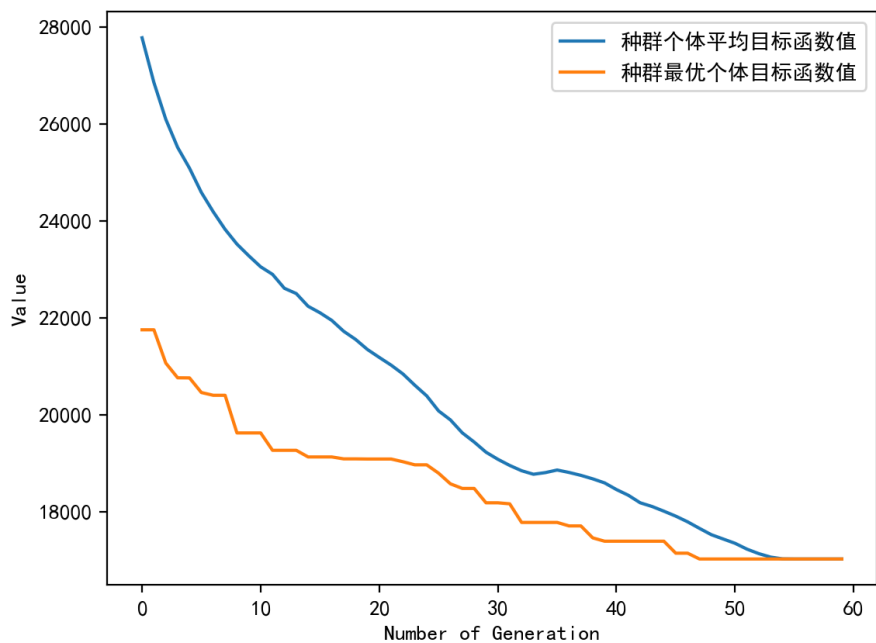


图15 问题 5-现问题优化过程图像

4.6.3 合并求最终解

上一小节中对现问题的求解仍规定七点为 0 时刻，然而实际上该阶段在整个问题中自第 30 分钟开始，因此，将 C 芯片加入后的求解结果向后顺延 30 分钟，并与表 合并，即可得到最终结果，已附于文后。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 18939 秒。进仓时间与芯片顺序示意图如下。

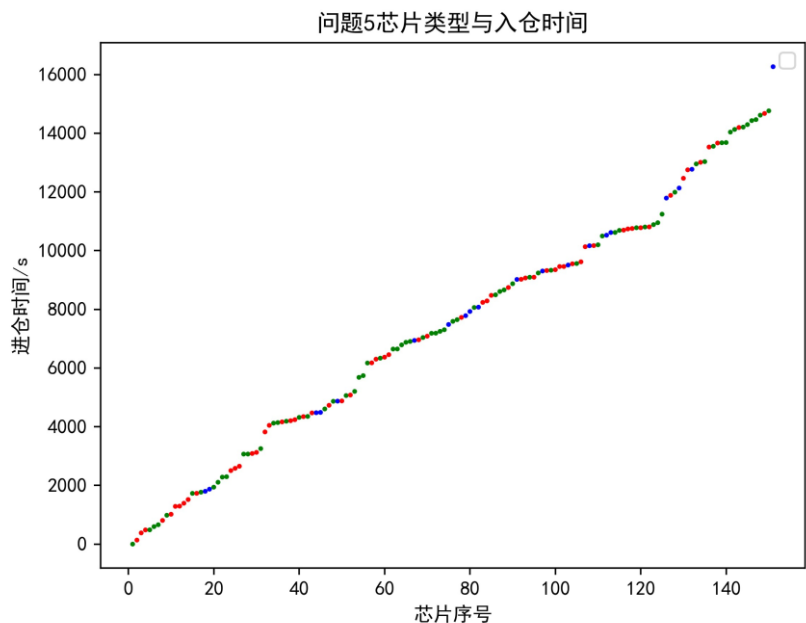


图16 芯片类型与入仓时间，其中颜色表示芯片类型不同，红色为 A 芯片，绿色为 B 芯片，蓝色为 C 芯片。

5 模型推广、评价与改进

5.1 模型推广

在问题 4 中, 若将我们所设计的模型和算法推广到芯片数量为 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, 并且推广到 N 种类型的芯片, 那么按照我们所设计的规划算法进行运算和利用数据进行编码时, 所需要考虑的条件 (包括时间约束和空间约束) 就会较为复杂. 因此在总体上规划算法的实现随着芯片的数量更多、种类更复杂, 其最终在实现上就会更难. 而我们在适用于柔性 Flow-Shop 调度问题的遗传算法时, 虽然在编码矩阵部分就会比较复杂, 但其利用矩阵的表达形式以及相应的选择标准会使得问题变得简单. 所以遗传算法来计算并不受芯片数量、温育时间和芯片种类的限制, 其适用性更强.

5.2 模型的优点

- (1) 规划模型的建立充分结合了仪器使用的实际, 通过合理地假设, 考虑了诸多重要因素得到合理的模型, 如: 转动对位过程必须在上一工序完成后才能开始, 前处理工作台、磁珠加样盒及清洗盒中均存在一个大容量“待机”区域. 这样得到的模型贴合实际, 具有较高的应用价值, 可以推广到其他同类型的仪器上;
- (2) 模型运用实验条件理想化和抓住主要矛盾的思想, 抓住影响芯片检测调度问题的重要因素, 将复杂问题转化为简单问题, 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) 本文所使用的适用于柔性 Flow-Shop 调度问题的改进遗传算法具有很好的鲁棒性, 并行效率高, 具有实际应用的价值.

5.3 模型的不足

- (1) 在问题 2~4 使用 FFSP 的遗传算法模型中, 所需求解时间较长, 并且模型强制将温育盘分开, 一方面简化了计算, 另一方面也限制了求解的准确性;
- (2) 在问题 5 求解过程中, 求解过程需要人为干预, 且未讨论原问题与现问题连接处的两枚芯片之间是否满足时间约束;
- (3) 由于遗传算法中存在随机选择、交叉和变异等操作, 其结果具有一定的随机性. 多次求解并求平均可以帮助减小这种随机性的影响, 提高结果的可靠性和稳定性. 但由于时间限制, 未能实施;
- (4) 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. 对于其他情形 (如: 复杂多种的芯片), 可能无法达到较好的效果;
- (5) 本文采用的 FFSP 的遗传算法模型中, 为了简化问题, 将温育盘中的卡槽按照所有待处理芯片两次温育的总时长做了硬性划分, 与实际情况有所出入, 影响优化结果.

5.4 模型的改进

在运用适合柔性 Flow-Shop 调度问题的改进遗传算法时, 主要考虑了整个线程上的效率最优、时间最短, 而未曾考虑对于在同一个工序中不同的工作槽位的利用率问题. 结合参考文献^{[4] [5]}, 进一步考虑分析仪同一工序不同工作槽位的影响, 从而得到更合理的模型, 进而得到解决此类问题更优化的解.

参考文献

- [1] 张建鹏. 全自动化学发光免疫分析仪[D].天津大学,2006.
- [2] 陈齐圳. 多重入制造系统动态作业调度仿真与优化[D].武汉理工大学,2018.
- [3] 周辉仁, 唐万生, 魏颖辉.柔性 Flow—Shop 调度的遗传算法优化[J], 计算机工程与应用, 2009, 45(30): 224-226, 233.
- [4] 王万良, 姚明海, 吴云高, 等. 基于遗传算法的混合 Flow-Shop 调度方法[J], 系统仿真学报, 2002, 14(7): 863-865, 869.
- [5] 于兆勤, 苏平. 基于遗传算法和仿真分析的混合装配线平衡问题研究[J]. 计算机集成制造系统, 2008, 14 (6) : 1120-1129.

附 录

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	30	186	794	823	1139	1483	1508
3	70	226	834	867	1183	1520	1545
4	111	267	875	904	1220	1586	1611
5	144	300	908	941	1257	1623	1648
6	174	330	938	967	1283	1660	1685
7	218	374	982	1015	1331	1697	1722
8	248	404	1012	1041	1357	1734	1759
9	321	477	1085	1118	1434	1771	1796
10	362	518	1126	1155	1471	1837	1862
11	395	551	1159	1192	1508	1874	1899
12	436	592	1200	1229	1545	1911	1936
13	469	625	1233	1266	1582	1948	1973
14	499	655	1263	1296	1612	1985	2010
15	543	699	1307	1340	1656	2022	2047
16	573	729	1337	1370	1686	2059	2084
17	646	802	1410	1443	1759	2096	2121
18	683	839	1447	1480	1796	2133	2158
19	720	876	1484	1517	1833	2170	2195
20	757	913	1521	1554	1870	2207	2232
21	794	950	1558	1591	1907	2244	2269
22	824	980	1588	1621	1937	2281	2306
23	868	1024	1632	1665	1981	2318	2343
24	898	1054	1662	1695	2011	2413	2438
25	971	1127	1735	1768	2084	2450	2475
26	1008	1164	1772	1805	2121	2487	2512
27	1045	1201	1809	1842	2158	2524	2549
28	1082	1238	1846	1879	2195	2561	2586
29	1119	1275	1883	1916	2232	2598	2623
30	1149	1305	1913	1946	2262	2635	2660
31	1193	1349	1957	1990	2306	2672	2697
32	1223	1379	1987	2020	2336	2709	2734
33	1296	1452	2060	2093	2409	2746	2771
34	1333	1489	2097	2130	2446	2783	2808
35	1370	1526	2134	2167	2483	2820	2845

36	1407	1563	2171	2204	2520	2857	2882
37	1444	1600	2208	2241	2557	2894	2919
38	1474	1630	2238	2271	2587	2989	3014
39	1518	1674	2282	2315	2631	3026	3051
40	1555	1711	2319	2352	2668	3063	3088
41	1621	1777	2385	2418	2734	3100	3125
42	1658	1814	2422	2455	2771	3137	3162
43	1695	1851	2459	2492	2808	3174	3199
44	1732	1888	2496	2529	2845	3211	3236
45	1769	1925	2533	2566	2882	3248	3273
46	1806	1962	2570	2603	2919	3285	3310
47	1843	1999	2607	2640	2956	3322	3347
48	1880	2036	2644	2677	2993	3359	3384
49	1946	2102	2710	2743	3059	3396	3421
50	1983	2139	2747	2780	3096	3433	3458
51	2020	2176	2784	2817	3133	3470	3495
52	2057	2213	2821	2854	3170	3507	3532
53	2094	2250	2858	2891	3207	3544	3569
54	2131	2287	2895	2928	3244	3581	3606
55	2168	2324	2932	2965	3281	3618	3643
56	2205	2361	2969	3002	3318	3655	3680
57	2271	2427	3035	3068	3384	3750	3775
58	2308	2464	3072	3105	3421	3787	3812
59	2345	2501	3109	3142	3458	3824	3849
60	2382	2538	3146	3179	3495	3861	3886
61	2419	2575	3183	3216	3532	3898	3923
62	2456	2612	3220	3253	3569	3935	3960
63	2493	2649	3257	3290	3606	3972	3997
64	2530	2686	3294	3327	3643	4009	4034
65	2596	2752	3360	3393	3709	4046	4071
66	2633	2789	3397	3430	3746	4083	4108
67	2670	2826	3434	3467	3783	4120	4145
68	2707	2863	3471	3504	3820	4157	4182
69	2744	2900	3508	3541	3857	4194	4219
70	2781	2937	3545	3578	3894	4231	4256
71	2818	2974	3582	3615	3931	4268	4293
72	2855	3011	3619	3652	3968	4305	4330
73	2921	3077	3685	3718	4034	4371	4396
74	2958	3114	3722	3755	4071	4408	4433
75	2995	3151	3759	3792	4108	4445	4470
76	3032	3188	3796	3829	4145	4482	4507
77	3069	3225	3833	3866	4182	4519	4544

78	3106	3262	3870	3903	4219	4556	4581
79	3143	3299	3907	3940	4256	4593	4618
80	3184	3340	3948	3977	4293	4630	4655

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	30	186	2594	2623	3839	4183	4208
3	60	216	2624	2657	3873	4220	4245
4	90	246	2654	2683	3899	4286	4311
5	120	276	2684	2717	3933	4323	4348
6	150	306	2714	2743	3959	4360	4385
7	214	370	2778	2811	4027	4397	4422
8	255	411	2819	2848	4064	4434	4459
9	321	477	2885	2918	4134	4471	4496
10	354	510	2918	2951	4167	4537	4562
11	385	541	2949	2982	4198	4574	4599
12	415	571	2979	3012	4228	4611	4636
13	445	601	3009	3042	4258	4648	4673
14	502	658	3066	3099	4315	4685	4710
15	539	695	3103	3136	4352	4722	4747
16	576	732	3140	3173	4389	4759	4784
17	646	802	3210	3243	4459	4796	4821
18	679	835	3243	3276	4492	4833	4858
19	716	872	3280	3313	4529	4870	4895
20	753	909	3317	3350	4566	4907	4932
21	790	946	3354	3387	4603	4944	4969
22	827	983	3391	3424	4640	4981	5006
23	864	1020	3428	3461	4677	5018	5043
24	901	1057	3465	3498	4714	5113	5138
25	971	1127	3535	3568	4784	5150	5175
26	1004	1160	3568	3601	4817	5187	5212
27	1041	1197	3605	3638	4854	5224	5249
28	1078	1234	3642	3675	4891	5261	5286
29	1115	1271	3679	3712	4928	5298	5323
30	1152	1308	3716	3749	4965	5335	5360
31	1189	1345	3753	3786	5002	5372	5397
32	1226	1382	3790	3823	5039	5409	5434
33	1296	1452	3860	3893	5109	5446	5471

34	1329	1485	3893	3926	5142	5483	5508
35	1366	1522	3930	3963	5179	5520	5545
36	1407	1563	3971	4004	5220	5557	5582
37	1440	1596	4004	4037	5253	5594	5619
38	1477	1633	4041	4074	5290	7717	7742
39	1514	1670	4078	4111	5327	7754	7779
40	1551	1707	4115	4148	5364	7791	7816
41	3653	3809	6217	6246	7462	7828	7853
42	3683	3839	6247	6280	7496	7865	7890
43	3717	3873	6281	6314	7530	7902	7927
44	3747	3903	6311	6344	7560	7939	7964
45	3777	3933	6341	6374	7590	7976	8001
46	3834	3990	6398	6431	7647	8013	8038
47	3871	4027	6435	6468	7684	8050	8075
48	3908	4064	6472	6505	7721	8087	8112
49	3978	4134	6542	6571	7787	8124	8149
50	4011	4167	6575	6608	7824	8161	8186
51	4042	4198	6606	6639	7855	8198	8223
52	4072	4228	6636	6669	7885	8235	8260
53	4102	4258	6666	6699	7915	8272	8297
54	4159	4315	6723	6756	7972	8309	8334
55	4196	4352	6760	6793	8009	8346	8371
56	4233	4389	6797	6830	8046	8383	8408
57	4303	4459	6867	6896	8112	8478	8503
58	4336	4492	6900	6933	8149	8515	8540
59	4373	4529	6937	6970	8186	8552	8577
60	4410	4566	6974	7007	8223	8589	8614
61	4447	4603	7011	7044	8260	8626	8651
62	4484	4640	7048	7081	8297	8663	8688
63	4521	4677	7085	7118	8334	8700	8725
64	4558	4714	7122	7155	8371	8737	8762
65	4628	4784	7192	7221	8437	8774	8799
66	4661	4817	7225	7258	8474	8811	8836
67	4698	4854	7262	7295	8511	8848	8873
68	4735	4891	7299	7332	8548	8885	8910
69	4772	4928	7336	7369	8585	8922	8947
70	4809	4965	7373	7406	8622	8959	8984
71	4846	5002	7410	7443	8659	8996	9021
72	4883	5039	7447	7480	8696	9033	9058
73	4953	5109	7517	7546	8762	9099	9124
74	4986	5142	7550	7583	8799	9136	9161
75	5023	5179	7587	7620	8836	9173	9198

76	5064	5220	7628	7657	8873	9210	9235
77	5097	5253	7661	7694	8910	9247	9272
78	5134	5290	7698	7731	8947	9284	9309
79	5171	5327	7735	7768	8984	9321	9346
80	5208	5364	7772	7805	9021	9358	9383

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	A	0	156	764	793	1109	1446	1471
2	B	55	211	2619	2648	3864	4201	4226
3	B	178	334	2742	2771	3987	4324	4349
4	A	203	359	967	996	1312	1649	1674
5	B	223	379	2787	2816	4032	4369	4394
6	A	425	581	1189	1218	1534	1871	1896
7	A	878	1034	1642	1671	1987	2324	2349
8	B	927	1083	3491	3520	4736	5073	5098
9	B	1026	1182	3590	3619	4835	5172	5197
10	B	1279	1435	3843	3872	5088	5425	5450
11	A	1586	1742	2350	2379	2695	3032	3057
12	A	1814	1970	2578	2607	2923	3260	3285
13	A	1870	2026	2634	2663	2979	3316	3341
14	A	2349	2505	3113	3142	3458	3795	3820
15	A	2402	2558	3166	3195	3511	3848	3873
16	B	2463	2619	5027	5056	6272	6609	6634
17	B	2482	2638	5046	5075	6291	6628	6653
18	A	2666	2822	3430	3459	3775	4112	4137
19	B	2859	3015	5423	5452	6668	7005	7030
20	B	3019	3175	5583	5612	6828	7165	7190
21	B	3047	3203	5611	5640	6856	7193	7218
22	B	3156	3312	5720	5749	6965	7302	7327
23	A	3199	3355	3963	3992	4308	4645	4670
24	B	3226	3382	5790	5819	7035	7372	7397
25	B	3249	3405	5813	5842	7058	7395	7420
26	A	3328	3484	4092	4121	4437	4774	4799
27	B	3484	3640	6048	6077	7293	7630	7655
28	A	3772	3928	4536	4565	4881	5218	5243
29	B	3808	3964	6372	6401	7617	7954	7979
30	A	3833	3989	4597	4626	4942	5279	5304
31	B	3851	4007	6415	6444	7660	7997	8022

32	B	3872	4028	6436	6465	7681	8018	8043
33	A	3933	4089	4697	4726	5042	5379	5404
34	A	3958	4114	4722	4751	5067	5404	5429
35	A	4144	4300	4908	4937	5253	5590	5615
36	B	4188	4344	6752	6781	7997	8334	8359
37	B	4191	4347	6755	6784	8000	8337	8362
38	A	4313	4469	5077	5106	5422	5759	5784
39	B	4342	4498	6906	6935	8151	8488	8513
40	A	4396	4552	5160	5189	5505	5842	5867
41	A	4422	4578	5186	5215	5531	5868	5893
42	B	4451	4607	7015	7044	8260	8597	8622
43	A	4478	4634	5242	5271	5587	5924	5949
44	A	4495	4651	5259	5288	5604	5941	5966
45	A	4569	4725	5333	5362	5678	6015	6040
46	B	4630	4786	7194	7223	8439	8776	8801
47	B	4688	4844	7252	7281	8497	8834	8859
48	A	4748	4904	5512	5541	5857	6194	6219
49	B	4794	4950	7358	7387	8603	8940	8965
50	A	4832	4988	5596	5625	5941	6278	6303
51	B	4866	5022	7430	7459	8675	9012	9037
52	A	4877	5033	5641	5670	5986	6323	6348
53	B	4953	5109	7517	7546	8762	9099	9124
54	A	5047	5203	5811	5840	6156	6493	6518
55	B	5066	5222	7630	7659	8875	9212	9237
56	A	5071	5227	5835	5864	6180	6517	6542
57	A	5080	5236	5844	5873	6189	6526	6551
58	A	5100	5256	5864	5893	6209	6546	6571
59	B	5198	5354	7762	7791	9007	9344	9369
60	A	5276	5432	6040	6069	6385	6722	6747
61	A	5284	5440	6048	6077	6393	6730	6755
62	A	5301	5457	6065	6094	6410	6747	6772
63	A	5323	5479	6087	6116	6432	6769	6794
64	A	5645	5801	6409	6438	6754	7091	7116
65	B	5679	5835	8243	8272	9488	9825	9850
66	B	5887	6043	8451	8480	9696	10033	10058
67	B	5949	6105	8513	8542	9758	10095	10120
68	B	5981	6137	8545	8574	9790	10127	10152
69	A	5994	6150	6758	6787	7103	7440	7465
70	A	6041	6197	6805	6834	7150	7487	7512
71	B	6203	6359	8767	8796	10012	10349	10374
72	B	6214	6370	8778	8807	10023	10360	10385
73	B	6299	6455	8863	8892	10108	10445	10470

74	A	6398	6554	7162	7191	7507	7844	7869
75	A	6418	6574	7182	7211	7527	7864	7889
76	A	6425	6581	7189	7218	7534	7871	7896
77	A	6482	6638	7246	7275	7591	7928	7953
78	A	6672	6828	7436	7465	7781	8118	8143
79	A	6847	7003	7611	7640	7956	8293	8318
80	B	6857	7013	9421	9450	10666	11003	11028
81	B	6922	7078	9486	9515	10731	11068	11093
82	A	6983	7139	7747	7776	8092	8429	8454
83	B	7025	7181	9589	9618	10834	11171	11196
84	B	7051	7207	9615	9644	10860	11197	11222
85	A	7056	7212	7820	7849	8165	8502	8527
86	B	7080	7236	9644	9673	10889	11226	11251
87	B	7196	7352	9760	9789	11005	11342	11367
88	B	7504	7660	10068	10097	11313	11650	11675
89	B	7526	7682	10090	10119	11335	11672	11697
90	B	7567	7723	10131	10160	11376	11713	11738
91	A	7690	7846	8454	8483	8799	9136	9161
92	B	7716	7872	10280	10309	11525	11862	11887
93	A	7779	7935	8543	8572	8888	9225	9250
94	B	7822	7978	10386	10415	11631	11968	11993
95	A	7845	8001	8609	8638	8954	9291	9316
96	A	7846	8002	8610	8639	8955	9292	9317
97	A	7880	8036	8644	8673	8989	9326	9351
98	A	7941	8097	8705	8734	9050	9387	9412
99	B	8071	8227	10635	10664	11880	12217	12242
100	A	8116	8272	8880	8909	9225	9562	9587
101	B	8232	8388	10796	10825	12041	12378	12403
102	A	8354	8510	9118	9147	9463	9800	9825
103	B	8559	8715	11123	11152	12368	12705	12730
104	B	8729	8885	11293	11322	12538	12875	12900
105	A	8899	9055	9663	9692	10008	10345	10370
106	B	8946	9102	11510	11539	12755	13092	13117
107	A	9155	9311	9919	9948	10264	10601	10626
108	A	9427	9583	10191	10220	10536	10873	10898
109	B	9527	9683	12091	12120	13336	13673	13698
110	B	9684	9840	12248	12277	13493	13830	13855
111	B	9692	9848	12256	12285	13501	13838	13863
112	A	9693	9849	10457	10486	10802	11139	11164
113	B	9698	9854	12262	12291	13507	13844	13869
114	A	9752	9908	10516	10545	10861	11198	11223
115	B	9780	9936	12344	12373	13589	13926	13951

116	B	9800	9956	12364	12393	13609	13946	13971
117	A	9809	9965	10573	10602	10918	11255	11280
118	A	9847	10003	10611	10640	10956	11293	11318
119	A	9873	10029	10637	10666	10982	11319	11344
120	B	10004	10160	12568	12597	13813	14150	14175
121	A	10017	10173	10781	10810	11126	11463	11488
122	B	10066	10222	12630	12659	13875	14212	14237
123	B	10069	10225	12633	12662	13878	14215	14240
124	A	10120	10276	10884	10913	11229	11566	11591
125	A	10322	10478	11086	11115	11431	11768	11793
126	A	10324	10480	11088	11117	11433	11770	11795
127	A	10413	10569	11177	11206	11522	11859	11884
128	A	10555	10711	11319	11348	11664	12001	12026
129	B	10599	10755	13163	13192	14408	14745	14770
130	A	10655	10811	11419	11448	11764	12101	12126
131	B	10691	10847	13255	13284	14500	14837	14862
132	A	10698	10854	11462	11491	11807	12144	12169
133	B	11073	11229	13637	13666	14882	15219	15244
134	B	11132	11288	13696	13725	14941	15278	15303
135	A	11161	11317	11925	11954	12270	12607	12632
136	B	11265	11421	13829	13858	15074	15411	15436
137	B	11413	11569	13977	14006	15222	15559	15584
138	B	11461	11617	14025	14054	15270	15607	15632
139	A	11496	11652	12260	12289	12605	12942	12967
140	B	11580	11736	14144	14173	15389	15726	15751
141	B	11600	11756	14164	14193	15409	15746	15771
142	A	11840	11996	12604	12633	12949	13286	13311
143	A	12064	12220	12828	12857	13173	13510	13535
144	B	12239	12395	14803	14832	16048	16385	16410
145	B	12262	12418	14826	14855	16071	16408	16433
146	B	12439	12595	15003	15032	16248	16585	16610
147	B	12488	12644	15052	15081	16297	16634	16659
148	A	12599	12755	13363	13392	13708	14045	14070
149	B	12687	12843	15251	15280	16496	16833	16858
150	A	12808	12964	13572	13601	13917	14254	14279
151	B	13185	13341	15749	15778	16994	17331	17356
152	B	13272	13428	15836	15865	17081	17418	17443
153	B	13283	13439	15847	15876	17092	17429	17454
154	B	13397	13553	15961	15990	17206	17543	17568
155	B	13934	14090	16498	16527	17743	18080	18105
156	A	14401	14557	15165	15194	15510	15847	15872
157	A	15016	15172	15780	15809	16125	16462	16487

158	A	16105	16261	16869	16898	17214	17551	17576
159	A	16410	16566	17174	17203	17519	17856	17881
160	A	16590	16746	17354	17383	17699	18036	18061

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	A	0	156	764	793	1109	1446	1471
2	B	163	319	2727	2756	3972	4309	4334
3	C	403	559	2067	2096	2712	3049	3074
4	C	438	594	2102	2131	2747	3084	3109
5	A	580	736	1344	1373	1689	2026	2051
6	B	582	738	3146	3175	4391	4728	4753
7	B	603	759	3167	3196	4412	4749	4774
8	A	656	812	1420	1449	1765	2102	2127
9	A	666	822	1430	1459	1775	2112	2137
10	B	960	1116	3524	3553	4769	5106	5131
11	C	985	1141	2649	2678	3294	3631	3656
12	A	996	1152	1760	1789	2105	2442	2467
13	A	1039	1195	1803	1832	2148	2485	2510
14	A	1137	1293	1901	1930	2246	2583	2608
15	C	1316	1472	2980	3009	3625	3962	3987
16	A	1378	1534	2142	2171	2487	2824	2849
17	C	1449	1605	3113	3142	3758	4095	4120
18	A	1506	1662	2270	2299	2615	2952	2977
19	A	1621	1777	2385	2414	2730	3067	3092
20	A	1865	2021	2629	2658	2974	3311	3336
21	B	1953	2109	4517	4546	5762	6099	6124
22	A	2001	2157	2765	2794	3110	3447	3472
23	A	2038	2194	2802	2831	3147	3484	3509
24	A	2049	2205	2813	2842	3158	3495	3520
25	C	2062	2218	3726	3755	4371	4708	4733
26	C	2163	2319	3827	3856	4472	4809	4834
27	B	2280	2436	4844	4873	6089	6426	6451
28	A	2303	2459	3067	3096	3412	3749	3774
29	A	2425	2581	3189	3218	3534	3871	3896
30	C	2433	2589	4097	4126	4742	5079	5104
31	A	2528	2684	3292	3321	3637	3974	3999
32	A	2585	2741	3349	3378	3694	4031	4056
33	A	2587	2743	3351	3380	3696	4033	4058

34	C	2731	2887	4395	4424	5040	5377	5402
35	B	2757	2913	5321	5350	6566	6903	6928
36	C	2842	2998	4506	4535	5151	5488	5513
37	B	2873	3029	5437	5466	6682	7019	7044
38	A	3043	3199	3807	3836	4152	4489	4514
39	A	3174	3330	3938	3967	4283	4620	4645
40	C	3190	3346	4854	4883	5499	5836	5861
41	A	3192	3348	3956	3985	4301	4638	4663
42	B	3233	3389	5797	5826	7042	7379	7404
43	C	3252	3408	4916	4945	5561	5898	5923
44	A	3264	3420	4028	4057	4373	4710	4735
45	A	3313	3469	4077	4106	4422	4759	4784
46	B	3429	3585	5993	6022	7238	7575	7600
47	C	3480	3636	5144	5173	5789	6126	6151
48	A	3515	3671	4279	4308	4624	4961	4986
49	B	3634	3790	6198	6227	7443	7780	7805
50	B	3649	3805	6213	6242	7458	7795	7820
51	C	3710	3866	5374	5403	6019	6356	6381
52	B	3839	3995	6403	6432	7648	7985	8010
53	C	3869	4025	5533	5562	6178	6515	6540
54	A	3924	4080	4688	4717	5033	5370	5395
55	A	3941	4097	4705	4734	5050	5387	5412
56	C	3956	4112	5620	5649	6265	6602	6627
57	A	4296	4452	5060	5089	5405	5742	5767
58	B	4378	4534	6942	6971	8187	8524	8549
59	A	4478	4634	5242	5271	5587	5924	5949
60	C	4554	4710	6218	6247	6863	7200	7225
61	C	4571	4727	6235	6264	6880	7217	7242
62	B	4685	4841	7249	7278	8494	8831	8856
63	B	4716	4872	7280	7309	8525	8862	8887
64	C	4721	4877	6385	6414	7030	7367	7392
65	C	4788	4944	6452	6481	7097	7434	7459
66	A	4788	4944	5552	5581	5897	6234	6259
67	B	4792	4948	7356	7385	8601	8938	8963
68	C	4847	5003	6511	6540	7156	7493	7518
69	B	4851	5007	7415	7444	8660	8997	9022
70	B	5007	5163	7571	7600	8816	9153	9178
71	C	5021	5177	6685	6714	7330	7667	7692
72	C	5103	5259	6767	6796	7412	7749	7774
73	A	5155	5311	5919	5948	6264	6601	6626
74	B	5166	5322	7730	7759	8975	9312	9337
75	B	5175	5331	7739	7768	8984	9321	9346

76	A	5549	5705	6313	6342	6658	6995	7020
77	B	5583	5739	8147	8176	9392	9729	9754
78	A	5595	5751	6359	6388	6704	7041	7066
79	C	5639	5795	7303	7332	7948	8285	8310
80	C	5664	5820	7328	7357	7973	8310	8335
81	A	5670	5826	6434	6463	6779	7116	7141
82	A	5697	5853	6461	6490	6806	7143	7168
83	B	5765	5921	8329	8358	9574	9911	9936
84	C	5772	5928	7436	7465	8081	8418	8443
85	B	5795	5951	8359	8388	9604	9941	9966
86	B	5914	6070	8478	8507	9723	10060	10085
87	C	5957	6113	7621	7650	8266	8603	8628
88	A	6031	6187	6795	6824	7140	7477	7502
89	A	6036	6192	6800	6829	7145	7482	7507
90	C	6036	6192	7700	7729	8345	8682	8707
91	A	6159	6315	6923	6952	7268	7605	7630
92	C	6204	6360	7868	7897	8513	8850	8875
93	B	6357	6513	8921	8950	10166	10503	10528
94	C	6492	6648	8156	8185	8801	9138	9163
95	A	6513	6669	7277	7306	7622	7959	7984
96	A	6609	6765	7373	7402	7718	8055	8080
97	B	6662	6818	9226	9255	10471	10808	10833
98	C	6722	6878	8386	8415	9031	9368	9393
99	B	6744	6900	9308	9337	10553	10890	10915
100	A	6774	6930	7538	7567	7883	8220	8245
101	A	6968	7124	7732	7761	8077	8414	8439
102	B	6971	7127	9535	9564	10780	11117	11142
103	A	7055	7211	7819	7848	8164	8501	8526
104	C	7144	7300	8808	8837	9453	9790	9815
105	A	7409	7565	8173	8202	8518	8855	8880
106	A	7421	7577	8185	8214	8530	8867	8892
107	C	7462	7618	9126	9155	9771	10108	10133
108	C	7557	7713	9221	9250	9866	10203	10228
109	B	7563	7719	10127	10156	11372	11709	11734
110	A	7667	7823	8431	8460	8776	9113	9138
111	C	7682	7838	9346	9375	9991	10328	10353
112	C	7759	7915	9423	9452	10068	10405	10430
113	B	7840	7996	10404	10433	11649	11986	12011
114	C	7865	8021	9529	9558	10174	10511	10536
115	B	7873	8029	10437	10466	11682	12019	12044
116	B	7876	8032	10440	10469	11685	12022	12047
117	C	7959	8115	9623	9652	10268	10605	10630

118	A	7971	8127	8735	8764	9080	9417	9442
119	A	7975	8131	8739	8768	9084	9421	9446
120	B	8087	8243	10651	10680	11896	12233	12258
121	C	8101	8257	9765	9794	10410	10747	10772
122	A	8316	8472	9080	9109	9425	9762	9787
123	B	8499	8655	11063	11092	12308	12645	12670
124	C	8502	8658	10166	10195	10811	11148	11173
125	A	8610	8766	9374	9403	9719	10056	10081
126	C	8641	8797	10305	10334	10950	11287	11312
127	C	8794	8950	10458	10487	11103	11440	11465
128	A	9015	9171	9779	9808	10124	10461	10486
129	C	9087	9243	10751	10780	11396	11733	11758
130	A	9403	9559	10167	10196	10512	10849	10874
131	B	9577	9733	12141	12170	13386	13723	13748
132	A	9730	9886	10494	10523	10839	11176	11201
133	A	9780	9936	10544	10573	10889	11226	11251
134	C	9928	10084	11592	11621	12237	12574	12599
135	A	10058	10214	10822	10851	11167	11504	11529
136	A	10179	10335	10943	10972	11288	11625	11650
137	B	10341	10497	12905	12934	14150	14487	14512
138	C	10412	10568	12076	12105	12721	13058	13083
139	B	10436	10592	13000	13029	14245	14582	14607
140	B	10447	10603	13011	13040	14256	14593	14618
141	C	10536	10692	12200	12229	12845	13182	13207
142	A	10984	11140	11748	11777	12093	12430	12455
143	B	11119	11275	13683	13712	14928	15265	15290
144	B	11157	11313	13721	13750	14966	15303	15328
145	B	11260	11416	13824	13853	15069	15406	15431
146	C	11507	11663	13171	13200	13816	14153	14178
147	B	11673	11829	14237	14266	15482	15819	15844
148	C	11846	12002	13510	13539	14155	14492	14517
149	B	11848	12004	14412	14441	15657	15994	16019
150	A	11991	12147	12755	12784	13100	13437	13462
151	B	12066	12222	14630	14659	15875	16212	16237
152	B	12278	12434	14842	14871	16087	16424	16449
153	B	12479	12635	15043	15072	16288	16625	16650
154	B	12616	12772	15180	15209	16425	16762	16787
155	A	12819	12975	13583	13612	13928	14265	14290
156	B	13508	13664	16072	16101	17317	17654	17679
157	B	13768	13924	16332	16361	17577	17914	17939
158	A	15016	15172	15780	15809	16125	16462	16487
159	C	15232	15388	16896	16925	17541	17878	17903

160	C	15534	15690	17198	17227	17843	18180	18205
-----	---	-------	-------	-------	-------	-------	-------	-------

附表 5: 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	B	0	156	2564	2593	3809	4146	4171
2	A	137	293	901	930	1246	1583	1608
3	A	382	538	1146	1175	1491	1828	1853
4	A	483	639	1247	1276	1592	1929	1954
5	B	486	642	3050	3079	4295	4632	4657
6	B	596	752	3160	3189	4405	4742	4767
7	B	660	816	3224	3253	4469	4806	4831
8	A	807	963	1571	1600	1916	2253	2278
9	B	983	1139	3547	3576	4792	5129	5154
10	A	1017	1173	1781	1810	2126	2463	2488
11	A	1286	1442	2050	2079	2395	2732	2757
12	A	1294	1450	2058	2087	2403	2740	2765
13	A	1393	1549	2157	2186	2502	2839	2864
14	A	1518	1674	2282	2311	2627	2964	2989
15	B	1729	1885	4293	4322	5538	5875	5900
16	A	1731	1887	2495	2524	2840	3177	3202
17	B	1769	1925	4333	4362	5578	5915	5940
18	C	1800	1956	3464	3493	4109	4446	4471
19	C	1872	2028	3536	3565	4181	4518	4543
20	B	1940	2096	4504	4533	5749	6086	6111
21	B	2108	2264	4672	4701	5917	6254	6279
22	B	2283	2439	4847	4876	6092	6429	6454
23	B	2296	2452	4860	4889	6105	6442	6467
24	A	2505	2661	3269	3298	3614	3951	3976
25	A	2582	2738	3346	3375	3691	4028	4053
26	A	2654	2810	3418	3447	3763	4100	4125
27	B	3068	3224	5632	5661	6877	7214	7239
28	B	3069	3225	5633	5662	6878	7215	7240
29	A	3088	3244	3852	3881	4197	4534	4559
30	A	3126	3282	3890	3919	4235	4572	4597
31	B	3254	3410	5818	5847	7063	7400	7425
32	A	3823	3979	4587	4616	4932	5269	5294
33	A	4047	4203	4811	4840	5156	5493	5518
34	B	4122	4278	6686	6715	7931	8268	8293
35	B	4141	4297	6705	6734	7950	8287	8312

36	A	4164	4320	4928	4957	5273	5610	5635
37	B	4188	4344	6752	6781	7997	8334	8359
38	A	4205	4361	4969	4998	5314	5651	5676
39	A	4240	4396	5004	5033	5349	5686	5711
40	B	4318	4474	6882	6911	8127	8464	8489
41	A	4345	4501	5109	5138	5454	5791	5816
42	B	4351	4507	6915	6944	8160	8497	8522
43	A	4471	4627	5235	5264	5580	5917	5942
44	C	4477	4633	6141	6170	6786	7123	7148
45	C	4488	4644	6152	6181	6797	7134	7159
46	B	4606	4762	7170	7199	8415	8752	8777
47	A	4730	4886	5494	5523	5839	6176	6201
48	B	4867	5023	7431	7460	8676	9013	9038
49	C	4872	5028	6536	6565	7181	7518	7543
50	A	4879	5035	5643	5672	5988	6325	6350
51	B	5062	5218	7626	7655	8871	9208	9233
52	A	5080	5236	5844	5873	6189	6526	6551
53	B	5204	5360	7768	7797	9013	9350	9375
54	B	5682	5838	8246	8275	9491	9828	9853
55	B	5741	5897	8305	8334	9550	9887	9912
56	B	6171	6327	8735	8764	9980	10317	10342
57	A	6177	6333	6941	6970	7286	7623	7648
58	A	6302	6458	7066	7095	7411	7748	7773
59	B	6336	6492	8900	8929	10145	10482	10507
60	A	6370	6526	7134	7163	7479	7816	7841
61	A	6455	6611	7219	7248	7564	7901	7926
62	B	6647	6803	9211	9240	10456	10793	10818
63	B	6650	6806	9214	9243	10459	10796	10821
64	B	6788	6944	9352	9381	10597	10934	10959
65	B	6877	7033	9441	9470	10686	11023	11048
66	B	6906	7062	9470	9499	10715	11052	11077
67	C	6941	7097	8605	8634	9250	9587	9612
68	A	6961	7117	7725	7754	8070	8407	8432
69	B	7040	7196	9604	9633	10849	11186	11211
70	A	7086	7242	7850	7879	8195	8532	8557
71	B	7184	7340	9748	9777	10993	11330	11355
72	B	7188	7344	9752	9781	10997	11334	11359
73	B	7250	7406	9814	9843	11059	11396	11421
74	B	7305	7461	9869	9898	11114	11451	11476
75	C	7484	7640	9148	9177	9793	10130	10155
76	B	7591	7747	10155	10184	11400	11737	11762
77	B	7648	7804	10212	10241	11457	11794	11819

78	A	7725	7881	8489	8518	8834	9171	9196
79	C	7784	7940	9448	9477	10093	10430	10455
80	C	7927	8083	9591	9620	10236	10573	10598
81	B	8063	8219	10627	10656	11872	12209	12234
82	C	8074	8230	9738	9767	10383	10720	10745
83	A	8237	8393	9001	9030	9346	9683	9708
84	A	8288	8444	9052	9081	9397	9734	9759
85	A	8477	8633	9241	9270	9586	9923	9948
86	B	8494	8650	11058	11087	12303	12640	12665
87	B	8608	8764	11172	11201	12417	12754	12779
88	B	8662	8818	11226	11255	12471	12808	12833
89	A	8744	8900	9508	9537	9853	10190	10215
90	B	8873	9029	11437	11466	12682	13019	13044
91	C	9019	9175	10683	10712	11328	11665	11690
92	A	9024	9180	9788	9817	10133	10470	10495
93	A	9069	9225	9833	9862	10178	10515	10540
94	B	9094	9250	11658	11687	12903	13240	13265
95	A	9095	9251	9859	9888	10204	10541	10566
96	B	9239	9395	11803	11832	13048	13385	13410
97	C	9306	9462	10970	10999	11615	11952	11977
98	A	9322	9478	10086	10115	10431	10768	10793
99	B	9333	9489	11897	11926	13142	13479	13504
100	A	9351	9507	10115	10144	10460	10797	10822
101	A	9459	9615	10223	10252	10568	10905	10930
102	A	9460	9616	10224	10253	10569	10906	10931
103	C	9509	9665	11173	11202	11818	12155	12180
104	A	9551	9707	10315	10344	10660	10997	11022
105	B	9561	9717	12125	12154	13370	13707	13732
106	A	9619	9775	10383	10412	10728	11065	11090
107	A	10134	10290	10898	10927	11243	11580	11605
108	C	10167	10323	11831	11860	12476	12813	12838
109	A	10172	10328	10936	10965	11281	11618	11643
110	B	10199	10355	12763	12792	14008	14345	14370
111	B	10499	10655	13063	13092	14308	14645	14670
112	C	10527	10683	12191	12220	12836	13173	13198
113	C	10618	10774	12282	12311	12927	13264	13289
114	B	10620	10776	13184	13213	14429	14766	14791
115	B	10691	10847	13255	13284	14500	14837	14862
116	A	10699	10855	11463	11492	11808	12145	12170
117	A	10740	10896	11504	11533	11849	12186	12211
118	A	10755	10911	11519	11548	11864	12201	12226
119	B	10781	10937	13345	13374	14590	14927	14952

120	A	10781	10937	11545	11574	11890	12227	12252
121	B	10806	10962	13370	13399	14615	14952	14977
122	A	10809	10965	11573	11602	11918	12255	12280
123	B	10887	11043	13451	13480	14696	15033	15058
124	B	10952	11108	13516	13545	14761	15098	15123
125	B	11243	11399	13807	13836	15052	15389	15414
126	C	11791	11947	13455	13484	14100	14437	14462
127	A	11883	12039	12647	12676	12992	13329	13354
128	B	11992	12148	14556	14585	15801	16138	16163
129	C	12133	12289	13797	13826	14442	14779	14804
130	A	12466	12622	13230	13259	13575	13912	13937
131	A	12754	12910	13518	13547	13863	14200	14225
132	C	12773	12929	14437	14466	15082	15419	15444
133	B	12955	13111	15519	15548	16764	17101	17126
134	A	13010	13166	13774	13803	14119	14456	14481
135	B	13033	13189	15597	15626	16842	17179	17204
136	A	13531	13687	14295	14324	14640	14977	15002
137	B	13557	13713	16121	16150	17366	17703	17728
138	A	13667	13823	14431	14460	14776	15113	15138
139	B	13676	13832	16240	16269	17485	17822	17847
140	B	13684	13840	16248	16277	17493	17830	17855
141	B	14040	14196	16604	16633	17849	18186	18211
142	B	14131	14287	16695	16724	17940	18277	18302
143	A	14197	14353	14961	14990	15306	15643	15668
144	B	14210	14366	16774	16803	18019	18356	18381
145	B	14296	14452	16860	16889	18105	18442	18467
146	B	14431	14587	16995	17024	18240	18577	18602
147	B	14466	14622	17030	17059	18275	18612	18637
148	B	14616	14772	17180	17209	18425	18762	18787
149	A	14675	14831	15439	15468	15784	16121	16146
150	B	14764	14920	17328	17357	18573	18910	18935
151	C	16268	16424	17932	17961	18577	18914	18939

支撑材料列表

序号	文件名	材料说明
1	问题 1-A.xlsx	问题 1-A 型芯片的调度结果
2	问题 1-B.xlsx	问题 1-B 型芯片的调度结果

3	问题 2.xlsx	问题 2 芯片调度结果
4	问题 3.xlsx	问题 3 芯片调度结果
5	问题 4.xlsx	问题 4 芯片调度结果
6	问题 5.xlsx	问题 5 芯片调度结果
7	问题 1-A.lg4	问题 1-A 型芯片处理的 Lingo 程序
8	问题 1-B.lg4	问题 1-B 型芯片处理的 Lingo 程序
9	芯片数_检测时间.py	芯片序号对应各工序起始时间绘图
10	芯片顺序.py	多类型芯片问题下芯片序号与入仓时间对照绘图
11	遗传算法问题 2.py	问题 2 处理程序
12	遗传算法问题 3.py	问题 3 处理程序
13	遗传算法问题 4.py	问题 4 处理程序
14	遗传算法问题 5.py	问题 5 处理程序

附录 6: 主要程序/关键代码

代 码 环 境	操作系统: Windows11 编程语言: python 3.9、LINGO 编辑器: Pycharm Community Edition 2022.03、LINGO18.0 代码详见: 问题 1-A.lg4、问题 1-B.lg4
------------------	--

代码清单 1 单类型芯片规划模型约束条件

!时间约束; @for(xinpian(i):T1(i)=T0(i)+150+6); @for(xinpian(i):T2(i)=T1(i)+40*60+8); @for(xinpian(i):T3(i)>T2(i)+21+8); @for(xinpian(i):T3(i)<T2(i)+25+8); @for(xinpian(i):T4(i)=T3(i)+20*60+16); @for(xinpian(i):T5(i)>T4(i)+325+12); @for(xinpian(i):T6(i)=T5(i)+25); !单空间约束; @for(xinpian(i) i#ne#1:T0(i)-T0(i-1)>30);!单独考虑前处理, 第i-1个进入下一阶段时, 第i个才能发起;
--

```

@for(xinpian(i)|i#gt#40:T1(i)>T1(i-40)+40*60+8);!单独考虑一温,第i-40个下转盘时,第i个还没上转盘;
@for(xinpian(i)|i#gt#1:T2(i)-T2(i-1)>21);!单独考虑加样,第i-1个离开加样台进入待机时(至少需要21秒),第i个才能上加样台;
@for(xinpian(i)|i#gt#40:T3(i)>T3(i-40)+20*60+16);!单独考虑二温,第i-40个下转盘时,第i个还没上转盘;
@for(xinpian(i)|i#gt#8:T4(i)-T4(i-8)>325);!单独考虑清洗,第i个芯片进入清洗盒的瞬间,第i-8个必须洗好离开;
@for(xinpian(i)|i#gt#1:T5(i)-T5(i-1)>25+12);!单独考虑检测,第i-1个芯片离开检测工位时,第i个芯片必须已经弹出了;

!多重入空间约束;
@for(xinpian(i)|i#gt#40:T1(i)>T3(i-40)+0.01);
@for(xinpian(i)|i#gt#40:T1(i)>T3(i-40)+20*60+16);

```

代码清单 2 FFSP 问题遗传算法优化适应度函数

```

def aim(AA, index): # 染色体矩阵
    RESULTS = []
    SHIZHI = []
    index_save = pd.ExcelWriter('D:/ArisMaster/MCM/2023/遗传算法绘图与数据/问题2/' + str(index) + '.xlsx')
    for line_index in range(len(AA)):
        A = AA[line_index].reshape(S, sum(num_Type)) # 编码矩阵
        "还原矩阵B, 即矩阵A的整数部分. 顺便矩阵C, 即矩阵A的小数部分, 并乘1000. "
        B = np.zeros((S, sum(num_Type))) # 矩阵A的整数部分
        C = np.zeros((S, sum(num_Type))) # 矩阵A的小数部分
        for i3 in range(S):
            for j3 in range(sum(num_Type)):
                B[i3, j3] = int(A[i3, j3])
                C[i3, j3] = round(1000 * (A[i3, j3] - int(A[i3, j3])), 0)
        "加工时间矩阵T"
        T = np.zeros((55, sum(num_Type)))
        # print(T)
        for si in S_M:
            if si in Time_machine_1:
                for ti in S_M[si]:
                    for tj in range(sum(num_Type)):
                        T[int(ti - 1), tj] = Time_machine_1[si]
            elif si == 6:
                for ti in S_M[si]:
                    T[int(ti - 1)] = np.array(Q) + 8
            else: # si == 8:
                for ti in S_M[si]:
                    T[int(ti - 1)] = np.array(R) + 16
        # print(T.shape) # 完成, (55, 160)
        "生成工序选取的机器矩阵S"
        matrixS = np.zeros((55, sum(num_Type)))
        for tj in range(sum(num_Type)):
            Btj = B[:, tj]
            # print(Btj)

```

```

        for ti in Btj:
            matrixS[int(ti - 1), tj] = 1
# print(matrixS.shape) # 完成, (55, 160)
"T • matrixS = TT, 点乘, 即一一对应相乘"
TT = T * matrixS
# print(T)
# print(matrixS)
# print(TT)
# print(TT.shape) # 完成, (55, 160)
"根据矩阵A含义和TT所示生成相应工序, 生成在所选机器上的完工时间矩阵ZT"
"将矩阵ZT改造成10*160"
ZT = np.zeros((10, sum(num_Type)))
# print(C)
# print(C.shape) # (10, 160)
ZT[0, :] = C[0, :] * 30 # 完成第一道工序的时间
for i in [2, 3, 4, 5]: # 第i道工序, i为2\3\4\5
    hang_index = S_M[i][0]
    for column_index in range(sum(num_Type)): # 对于160列也就是160个工件来说
        # 上一道工序的完成时间
        # 这一道工序的执行时间=TT[int(hang_index-1),lie_index]
        ZT[int(hang_index - 1), column_index] = ZT[int(hang_index - 2), column_index]
+ 30 * C[
            int(hang_index - 1), column_index]
# print(ZT)
ZT[4, :] = ZT[4, :] + 6 # 加上转动对位时间
# for i in [6]: #
"第6道工序, 即第一次温育过程"
for column_index in range(sum(num_Type)): # 对于160列也就是160个工件来说
    ZT[5, column_index] = ZT[4, column_index] # 先是上一道的完成时间
    duration = Q[column_index] + 8 # 这一道工序的执行时间
    machine_number = B[5, column_index] # 这一道工序的完成地点
    machine_queue = C[5, column_index] # 这一道工序在此机器的完成顺序
    # 要加上其他在这一机器执行一温的执行时间, 先找到在这里执行的其他芯片的索引
    all_indexs = np.where(B[5] == machine_number)[0]
    for all_index in all_indexs:
        if C[5, all_index] < C[5, column_index]:
            ZT[5, column_index] = ZT[5, column_index] + Q[all_index] + 8
    ZT[5, column_index] = ZT[5, column_index] + duration # 加上自己, over
"第7道工序, 即加样过程"
ZT[6, column_index] = ZT[5, column_index] + 21 * C[6, column_index] # 加上自
己, over
ZT[6, :] = ZT[6, :] + 8 # 加上转动对位时间
"第8道工序, 即第二次温育过程"
for column_index in range(sum(num_Type)): # 对于160列也就是160个工件来说
    ZT[7, column_index] = ZT[6, column_index] # 先是上一道的完成时间
    duration = R[column_index] + 16 # 这一道工序的执行时间
    machine_number = B[7, column_index] # 这一道工序的完成地点
    machine_queue = C[7, column_index] # 这一道工序在此机器的完成顺序

```

```

# 要加上其他在这一机器执行二温的执行时间, 先找到在这里执行的其他芯片的索引
all_indexs = np.where(B[7] == machine_number)[0]
for all_index in all_indexs:
    if C[7, all_index] < C[7, column_index]:
        ZT[7, column_index] = ZT[7, column_index] + R[all_index] + 16
    ZT[7, column_index] = ZT[7, column_index] + duration # 加上自己, over
"第9道工序, 即清洗过程"
for column_index in range(sum(num_Type)): # 对于160列也就是160个工作件来说
    ZT[8, column_index] = ZT[7, column_index] # 先是上一道的完成时间
    duration = 325 # 这一道工序的执行时间
    machine_number = B[8, column_index] # 这一道工序的完成地点
    machine_queue = C[8, column_index] # 这一道工序在此机器的完成顺序
    # 要加上其他在这一机器执行二温的执行时间, 先找到在这里执行的其他芯片的索引
    all_indexs = np.where(B[8] == machine_number)[0]
    for all_index in all_indexs:
        if C[8, all_index] < C[8, column_index]:
            ZT[8, column_index] = ZT[8, column_index] + 325
        ZT[8, column_index] = ZT[8, column_index] + duration # 加上自己, over
    "第10道工序, 即检测过程"
    ZT[9, column_index] = ZT[8, column_index] + (25 + 12) * C[9, column_index]
"ZT矩阵生成完毕, 现在开始计算等待时间, 改造ZT矩阵"
start_time = ZT[0]
wait_time = np.zeros(sum(num_Type))
wait6_5, wait7_6, wait8_7, wait9_8, wait10_9, more6_5, more7_6, more8_7, more9_8, more10_9 = np.zeros(
    sum(num_Type)), np.zeros(
    sum(num_Type)), np.zeros(sum(num_Type)), np.zeros(sum(num_Type)), np.zeros(sum(
    num_Type)), np.zeros(
    sum(num_Type)), np.zeros(sum(num_Type)), np.zeros(sum(num_Type)), np.zeros(sum(
    num_Type)), np.zeros(sum(num_Type))
    for column_index in range(sum(num_Type)):
        wait6_5[column_index] = ZT[5, column_index] - ZT[4, column_index] # 一温完毕
减前处理完毕
        wait7_6[column_index] = ZT[6, column_index] - ZT[5, column_index] # 加样完毕
减一温完毕
        wait8_7[column_index] = ZT[7, column_index] - ZT[6, column_index] # 二温完毕
减加样完毕
        wait9_8[column_index] = ZT[8, column_index] - ZT[7, column_index] # 清洗完毕
减二温完毕
        wait10_9[column_index] = ZT[9, column_index] - ZT[8, column_index] # 检测完毕
减清洗完毕
        more6_5[column_index] = wait6_5[column_index] - Q[column_index]
        more7_6[column_index] = wait7_6[column_index] - 21
        if more7_6[column_index] >= 4:
            more7_6[column_index] = more7_6[column_index] - 4
        more8_7[column_index] = wait8_7[column_index] - R[column_index]
        more9_8[column_index] = wait9_8[column_index] - 325
        more10_9[column_index] = wait10_9[column_index] - 25
        wait_time[column_index] = more6_5[column_index] + more7_6[column_index] + more8_7[column_index] + more9_8[
            column_index] + more10_9[
                column_index]

```

```

        "T0/T1/T2/T3/T4/T5/T6"
        T2, T3, T4, T5, T6 = np.zeros(sum(num_Type)), np.zeros(sum(num_Type)), np.zeros(sum
(num_Type)), np.zeros(
            sum(num_Type)), np.zeros(sum(num_Type))
        T00 = start_time + wait_time # 工序1结束加上等待时间, 将最小一项归零可得最终入仓
时间
        minT00 = T00.min()
        T0 = T00 - minT00
        T1 = T0 + 156
        for column_index in range(sum(num_Type)):
            T2[column_index] = T1[column_index] + Q[column_index] + 8
            T3[column_index] = T2[column_index] + 21 + 8
            T4[column_index] = T3[column_index] + R[column_index] + 16
            T5[column_index] = T4[column_index] + 325 + 12
            T6[column_index] = T5[column_index] + 25

        "目标函数"
        Cmax = T6.max()
        RESULTS.append(Cmax)
        # SHIZHI.append(math.exp(-Cmax))

    return np.array(RESULTS)

```

代码清单 3 遗传算法参数设置与过程

```

start_time = time.time() # 开始计时
NIND = 6000 # 初始种群大小
MAXGEN = 60 # 迭代次数
AA = initialization(NIND)
ObjV = aim(AA, 0) # 计算初始种群个体的目标函数值
best_ind = np.argmin(ObjV) # 计算当代最优个体的序号, 即最大值的索引
obj_trace = np.zeros((MAXGEN, 2)) # 定义目标函数值记录器
var_trace = np.zeros((MAXGEN, S * sum(num_Type))) # 定义染色体记录器, 记录每一代最优个
体的染色体

# 开始进化
for gen in range(MAXGEN):
    print(gen)
    FitnV = ea.ranking(ObjV.reshape((NIND, 1))) # 根据目标函数大小分配适应度值(由于遵循目
标最小化约定, 因此最大化问题要对目标函数值乘上-1)
    # print(FitnV.shape)
    SelCh = AA[ea.selecting('rws', FitnV, NIND - 1), :] # 选择, 采用'rws'轮盘赌选择RouletteWh
eelSelection
    SelCh = ea.recombin('xovsp', SelCh, 0.7) # 重组(采用两点交叉方式, 交叉概率为0.7)
    SelCh = ea.mutgau(SelCh, 0.1) # 十进制种群变异
    # 把父代精英个体与子代合并
    AA = np.vstack([AA[best_ind, :], SelCh])
    ObjV = aim(AA, gen + 1) # 求育种个体的目标函数值
    # print(ObjV.shape)

```

```

# 记录
best_ind = np.argmin(ObjV) # 计算当代最优个体的序号
obj_trace[gen, 0] = np.sum(ObjV) / NIND # 记录当代种群的目标函数均值
obj_trace[gen, 1] = ObjV[best_ind] # 记录当代种群最优个体目标函数值
# print(var_trace[gen, :].shape)
# print(AA[best_ind, :].shape)
var_trace[gen, :] = AA[best_ind, :] # 记录当代种群最优个体的变量值
print(time.time() - start_time)
end_time = time.time() # 结束计时
best_gen = np.argmin(obj_trace[:, [1]])
print('最优个体序号: ', best_gen)
print('目标函数最小值: ', obj_trace[best_gen, 1]) # 输出目标函数最大值
print('用时: ', end_time - start_time)
ea.trcplot(obj_trace, [['种群个体平均目标函数值', '种群最优个体目标函数值']])

```