

智能 RGV 的动态调度策略

摘要

本文根据题目给定的智能加工系统及系统作业参数，针对一道工序物料加工作业、两道工序物料加工作业、作业中故障处理等三种情况，建立数学模型，分别给出了相应的 RGV 最佳调度策略。

针对一道工序物料加工作业的情况，本文设计当 RGV 完成当前指令后若未接收到任何 CNC 的上料需求信号，RGV 将会根据调度模型立即判别执行一次移动指令，移动到下一步发出上料需求信号的 CNC 前。并将作业效率最佳问题转换为一班次 8 小时内 CNC 处于工作状态总时间最长，并假设 RGV 具有短时间的记忆储存功能，能够记录与匹配 RGV 与各 CNC 进行最后一次交互的时间，为 RGV 设计“八步一走”调度模型，在 RGV 进行移动指令之前都会遍历搜索选择未来八次移动过后八台 CNC 的总等待时间最小的路径的第一步移动指令作为当前的移动指令。遍历所有可能的初始八台 CNC 的上料情况，依据 RGV“八步一走”调度模型取成料数最多的初始 CNC 上料顺序，完成任务 1。将题目给定的针对一道工序的三组数据带入模型计算，得出第一组最大物料加工数量为 382，第二组为 359，第三组为 392；推算了不考虑 RGV 运动时间的理想状态下，三组数据的最大加工数量分别为 384、368、392；得到三组数据下加工系统的作业效率分别为 99.48%、97.55%、100%，完成任务 2。

针对两道工序物料加工作业的情况，在不可更换刀具的前提下，由第一道工序与第二道工序的比值，兼容考虑第二道工序之后的清洗时间，按比例分别为 CNC 安装 4:4、3:5、5:3 的刀具配比，并在对称性原则基础上调试具体安装方案；为 RGV 设计三步捆绑（或四步捆绑加工调度模型）：RGV 遍历三步，取捆绑加工后的完成时间最前的走法。遍历所有的初始可能路径，依据捆绑调度模型取成料数最多的初始 CNC 上料顺序，完成任务 1。将给定的针对两道工序的三组数据带入模型计算，得出三组最大物料加工数量分别为 253、209、236；选择的两类 CNC 数量配比分别为 4:4、3:5、5:3；通过与理想状态下最大物料加工数量 268、216、236 进行比较，得到三组数据下加工系统的作业效率分别为 94.40%、96.76%、100%，完成任务 2。

针对作业中故障处理的情况，本文将每一道工序加工的故障概率设为 1%，在判定故障的 CNC 的加工时间内，以均匀分布随机一个时间点作为故障发生时间点，并从 600~1200 秒之间均匀随机生成一个整数作为修复时间，在一道工序与二道工序的模型中作出以下调整：在故障发生的那一刻起，在 CNC 未修复并发出上料需求信号之前，将该 CNC 从系统中暂时抹去，RGV 在执行完当前指令后，不再进行有关该 CNC 的指令操作，直至 CNC 修复发出上料需求信号。考虑到故障发生的不确定性，以及人工修复时间的可操作性，在完成任任务的基础下，再分别取修复时间为 600~1200 秒随机，600 秒，900 秒，1200 秒做 20 组的随机试验探究成料数规律，进行均值和方差计算如下：一道工序的情况下，第一组数据关于 4 类修复时间的成料数方差分别为 12.20,9.55,11.95,9.82；第二组数据方差分别为 15.57,18.68,19.55,14.68；第三组数据方差分别为 10.03,13.41,8.92,13.73；两道工序的情况下，第一组数据关于 4 类修复时间的成料数方差分别为 9.66,7.38,7.12,13.17；第二组数据方差分别为 7.85,3.39,5.87,9.69；第三组数据方差分别为 7.66,4.58,7.72,10.13。由此可知，实际修复时，提升技工技术，将人工修复时间尽量控制在 10~15 分钟左右，可以较好增加结果稳定性。

关键词：RGV 智能动态调度 “八步一走” 多步捆绑联动 故障排查概率分析

一、问题重述

1.1 问题背景

RGV 是一种无人驾驶、能在固定轨道上自由运行的智能车。在智能加工系统中，轨道式自动引导车 RGV 的调度方案对系统的加工效率存在着决定性的影响。RGV 在智能加工系统中，因面临的工作环境各不相同，因此目前仍没有一个理想的算法可以对 RGV 的调度路径进行最佳优化，因此针对特定的系统工作环境，RGV 的动态调度仍存在着很大的研究空间。本文研究的工作环境由 8 台 CNC、1 辆 RGV、1 条 RGV 直线轨道、1 条上料传送带、1 条下料传送带等设备组成。

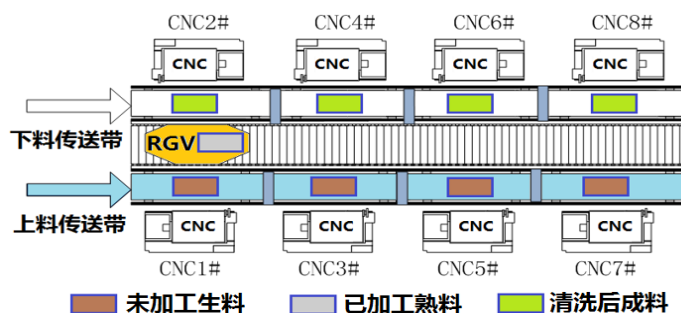


图 1：智能加工系统示意图

1.2 系统作业参数

表 1：智能加工系统作业参数的 3 组数据表

时间单位：秒

系统作业参数	第 1 组	第 2 组	第 3 组
RGV 移动 1 个单位所需时间	20	23	18
RGV 移动 2 个单位所需时间	33	41	32
RGV 移动 3 个单位所需时间	46	59	46
CNC 加工完成一个一道工序的物料所需时间	560	580	545
CNC 加工完成一个两道工序物料的第一道工序所需时间	400	280	455
CNC 加工完成一个两道工序物料的第二道工序所需时间	378	500	182
RGV 为 CNC1#、3#、5#、7#一次上下料所需时间	28	30	27
RGV 为 CNC2#、4#、6#、8#一次上下料所需时间	31	35	32
RGV 完成一个物料的清洗作业所需时间	25	30	25

注：每班连续作业 8 小时。

1.3 三种具体情况——工作环境

(1)一道工序的物料加工作业情况，每台 CNC 安装同样的刀具，物料可以在任一 CNC 上加工完成；

(2)两道工序的物料加工作业情况，每个物料的第一和第二道工序分别由两台不同的 CNC 依次加工完成；

(3)CNC 在加工过程中可能发生故障（故障发生概率约为 1%），每次人工处理故障（未完成的物料报废）需要耗时 10~20 分钟，故障排除后 CNC 即刻加入作业序列。要求分别考虑一道工序和两道工序的物料加工作业情况。

1.4 两个任务——问题要求

任务 1：对一般问题进行研究，给出 RGV 动态调度模型和相应的求解算法；

任务 2：利用表 1 中系统作业参数的 3 组数据分别检验模型的实用性和算法的有效性，给出 RGV 的调度策略和系统的作业效率，并将结果分别填入相应 EXCEL 表中。

二、问题分析

2.1 一道工序物料加工作业的问题分析

在该工作环境下，每台 CNC 都安装有相同的刀具，进行同一种加工工序，每一个生料均需在意八台 CNC 之一上完成一道工序的加工并由 RGV 清洗后方可成料。本文为提高系统加工效率，考虑到当所有 CNC 处于加工状态时，RGV 完成当前指令后，将会有长时间 CNC 不会对 RGV 发出上料需求信号，当 CNC 再次对 RGV 发出上料需求信号时，RGV 将从上一次指令完成位置进行移动指令，因此考虑若在无谓的等原地待之前 RGV 提前执行移动指令将会更好地节约时间。因此在为 RGV 设计调度模型时，设计当 RGV 完成当前指令后若未接收到任何 CNC 的上料需求信号，RGV 将会根据调度模型立即判别执行移动指令。为增大作业效率，即尽可能增加在一班次 8 小时内成料数量，即使得 CNC 尽可能多时间处于工作状态，换言之，即为 CNC 的总等待（闲置）时间尽可能小。综合以上考量，假设 RGV 具有短时间的记忆储存功能，能够记录与匹配 RGV 与各 CNC 进行最后一次交互的时间，由此时间为 RGV 设计“八步一走”调度模型，即 RGV 在判别执行移动指令之前，要遍历计算未来移动（或原地停留）八次，并要求每一次移动或停留过后必须间隔一道指令以上方可再次进行移动或停留指令，遍历搜索未来移动八次后八台 CNC 的总等待时间最小的路径的第一步移动指令（或原地停留）作为 RGV 的下一道指令，并要求实施检测判别完成当前指令后是否留有足够的时间在一班次 8 小时末回到初始点，若不能则拒绝接受有关移动回到初始点之外的任何指令。将表 1 系统作业参数的 3 组数据带入模型，给出 RGV 的调度策略，并估计理想状态下 RGV 运转速度足够快，忽略 RGV 的移动时间和清洗熟料的时间，进行计算此时情景下理想能加工完成的最大成料数，与模型结果进行比较估计系统的作业效率，并说明模型的实用性和算法的有效性

2.2 两道工序物料加工作业的问题分析

在该工作环境下，CNC 可选择安装两种刀具中的一种，对应进行两种加工工序，每一个生料均需在意八台 CNC 之一上完成一道工序的加工并由 RGV 清洗后方可成料。仍然考虑当为 RGV 设计调度模型时，当 RGV 完成当前指令后若未接收到任何 CNC 的上料需求信号，RGV 将会根据调度模型立即判别执行移动指令。此时需要考虑 CNC 上安装的刀具可以在初始固定后不再改变，也可以在 CNC 未进行加工时进行更换，这使得问题分为更换刀具与不更换刀具两种情况考虑。对于不更换刀具的情况，需要考虑安装不同刀具的 CNC 的配比及放置位置。就放置位置而言，可以根据简单的推算得出一个最佳方案；就配比而言，我们需要先筛选出可能出现最佳调度方案的几种情况，再对这几种情况分别建立模型完成调度。对每一种配比情况，由于该问题有两道工序，以 CNC 等待时间最短为目标会使计算更加复杂，因此直接以物料最大加工数量为目标。针对每一配比情况，会有一种最佳的分组方案，即以固定个数的物料完成加工的过程为一组，按照组序进行逐步调度。对每一组，有若干可能的顺序完成组内所有步骤，取其中时间最短的顺序方案，将每一组的方案组合成最终的调度方案，即为所求的模型。对于更换刀具的情况，需要考虑其更换刀具的频率、提高的加工效率，以此判断更换刀具的模型是否有意义，若无意义，则忽略此情形。将表 1 系统作业参数的 3 组数据带入模型，给出 RGV 的调度策略，并估计理想状态下 RGV 运转速度足够快，忽略 RGV 的移动时间和清洗熟料的时间，进行计算此时情景下理想能加工完成的最大成料数，与模型结果进行比较估计系统的作业效率，并说明模型的实用性和算法的有效性。

2.3 作业中故障处理的问题分析

在该工作环境下，CNC 在加工过程中约有 1% 的概率发生故障，故障发生时间出现

在 CNC 加工过程中的任一时刻的概率均相同，对故障进行人工排查修复处理需要耗时为 10~20 分钟，故障排除后 CNC 即刻加入作业序列。假设一道工序物料加工过程和两道工序的各步物料加工过程中故障发生的概率相同，皆视为 1%。每一道工序加工均有 1% 的概率发生故障，在 CNC 的加工时间内，以均匀分布随机一个时间点作为故障发生时间点，并以 10~20 分钟，即 600~1200 秒之间均匀随机生成一个整数作为修复时间。对一道工序物料加工作业时间，两道工序物料加工作业的模型作出以下调整：在故障发生的那一刻起，在 CNC 未修复并发出上料需求信号之前，将该 CNC 从系统中暂时抹去，RGV 在执行完当前指令后，不再进行有关该 CNC 的指令操作，直至 CNC 修复发出上料需求信号。考虑到故障发生的不确定性，模型结果的稳定性与能够人工干预的故障修复时间息息相关，因此本文在建模求解 600~1200 秒之间均匀随机生成一个整数作为修复时间后，还将分别取修复时间为 600 秒，900 秒，1200 秒做 20 组的随机试验，查验当将修复时间尽量控制在哪一个范围内，模型结果的稳定性将会更好。

三、模型假设

- 1.假设对于确定的某一工序，CNC 加工时间恒定且一致。
- 2.假设传送带会将物料及时运送到上料处，不产生等待时间
- 3.不考虑两道工序间废渣的处理，即假设第一道工序加工产生的废渣不会对第二道工序的加工产生影响。
- 4.假设 RGV 在接收到 CNC 发出的上料需求指令之后，当 RGV 移动到该 CNC 位置前时，RGV 才会对其进行上下料。
- 5.假设 RGV 具有短时间的记忆储存功能，能够记录与匹配与各 CNC 进行最后一次交互的时间。
- 6.假设一道工序物料加工过程和两道工序的各步物料加工过程中故障发生的概率相同，皆视为 1%。
- 7.假设 RGV 性能良好，严格按照各组参数时间按时执行指令。

四、符号说明

符号	符号意义
A_i	编号为 i 的 CNC 的状态
H	所有 CNC 的总等待时间
t_i	移动 i 个单位所需要的时间
u_{1k}	上料开始时间点
u_{2k}	下料开始时间点
h_{ik}	表示第 i 台 CNC 对第 k 个物料的等待时间
tt	CNC 完成一个物料的加工需要时间

H_v	第 v 个初始情况对应的全过程 CNC 总等待时间
N_i	理想情况一个班次能生产的成品数
η_i	系统的作业效率
C_i	第 i 台 CNC 的故障状态

五、模型的建立和求解

5.1 一道工序加工模型建立与求解

5.1.1 任务 1

任务 1 要求给出一般情况下 RGV 的动态调度模型与相应的求解算法。一般情况表示适应可选范围内的任意情况，即不能仅仅建立适应于某一组或几组参数值的模型，而要给出对任何可选参数值均适应的一般化模型。因此，模型的建立不能基于任何特定参数值、依赖任何特殊情形；同理，相应的求解算法也不能针对任何一组特定参数，应当在给出任何一组可取参数时，都能给出相应的理想解。

5.1.1.1 智能加工系统的工作原理

问题展示的智能加工系统主要由三部分构成，即 8 台计算机数控机床（CNC）、1 辆轨道式自动引导车（RGV）与 1 条 RGV 直线轨道、1 条上料传送带与 1 条下料传送带，每个物料完成加工都需要经过三个部分的联动工作。因此，首先需要明确每个部分的工作原理，并对每个部分的工作过程准确进行定量描述。

1) CNC 的工作原理

在整个加工过程中，CNC 共有空闲状态、上下料状态、加工状态、需求状态等 4 个状态。其中，空闲状态表示 CNC 上没有物料的状态；上下料状态表示 RGV 正在给 CNC 上下料的状态；加工状态表示 CNC 正在对物料进行加工的状态；需求状态表示 CNC 完成加工后等待 RGV 下料的状态。问题给出的加工系统中有 8 台已编号的 CNC，每台 CNC 的具体位置如图所示：

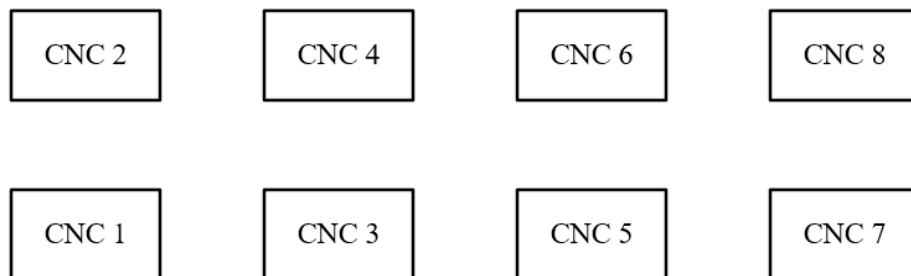


图 2 CNC 位置示意图

由图可知，横向上相邻两台 CNC 的间隔相等；纵向上 4 组 CNC 两两位于同一竖直线。

设编号为 i 的 CNC 的状态为 A_i ，并对各状态进行赋值：

$$A_i = \begin{cases} 0, & \text{空闲状态} \\ 1, & \text{上下料状态} \\ 2, & \text{加工状态} \\ 3, & \text{需求状态} \end{cases} \quad (1)$$

其中，0、3 表示的状态 CNC 未进行工作，1、2 表示的状态 CNC 正在进行工作；在实际情况下，我们期望 CNC 进行工作的时间尽可能长。

2) RGV 的工作原理

在整个加工过程中，RGV 始终位于 RGV 直线轨道上，共有等待状态、运动状态、上下料状态、清洗状态等 4 个状态。其中，等待状态表示 RGV 静止在轨道上不进行任何工作的状态；运动状态表示 RGV 由一个位置驶向另一个位置时的状态，此状态不进行工作；上下料状态表示 RGV 正在给 CNC 进行上下料的状态；清洗状态表示 RGV 给物料进行清洗并将其放置到下料传送带时的状态^[1]。

设 RGV 的状态为 B ，并对各状态进行赋值：

$$B = \begin{cases} 0, & \text{等待状态} \\ 1, & \text{运动状态} \\ 2, & \text{上下料状态} \\ 3, & \text{清洗状态} \end{cases} \quad (2)$$

其中，0、2、3 状态时，RGV 只能处于轨道给定 4 个位置中的任意一个；1 状态时，RGV 正在由其中一个位置运动至另一个位置。当 RGV 处于某一位置时，只能与该位置的上下两台 CNC 进行联动。

设 j 为 RGV 所处的位置，则：

$$j = \begin{cases} 1, & i = 1 \text{ or } i = 2 \\ 2, & i = 3 \text{ or } i = 4 \\ 3, & i = 5 \text{ or } i = 6 \\ 4, & i = 7 \text{ or } i = 8 \end{cases} \quad (3)$$

其中， i 表示 CNC 的编号。

当 RGV 处于 1 状态时，设其运动状态为：

$$(j_1, j_2) \quad (4)$$

表示 RGV 此时正由位置 j_1 运动至位置 j_2 。

3) 传送带的工作原理

传送带在加工系统启用时仅有一个状态，即运动状态。上料传送带将负责运送未加工的生料，下料传送带负责运送已加工的熟料。已知传送带既可连动，也可独立运动；在实际中，传送带的运动速度也能达到数米每秒。基于上述两点，不妨假设当 RGV 需要时，传送带总能在相应位置给出生料或熟料空位，无需等待。

5.1.1.2 加工系统的调度时间

对于一道工序的加工系统而言，首要的目标是尽可能提高加工效率。最能直接反应这一目标的即为相同时间内加工的物料个数，个数越大，加工效率越高。然而物料的加工是一个复杂动态过程，物料在不同时间可能处于 CNC 中，也可能处于 RGV 中，我们需要进一步将目标进行转化。

容易得到，当物料处于加工状态时，相应的 CNC 必定处于工作状态；当物料处于非加工状态时，不对应工作时间的 CNC。因此，不妨将目标转化为尽可能减少 CNC 的非工作时间，即需要给出一个调度模型，使得 CNC 的非工作总时间最小。

1) 总时间的确定

加工系统的连续工作时间为 8 小时，不妨设 8 小时为一次完整的加工过程，目标为 8 小时内 CNC 的非工作总时间最小。由于上下料，位置移动等指令在实际中可以在几十秒内完成，因此将时间单位转化到秒，则一次加工过程的总时间为：

$$T = 8h = 28800s \quad (5)$$

则对于一次完整加工过程而言，第 0s 加工系统启动，RGV 由初始位置 1 开始执行指令；第 28800s 加工系统终止，RGV 回到初始位置 1。

2) 时间参数的确定

对于整个加工系统，需要相应的时间参数反应调度过程。

首先，RGV 需要在 4 个不同位置间来回移动，可能连续移动的位置单位有 1、2、3。因此，设 RGV 移动一个单位的时间为 t_1 ，移动两个单位的时间为 t_2 ，移动 3 个单位的时间为 t_3 。除此以外，RGV 自身拥有清洗物料的工作，故设完成一次清洗所需的时间为 t_4 。

其次，CNC 需要时间单独对物料进行加工，设 CNC 完成一个物料的加工需要时间 tt 。

最后，对于 RGV 与 CNC 的联动，由于实际中 RGV 与两边 CNC 的位置非完全对称，我们设 RGV 对 1、3、5、7 号 CNC 完成一次上下料所需时间为 s_1 ，对 2、4、6、8 号 CNC 完成一次上下料所需时间为 s_2 。

3) 时间变量的确定

为了给出最佳调度方案，需要对未知、不确定的时间变量进行确定。首先考虑 CNC 的总等待时间，不妨设为 H ，则有：

$$H = \sum_{i=1}^8 \sum_{k=1} h_{ik} \quad (6)$$

其中， h_{ik} 表示第 i 台 CNC 对第 k 个物料的等待时间，物料编号 k 根据 RGV 抓取生料的顺序来计。

进一步考虑物料 k ，设其上料开始时间点为 u_{1k} ，下料开始时间点为 u_{2k} ，则有：

$$u_{1k} - u_{2k} = tt + s \quad (7)$$

其中 s 表示 s_1 或 s_2 ，具体取值根据 k 情况而定。

5.1.1.3 加工系统的优化模型

1) 目标函数的确定

对于加工系统的一次 8 小时运行而言，需要给出调度方案使得 CNC 总等待时间最少，则建立目标函数如下：

$$\min H \quad (8)$$

H 表示 8 小时内的 CNC 总等待时间。

2) 初始方案的规划

Step1: 首先考虑初始情况。已知 RGV 的初始位置为左端位置 1，CNC 均属于空闲状态，而实际情况下，一次上料或移动的时间又远小于 CNC 完成一次加工的时间。综上，初始的指令确定为按照一定顺序完成对 8 台 CNC 的上料。

Step2: 当 RGV 在某一位置时，对该位置的两台 CNC 依次完成下料期间无需移动，

而对其它位置的 CNC 完成下料需要额外的移动时间。因此获得结论：当 RGV 对某一位置的一台 CNC 进行上下料时，下一个上下料的目标优先选择该位置的另一台 CNC；RGV 首先对初始位置 1 的两台 CNC 进行上料。

Step3: 由上述两个结论，RGV 完成初始对 8 台 CNC 的上料共有 6 种情况：

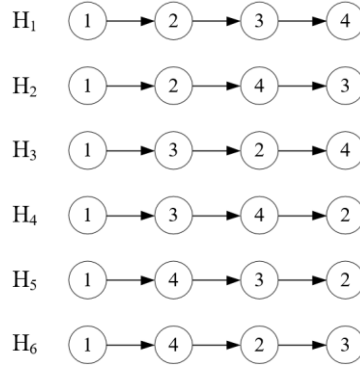


图 3 初始上料情况图

故目标函数转化为：

$$\min \{H_1, H_2, H_3, H_4, H_5, H_6\} \quad (9)$$

其中 H_v 表示第 v 个初始情况对应的调度方案中 CNC 的总等待时间。

3) 调度方案的优化模型

Step1: 对任一初始方案，完成 8 台 CNC 的上料后处于位置 j 。考虑下一步 RGV 的动向，虽然在后续加工中多出清洗的步骤，但实际中过长的加工时间仍会使 RGV 优先进行完一轮对 8 台 CNC 的上下料。因此考虑 RGV8 次连续完整作业的动态规划。

Step2: 对于整个加工过程，RGV 的工作流程如下：

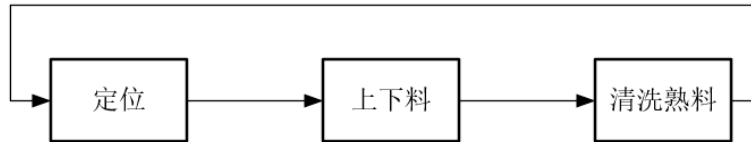


图 4 RGV 一次完整作业示意图

考虑 8 次如图所示的 RGV 一次完整作业，当 RGV 对某台 CNC 进行上下料时，其它 CNC 的加工可能已经完成并进行需求提示，则在 RGV 到达前，这些 CNC 都将产生等待时间；此外，根据之前的结论，RGV 会优先对同一位置的两台 CNC 依次作业。综上所述，对接下来的 8 次工作，共有 4^8 种调度方案。

Step3: 对任意一种调度方案，设 RGV 每进行一次作业操作，都会使第 i 个 CNC 产生等待时间 h_{im} ，其中 m 表示 8 次连续作业中的第 m 次。则这种调度方案 8 个步骤产生的 CNC 等待总时间为：

$$h_n = \sum_{m=1}^8 \sum_{i=1}^8 h_{im} \quad (10)$$

特别地，这里 $n=1$ ，表示进行全过程中的第一个 8 次连续作业。

Step4: 求出每一种调度方案下的总等待时间，共有 4^8 个结果，取最小总等待时间对应的方案，对应目标函数如下：

$$\min h_n = \sum_{m=1}^8 \sum_{i=1}^8 h_{im} \quad (11)$$

该最优方案对应的 8 次连续作业中，第一步中第 i 个 CNC 的最小等待时间为 h_{i1} 。

Step5: 选取 8 次连续作业中的第一次，令 RGV 执行该指令，完成调度方案中的一步。则该步骤中 8 个 CNC 产生的总等待时间为：

$$hh_n = \sum_{i=1}^8 h_{i1} \quad (12)$$

以第一个 8 次连续作业中确定的第一次结束作为开始，重复 Step3、4 的步骤，进行若干个 8 次连续作业，每个 8 次连续作业确定实际的一次作业，直到 28800 秒的工作流程结束。对结束前的状态进行讨论，若最后一个 8 次连续作业的产生实际的一次作业无法在第 28800 秒前完成并令 RGV 返回初始位置，则放弃该次作业，令 RGV 直接回到初始位置。则对第 v 个初始情况对应的完整调度方案，有：

$$H_v = \sum_{n=1} hh_n \quad (13)$$

其中， H_v 表示第 v 个初始情况对应的全过程 CNC 总等待时间。

Step6: 综合 Step1-5，得到完整的优化模型如下：

$$\begin{aligned} & \min \{H_1, H_2, H_3, H_4, H_5, H_6\} \\ & \text{s.t.} \begin{cases} H_v = \sum_{n=1} hh_n \\ hh_n = \sum_{i=1}^8 h_{i1} \\ \min h_n = \sum_{m=1}^8 \sum_{i=1}^8 h_{im} \end{cases} \end{aligned} \quad (14)$$

5.1.1.4 调度模型的求解算法

模型求解的前提为给定一组可靠的参数数据，关于参数的具体内容，已经在先前给出详细说明。

1) 优化模型求解

对于 4^8 种调度方案，在 MATLAB 中使用遍历搜索算法进行求解，得到总等待时间最小的方案；对于其余过程，均可利用 MATLAB 进行直接计算。

2) 初始目标求解

但此时我们必须考虑到，CNC 总等待时间最小值是我们为方便求解而转化得到的目标。利用上述模型求解到最优方案后，需将目标转化回最初状态，求解得到最多的物料加工数，并给出相应的上下料开始时间点。

已知物料编号 k 根据 RGV 抓取生料的顺序来计，其上料开始时间点为 u_{1k} ，下料开始时间点为 u_{2k} 。利用 MATLAB 对整个最优调度方案进行遍历搜索，得到第 k 个物料编号，第 k 个上料开始时间点 u_{1k} ，第 k 个下料开始时间点 u_{2k} ，并记录所有 k 对应的结果。

3) 结果总结

经过上述求解步骤，我们最终可以得到最优调度方案、最小总等到时间、最大物料加工数、每个被加工物料对应的上下料开始时间。

5.1.2 任务 2

5.1.2.1 一道工序加工模型的结果

将题目给出的三组参数分别带入模型，利用 MATLAB 编程求解(代码文件 Ques1.m 详见附录五)。得到的第一组结果如下表：

表 2 一道工序调度模型第一组数据结果表

加工物 料序号	加工 CNC 编 号	上料开 始时间	下料开 始时间
1	1	0	588
2	2	28	641
3	3	79	717
.....			
380	4	27956	28547
381	5	28032	28623
382	6	28085	28676

第二组最优结果如下表（两个最优序列）：

表 3-1 一道工序调度模型第二组数据结果表

加工物 料序号	加工 CNC 编号	上料开 始时间	下料开 始时间
1	1	0	610
2	2	30	670
3	3	88	758
4	4	118	818
5	7	194	924
6	8	224	984
7	5	282	1072
8	6	312	1132
.....			
359	5	28076	28704

表 3-2 一道工序调度模型第二组数据结果表

加工物 料序号	加工 CNC 编号	上料开 始时间	下料开 始时间
1	1	0	610
2	2	30	670
3	3	88	758
4	4	118	818
5	5	176	906
6	6	206	966
7	7	264	1054
8	8	294	1114
.....			
359	7	28058	28686

第三组的最优结果如下表：

表 4 一道工序调度模型

加工物 料序号	加工 CNC 编号	上料开 始时间	下料开 始时间
1	1	0	572
2	2	27	624
3	3	77	699
.....			
390	6	27997	28574
391	7	28072	28649
392	8	28124	28701

5.1.2.2 一道工序模型的实用性

从结果来看，模型针对 3 组参数的求解情况均符合实际，既没有出现异常的上下料时间、也没有异常的两个相邻编号的物料上下料开始时间间隔。而这 3 组数据各有特点，反应了不同工序的不同耗时配比，故 3 组结果均合理足以证明模型具有较强的实用性，可针对任何符合实际的一组参数进行求解，得到合理的最佳效率方案。

分析最优的结果序列，发现恰好为 1-8 循环作业的自然序列。这与实际的加工过程中，RGV 的运动通常存在某种规律、不会做无规则的复杂运动相符合，进一步证明了模型的实用性。

5.1.2.3 一道工序模型系统的作业效率

首先计算最理想情况下即仅考虑物料加工时间和上下料时间，8 小时能制造出多少个成品^[2]。

理想初始情况下，对 8 台 CNC 进行上料操作，忽略 RGV 移动所消耗的时间，记第 i 组初始上料的总时间为 t_{0i} ：

$$t_{0i} = 4s_1 + 4s_2 \quad (15)$$

理想情况下第 i 组加工的工件数量 N_i 为：

$$N_i = [(T - t_{0i}) / (t_{ti} + s_1)] \times 4 + [(T - t_{0i}) / (t_{ti} + s_2)] \times 4 \quad (16)$$

t_{ti} 表示第 i 组单个物料的加工时间， T 表示一个班次的总时间。[] 表示对括号内的式子向下取整。

将 3 组数据的参数分别带入上式得到： $N_1 = 384$ ， $N_2 = 368$ ， $N_3 = 392$ 。需注意，这是理想情况下求得的 8 小时最多能加工的零件，并没有考虑一开始小车路上移动的时间和因 RGV 未能及时到达发出需要上下料的 CNC 位置而延误的时间。所谓理想情况即 RGV 总能及时达到需要上下料的 CNC 位置。

第 i 组的作用效率如下：

$$\eta_i = \frac{n_i}{N_i} \times 100\% \quad (17)$$

n_{ij} 表示利用情况 i 的模型求得的第 j 组的成品数量。

得到： $\eta_{11} = 99.48\%$ ， $\eta_{12} = 97.55\%$ ， $\eta_{13} = 100\%$ 。

5.2 两道工序加工模型建立与求解

5.2.1 任务 1

对两道工序的情况而言，加工系统大部分的工作原理都与一道工序相同，仅少数发生了改变。因此，对没有改变的工作原理、参数、变量，均沿用情况一中的定义。

5.2.1.1 智能加工系统的工作原理

1) CNC 的工作原理

在两道工序中，CNC 的位置、工作状态仍与一道工序相同，不同之处在于原先 8 台 CNC 都进行同一道工序，而此时的 CNC 可以通过更换刀具改变可加工工序。我们设第 i 台 CNC 对应的可加工工序为 G_i ，则：

$$G_i = \begin{cases} 1, & \text{加工第一道工序} \\ 2, & \text{加工第二道工序} \end{cases} \quad (18)$$

2) RGV 的工作原理

对任意一块物料，当完成第一道工序时，RGV 需要将物料转移到能进行第二道工序的 CNC 中。在此过程中，第一道工序产生的碎屑可能需要进行处理，否则无法直接进行第二道工序。但由于此过程实际耗费的时间极小，考虑在模型中忽略这个时间。

考虑 RGV 的工作状态，在两道工序中，上下料状态显然会出现两种情况，即给加工第一道工序的 CNC 上下料和给加工第二道工序的 CNC 上下料。因此，设 RGV 的状态为 B_1 ，并对各状态进行赋值：

$$B_1 = \begin{cases} 0, & \text{等待状态} \\ 1, & \text{运动状态} \\ 2, & \text{第一道工序上下料状态} \\ 3, & \text{第二道工序上下料状态} \\ 4, & \text{清洗状态} \end{cases} \quad (18)$$

5.2.1.2 加工系统的调度时间

对于两道工序的加工系统而言，首要的目标仍然是尽可能提高加工效率。最能直接反应这一目标的即为相同时间内加工的物料个数，个数越大，加工效率越高。

然而区别于一道工序，两道工序的目标是不能盲目转换为 CNC 总等待时间最小的。对每一个物料，需要完成两道工序才能完成加工，而能加工每到工序的 CNC 编号是不确定的，故 CNC 总等待时间最小不能直接反应加工物料数最多，可能会出现更多半成品的情况。因此，两道工序的目标转化需要根据 CNC 具体调配情况而确定。

1) 时间参数的确定

不同加工工序的 CNC 加工物料的时间不同，因此设加工第一道工序的 CNC 单次加工时间为 tt_1 ，加工第二道工序的 CNC 单次加工时间为 tt_2 。

2) 时间变量的确定

根据 RGV 抓取生料的顺序，将第 k 个被抓取的生料记为物料 k 。则对每个物料 k ，设其第一道工序上料开始时间点为 u_{3k} ，下料开始时间点为 u_{4k} ；第二道工序上料开始时间点为 u_{5k} ，下料开始时间点为 u_{6k} 。

已知 RGV 对 1、3、5、7 号 CNC 完成一次上下料所需时间为 s_1 ，对 2、4、6、8 号 CNC 完成一次上下料所需时间为 s_2 。则对第一道工序，有：

$$u_{4k} - u_{3k} = tt_1 + s \quad (19)$$

对第二道工序，有：

$$u_{6k} - u_{5k} = tt_2 + s \quad (20)$$

其中 s 表示 s_1 、 s_2 中对应 k 的情况。

5.2.1.3 加工系统的优化模型

对于两道工序的加工而言，问题给出两条准则：CNC 在加工的过程中不能更换刀具；同一块物料的两道工序必须在不同的两台 CNC 上进行加工。分析这两条准则，我们不难发现准则在给出限定的同时也透漏了一条关键信息：CNC 在非加工时间可以更换刀具。因此，两道工序的加工过程中，CNC 可以持续的进行动态变化。

针对这一变化，我们不妨进行分类讨论，考虑可更换刀具与不可更换刀具两种情况，给出不同的模型使加工系统达到效率最高。进一步考虑不可更换刀具的情况，此情况下初始的 CNC 可加工工序设定即为整个流程的设定。首先排除 8 台 CNC 可加工工序相同的情况，此状态无法进行完整加工。于是可能出现的两道加工工序的 CNC 台数比为 1:7、

2:6、3:5、4:4。

综上，得到两道工序的加工系统的考虑分类情形如图所示：

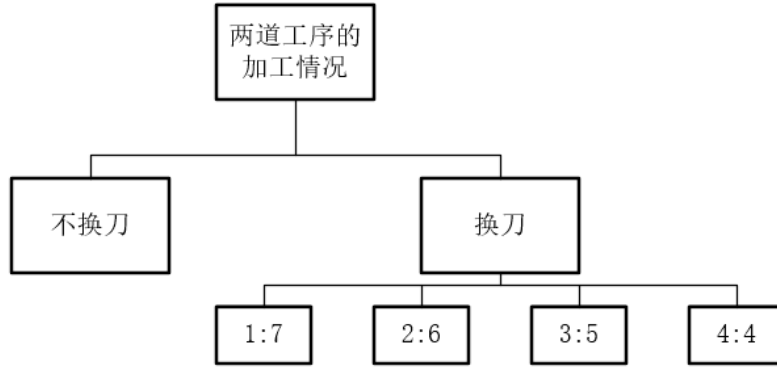


图 5 两道工序加工情况的分类图

● 不更换刀具

1) CNC 的配比方案

先前已经分析得出可能出现的两道加工工序的 CNC 台数比为 3:5、4:4，而具体应该使用怎样的配比，主要取决于两道工序的加工时间^[3]。例如，当两道工序的加工时间相近时，适合使用 4:4 的 CNC 配比；当一道工序的加工时间是另一道工序的 3 倍时，使用 4:4 会导致产生过长的 CNC 闲置等待时间，使用 2:6 的配比较为合理。但根据同一位置优先的法则，当比例为 2:6 或 7:1 时，模型会因法则失效程度过大而使可行性降低，且实际中两道工序的加工 CNC 比例不会严重失衡，故我们不考虑 2:6、1:7 的配比情况。因此，我们给出如下判断法则：

对第一道工序的 CNC 单次加工时间 tt_1 ，加工第二道工序的 CNC 单次加工时间 tt_2 ，有比值：

$$TT = \frac{tt_1}{tt_2} \quad (21)$$

3 种配比比例共有 7 个比值，分别为 3/5、1、5/3，将比值记作 Tt 。

寻找与加工时间比值最接近的配比，相应公式如下：

$$\min |TT - Tt| \quad (22)$$

将该最小值对应的配比 Tt 确定为当前参数对应选择的 CNC 配比。然而我们必须考虑到，配比的分子分母越接近，系统的周转能力会更加灵活，不排除比分子分母更接近的配比会产生更优方案的可能。因此，做如下规定：

当配比 Tt 确定时，将参数分别带入 Tt 配比与分子分母比其更接近的配比中求解最优结果，取这其中的最优结果作为最终结果。例如，求得 Tt 为 3/5，则取 3/5、4/4、5/3 这 3 中配比情况的模型分别求解，取其中最优解的配比作为最终方案。

利用上述法则，可以保证给出不同的加工时间参数时，可以选择更合理的初始 CNC 配比设置，使得在该配比下得到的最佳调度方案优于其它配比下的方案。

2) 4:4 型的调度方案

Step1: 首先需确定不同刀具类型的 CNC 应摆放的位置。考虑与一道工序中相似的问题，当 RGV 处于某一位置并完成对该位置一台第一道工序类型 CNC 的下料时，需要寻找第二道工序类型的 CNC 进行上料；此时，最佳的选择是该位置的另一台 CNC。因此，我们得出将两类 CNC 分别置于轨道两次为最佳的分配方式。

由一道工序的加工系统，我们已知轨道两侧的 CNC 上下料时间不同，因此具体哪一类 CNC 放置在哪一侧，需要根据给出的加工时间比值 TT ，而具体目标为尽量减少 RGV 的等待时间。综上给出判定方式如下：

$$\begin{cases} TT \geq 1, \text{第一道工序的CNC 放置上下料时间长的一侧} \\ TT < 1, \text{第二道工序的CNC 放置上下料时间长的一侧} \end{cases} \quad (23)$$

Step2: 考虑初始情况。RGV 从位置 1 出发，由于位置 1-4 各有个加工第一道工序的 CNC，故初始仅需要对 4 台 CNC 进行上料，则上料的顺序共有以下 6 种情况：

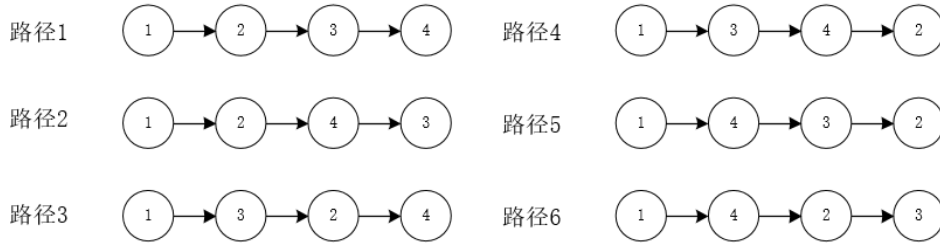


图 6 初始上料的顺序图

故模型将基于 6 种初始状态给出 6 种最佳调度方案，取 6 种方案中的最优作为结果。

Step3: 对于每一块物料，经历的加工流程如图所示：

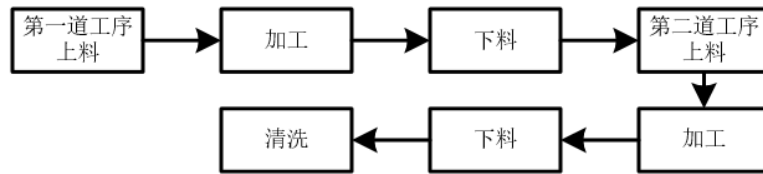


图 7 物料的加工流程图

对任意一种初始状态，RGV 下一步开始对目前 CNC 上的 4 块物料进行操作，直到 4 块物料都完成如图所示的流程，RGV 会开始对其它物料进行作业^[4]。因此，我们将每 4 块物料绑定为一组；对第一组，从初始放置结束为起始，到第 4 块物料完成清洗为结束；对接下来的第 x 组，以 $x-1$ 组结束为起始，第 $4x$ 个物料清洗完成为结束；循环这个流程，直到 8 小时工作结束。

Step4: 对于每一组而言，第 1 块物料完成第一道工序后有 4 个可选择的第二道工序 CNC 进行放置；第 2 块物料完成第一道工序后，有 3 个可选择的第二道工序 CNC 进行放置。以此类推，得到总的选择数为 $4 \times 3 \times 2 \times 1 = 24$ 。

同样，对于每一组而言，在 4 块进行第一道工序的物料中选择第一块加入第二道工序的物料，共有 4 个可能的选择；选择第二块加入第二道工序的物料，共有 3 个可能的选择。依次类推，得到总的可选择数为 $4 \times 3 \times 2 \times 1 = 24$ 。

因此，得到每一组 4 块物料加工过程的可选调度方案数为 $24 \times 24 = 576$ 。

设第 x 组的开始时间点为 f_{1x} ，结束时间为点 f_{2x} ，则对每一组的 576 种可选方案，取最小时间的方案为该组最终调度方案。公式表示为：

$$\min f_0 = f_{2x} - f_{1x} \quad (24)$$

其中， f_0 表示完成一组调度需要的总时间。

Step5: 考虑最后一组的情况。当 RGV 完成对最后一组第 m 块物料的清洗后返回初始位置时，整个流程时间恰好已超出 28800 秒，则放弃对第 m 块物料的加工，直接返回

初始位置。设最后一组加工完成的物料数为 d ，则：

$$d \in \{1, 2, 3, 4\} \quad (25)$$

Step6: 确定目标函数。设每种初始状态对应的最大加工物料数分别为 D_1 、 D_2 、 D_3 、 D_4 、 D_5 、 D_6 。则对任意一种初始状态，最大加工物料数表示为：

$$D_v = 4n + d \quad (26)$$

其中， n 为 28800 秒内完整完成的组数。

故目标函数表示为：

$$\max \{D_1, D_2, D_3, D_4, D_5, D_6\} \quad (27)$$

Step7: 根据 RGV 抓取生料的顺序，将第 k 个被抓取的生料记为物料 k 。记录每个 k ，并记录每个 k 对应的第一道工序上料开始时间点 u_{3k} ，下料开始时间点 u_{4k} ；第二道工序上料开始时间点 u_{5k} ，下料开始时间点 u_{6k} 。

3) 3:5 型的调度方案

Step1: 首先需确定不同刀具类型的 CNC 应摆放的位置。

第一，根据之前同一位置优先的原则，尽量使不同类型的 CNC 在同一 RGV 位置点的两侧。由此确定 3 个加工第一道工序的 CNC 应当不处于同一位置的两侧。

第二，由一道工序的加工系统，我们已知轨道两侧的 CNC 上下料时间不同，因此具体哪一类 CNC 放置在哪一侧，需要根据给出的加工时间比值 TT ，而具体目标为尽量减少 RGV 的等待时间。综上给出判定方式如下：

$$\begin{cases} TT \geq 1, & \text{第一道工序的 CNC 放置上下料时间长的一侧} \\ TT < 1, & \text{第二道工序的 CNC 放置上下料时间长的一侧} \end{cases} \quad (28)$$

这里需要注意，由于第一道工序的 CNC 仅有 3 台，因此第一道工序 CNC 的一侧必定会出现 1 台第二道工序的 CNC。

第三，基于前两点，可行的放置情况共有 4 种（假设在轨道哪一侧已经选定），如图所示：

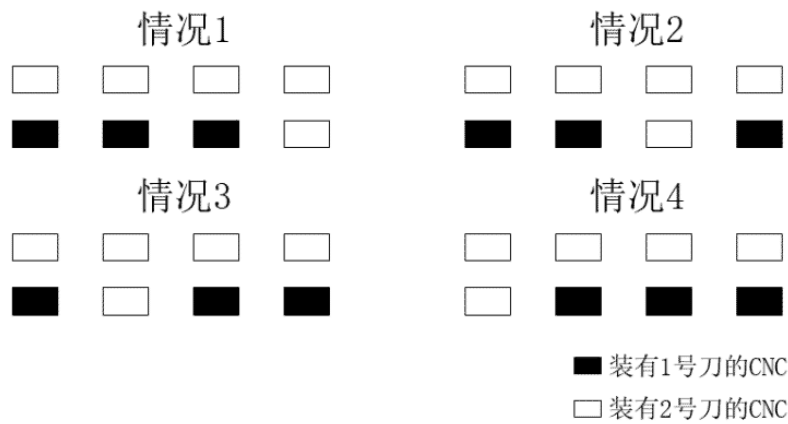


图 8 两种刀具的 CNC 的分布图

接下来从 4 种情况中确定最佳的放置方案。已知 RGV 的初始位置在位置 1，故优先选择在位置 1 放置第一道工序的 CNC；由对称关系，当 RGV 进行工作时，在位置 1、4 的情况近似，在位置 2、3 的情况近似，故在已经选择了位置 1 的情况下，在位置 4 也放置第一道工序的 CNC；同样根据对称性，在 2、3 位置放置的结果会近似相同，不妨

将最后一个 CNC 放置在位置 2。

综上所述，最终选定的放置方案为情况 2。

Step2: 考虑初始情况。RGV 从位置 1 出发，由于位置 1、2、4 各有个加工第一道工序的 CNC，故初始仅需要对 3 台 CNC 进行上料，上料的顺序共有以下 2 种情况：



图 9 上料顺序图

故模型将基于 2 种初始状态给出 2 种最佳调度方案，取 2 种方案中的最优作为结果。

Step3: 对任意一种初始状态，RGV 下一步开始对目前 CNC 上的 3 块物料进行操作，直到 3 块物料都完成先前给出的加工流程，RGV 会开始对其它物料进行作业。因此，我们将每 3 块物料绑定为一组；对第一组，从初始放置结束为起始，到第 3 块物料完成清洗为结束；对接下来的第 x 组，以 $x-1$ 组结束为起始，第 $3x$ 个物料清洗完成为结束；循环这个流程，直到 8 小时工作结束。

Step4: 对于每一组而言，第 1 块物料完成第一道工序后有 5 个可选择的第二道工序 CNC 进行放置；第 2 块物料完成第一道工序后，有 4 个可选择的第二道工序 CNC 进行放置。以此类推，得到总的选择数为 $5 \times 4 \times 3 = 60$ 。

同样，对于每一组而言，在 3 块进行第一道工序的物料中选择第一块加入第二道工序的物料，共有 3 个可能的选择；选择第二块加入第二道工序的物料，共有 2 个可能的选择。依次类推，得到总的可选择数为 6。

因此，得到每一组 3 块物料加工过程的可选调度方案数为 $60 \times 6 = 360$ 。

设第 x 组的开始时间点为 f_{1x} ，结束时间为点 f_{2x} ，则对每一组的 24×24 种可选方案，取最小时间的方案为该组最终调度方案。公式表示为：

$$\min f_0 = f_{2x} - f_{1x} \quad (29)$$

其中， f_0 表示完成一组调度需要的总时间。

Step5: 考虑最后一组的情况。当 RGV 完成对最后一组第 m 块物料的清洗后返回初始位置时，整个流程时间恰好已超出 28800 秒，则放弃对第 m 块物料的加工，直接返回初始位置。设最后一组加工完成的物料数为 d ，则：

$$d \in \{1, 2, 3\} \quad (30)$$

Step6: 确定目标函数。设每种初始状态对应的最大加工物料数分别为 D_1 、 D_2 。则对任意一种初始状态，最大加工物料数表示为：

$$D_v = 3n + d \quad (31)$$

其中， n 为 28800 秒内完整完成的组数。

故目标函数表示为：

$$\max \{D_1, D_2\} \quad (32)$$

Step7: 根据 RGV 抓取生料的顺序，将第 k 个被抓取的生料记为物料 k 。记录每个 k ，并记录每个 k 对应的第一道工序上料开始时间点 u_{3k} ，下料开始时间点 u_{4k} ；第二道工序上料开始时间点 u_{5k} ，下料开始时间点 u_{6k} 。

4) 5:3 型的调度方案

5:3 型的调度方案与 3:5 型有许多相似之处，因此接下来只对改变的步骤进行解释。

Step1 的改变：根据同一位置有效的法则，5:3 型固然先将 4 个第一道工序的 CNC 放置在同侧；在 3:5 型的放置方案中，第一道工序的 CNC 位于位置 1、2、4，为了不影响更优的位置，将第 5 个第一道工序的 CNC 放置在位置 3 的另一侧。故 CNC 放置方案确定。

Step2 的改变：考虑初始情况。RGV 从位置 1 出发，由于位置 1、2、3、4 都存在加工第一道工序的 CNC，故初始需要对 5 台 CNC 进行上料，结合同一位置优先的法则，上料的顺序共有以下 6 种情况：

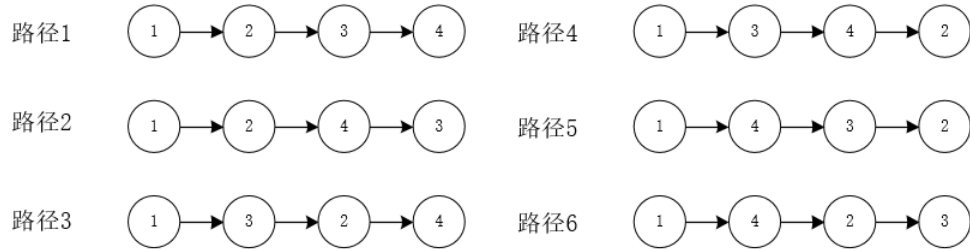


图 10 初始上料顺序图

故模型将基于 6 种初始状态给出 6 种最佳调度方案，取 6 种方案中的最优作为结果。

Step4 的改变：对于每一组而言，第 1 块物料完成第一道工序后有 3 个可选择的第二道工序 CNC 进行放置；第 2 块物料完成第一道工序后，有 2 个可选择的第二道工序 CNC 进行放置。以此类推，得到总的选择数为 6。

同样，对于每一组而言，在 3 块进行第一道工序的物料中选择第一块加入第二道工序的物料，共有 5 个可能的选择；选择第二块加入第二道工序的物料，共有 4 个可能的选择。依次类推，得到总的可选择数为 60。

因此，得到每一组 3 块物料加工过程的可选调度方案数为 $60 \times 6 = 360$ 。

Step6 的改变：由于初始状态有 6 种，故目标函数表示变为：

$$\max \{D_1, D_2, D_3, D_4, D_5, D_6\} \quad (33)$$

● 更换刀具

考虑更换刀具的情况，首先明确换刀的目的。当可以换刀时，不再有 CNC 的的配比放置方案，一切以提高效率为准。

Step1：显然，初始的最佳状态为 8 台 CNC 都进行第一道工序。

Step2：当 RGV 为物料更换工序时，利用一道工序的调度模型选取下一台 CNC 执行第二道工序。此时对该 CNC 进行刀具更换工作，并拾起该 CNC 先前加工好第一道工序的物料，放置前一个需要加工第二道工序的物料。

Step3：重复 Step2 的步骤，直到完成全部的调度。

这里我们不难发现，为了提高效率而制定的上述调度方案中换刀是极其频繁的，每执行一次操作时都大概率进行换刀，具体表现为满足两个法则：

1. 若前一个完工的 CNC 执行了第一道工序，则 RGV 下一个联动的 CNC 需更换为第二道工序的刀具。
2. 若前一个完工的 CNC 执行了第二道工序，则 RGV 下一个联动的 CNC 需更换为第一道工序的刀具。

在实际中换刀的时间大约为 2–5s，因此不会对加工效率产生明显影响。但是另有 3 点我们必须考虑到：

1. 加工系统是否容易发生故障取决于频繁的操作，如果为了提高效率盲目进行上

述频繁换刀的操作，会使得系统发生故障的概率大大提升，得不偿失。

2. 且在实际情况中，不会出现为提高效率而频繁换刀的操作。

3. 初步对频繁换刀的理想状态进行估计，相比不换刀的最佳方案，28800 秒内增加的物料加工数量均在 5 个左右，效率提高并不明显。

上述 3 个问题直接从多方面否定了在调度中频繁更换刀具的不合理性。因此，我们在确定后续的调度方案中，不考虑 CNC 更换刀具的情况，直接将其排除。

5.2.1.4 调度模型的求解算法

对于情况二，所有的求解步骤都已经在模型中一一给出；因此只需要依据模型，将遍历搜索算法与直接计算方法结合，利用 MATLAB 依次得到模型需要的结果，即可完成对情况二的求解。

5.2.2 任务 2

5.2.2.1 两道工序加工模型的结果

首先考虑不换刀的情况，将题目给出的三组参数分别带入模型，利用 MATLAB 编程求解。经过配比检验，得到第一组参数下确定使用 4:4 的放置方案；第二、三组参数需要分别带入 4:4、3:5、5:3 的模型中求解，第二组 4:4 得到成品 209 个成品、3:5 得到 200 个成品、5:3 得到 159 个成品；第三组 4:4 得到成品 232 个成品、3:5 得到 174 个成品、5:3 得到 236 个成品（三种配比情况的结果表详见附录二十六），取其中最优的放置方案。从得到的结果发现：第一、二组适用 4:4，第三组适用 5:3。（5 个一号刀，3 个二号刀）其中，两道工序加工的第一组结果如下(代码文件 Ques201.m 详见附录十二)：

表 5 两道工序调度模型第一组数据结果表

加工物料序号	工序 1 的 CNC 编号	上料开始时间	下料开始时间	工序 2 的 CNC 编号	上料开始时间	下料开始时间
1	1	0	428	2	456	884
2	3	48	507	4	535	988
3	5	96	586	6	614	1092
.....						
251	5	27584	28026	6	28054	28496
252	7	27688	28130	8	28158	28600
253	1	27818	28260	2	28288	28730

两道工序加工第二组结果如下(代码文件 Ques201.m 详见附录十二)：

表 6 两道工序调度模型第二组数据结果表

加工物料序号	工序 1 的 CNC 编号	上料开始时间	下料开始时间	工序 2 的 CNC 编号	上料开始时间	下料开始时间
1	1	0	310	2	340	875
2	3	53	398	4	428	993
3	5	106	486	6	516	1111
.....						
207	3	27273	27808	6	27861	28396
208	7	27414	27949	8	27979	28514
209	5	27532	28067	2	28160	28695

两道工序加工第三组结果如下(代码文件 Ques203.m 详见附录十四):

表 7 两道工序调度模型第三组数据结果表

加工物料序号	工序 1 的 CNC 编号	上料开始时间	下料开始时间	工序 2 的 CNC 编号	上料开始时间	下料开始时间
1	4	18	505	3	537	957
2	8	82	596	7	628	837
3	5	132	673	1	732	1059
.....						
234	5	27582	28193	3	28238	28474
235	4	27679	28442	3	28474	28683
236	6	27813	28308	7	28358	28567

结果表明, 第 1、2 组参数均在 4:4 放置 CNC 的情况下得到更好的解, 第 3 组参数在 5:3 放置 CNC 的情况下得到更好的解, 这也验证了之前判断配比的法则是有必要且正确的。

5.2.2.2 两道工序模型的实用性

从结果来看, 模型针对 3 组参数的求解情况均符合实际, 既没有出现异常的上下料时间、也没有异常的两个相邻编号的物料上下料开始时间间隔。而这 3 组数据各有特点, 反应了不同工序的不同耗时配比, 故 3 组结果均合理足以证明模型具有较强的实用性, 可针对任何符合实际的一组参数进行求解, 得到合理的最佳效率方案。

分析最优的结果序列, 虽然不全为 1-8 的自然序列, 但都会陷入一个循环路径。这与实际的加工过程中, RGV 的运动通常存在某种规律、不会做无规则的复杂运动相符合, 进一步证明了模型的实用性。

5.2.2.3 两道工序模型系统的作业效率

理想情况下, 不考虑 RGV 移动所消耗的时间, 则对于之前已经验证出一、二两组的 4:4 类型的 CNC 分布。作业效率如下:

$$\begin{aligned} e_{1i} &= tt_1 + s_1 \\ e_{2i} &= tt_2 + s_2 \\ e_i &= \max\{e_{1i}, e_{2i}\} \end{aligned} \quad (34)$$

tt_i 表示加工一个第 i 道工序的时间, e_{1i} 表示加工第一道工序的时间与上下料时间之和, e_{2i} 表示加工第二道工序的时间与上下料时间之和。

$$N_{2i} = \frac{(28800 - t_{0i})}{e_i} \times 4 \quad (35)$$

N_{2i} 表示理想情况下, 8 小时能加工两道工序的第 i 组的成品数。 t_{0i} 表示第一次上下料所消耗的时间。得到 $N_{21} = 268$, $N_{22} = 216$

$$\eta_{2i} = \frac{n_{2i}}{N_{2i}} \times 100\% \quad (36)$$

η_{2i} 表示两道工序情况下, 利用该模型系统的作业效率。

得到: $\eta_{21} = 94.40\%$, $\eta_{22} = 96.76\%$ 。

第三组是适用于 5:3 刀具类型的 CNC 分布, 如果想向之前那样具体算出理想情况下 8 小时能生产的成品数, 比较繁琐。那么, 仍以 4:4 刀具分布的 CNC 的理想情况作为比较基准, 先得到 $N_{23} = 236$ 再求得 $\eta_{23} = 100\%$ 。

5.3 随机故障模型的建立与求解

5.3.1 任务 1

对于情况三而言,首先需要给出加工系统的故障概率模型,再将模型分别嵌入情况一、二的模型中,分别考虑其影响。

5.3.1.1 概率故障模型

已知 CNC 在加工过程中发生故障的概率为1%,即每台 CNC 在每一次加工时,都有1% 的概率的出现故障。故给出第*i*台 CNC 的故障状态如下:

$$C_i = \begin{cases} 0, & \text{CNC 正常} \\ 1, & \text{CNC 故障} \end{cases} \quad (37)$$

其中,具体每次加工中, C_i 状态随机选取,有1%的概率为0,99%的概率为1。

当 CNC 发生故障时,其故障时长可能为 10-20 分钟。我们不妨假设故障时长服从均匀分布,再针对情况一、二的模型进行嵌入,分别固定带入这 3 个故障市场对结果进行求解^[5]。

5.3.1.2 一道工序的故障模型

Step1: 将概率故障模型嵌入一道工序的“八步看一步”的调度模型,从初始放置步骤开始进行调度操作。

Step2: 假设在随机的过程中,第*k*个物料在第*i*台 CNC 上加工时发生了故障;此时需要明确,故障的修复是人工操作,RGV 无法预知修复完成的时间。因此,从 CNC 发生故障的时刻开始,在 RGV 的视野中第*i*台 CNC 消失,系统中仅仅存在 7 台 CNC。

Step3: 在上述情况下进行一道工序的调度方案,以未来 8 步决策 1 步作业,直到 CNC 修复完成,重新回归到 RGV 的视野中^[6]。设第*w*步的时间区间为 $[w_1, w_2]$,此处需要进行考虑,当 CNC 发生故障时,RGV 正处于第*w*步的时间区间中,则 RGV 需将第*w*步完成,在考虑下一步时忽略故障 CNC。

Step4: 故障在一次加工过程中发生时,以秒为单位服从均匀分布,即随机在加工时间里的任意一秒开始出现故障;故障发生的时长也服从均匀分布,故随机选取 10-20 分钟内的任意一个加工时长。因此,对于发生故障的 CNC 正在加工的物料*k*,记录*k*及*k*的故障开始时间、结束时间。

Step5: 故障排除后恢复原先的调度过程,直到下一个随机故障出现,重复上述步骤,一直到 28800 秒结束,记录一道工序调度模型需要得到的结果。

5.3.1.3 两道工序的故障模型

两道工序的故障模型首先因基于情况二的完整过程。对于情况二,我们已经通过三组数据验证了当不更换刀具时,4:4 型为解决一般问题的最佳 CNC 配比方案,因此我们只对 4:4 型的调度方案嵌入概率故障模型,并给出其结果。

由于两道工序的故障模型与一道工序的故障模型区别仅仅存在于 Step3,故直接延用一道工序的故障模型,并对 Step 作出改变。

Step1: 将概率故障模型嵌入两道工序的调度模型,从初始放置步骤开始进行调度操作。

Step2: 假设在随机的过程中,第*k*个物料在第*i*台 CNC 上加工时发生了故障;此时需要明确,故障的修复是人工操作,RGV 无法预知修复完成的时间。因此,从 CNC 发生故障的时刻开始,在 RGV 的视野中第*i*台 CNC 消失,系统中仅仅存在 7 台 CNC。

Step3: 两道工序的调度方案是以 4 块物料完成加工为一组进行分步调度。设对第*k*块物料进行某道工序的加工过程中出现故障。设此时系统处于第*x*组物料的加工过程中,故先将该组的加工流程继续完成,跳过故障 CNC 对应的步骤;可以得出,由于故障原

因，此过程可能出现多余的往返。在进行下一组物料的加工时，由于某一类型的 CNC 有 1 个缺失，故可能的选择情况由 24×24 转变为 144。因此，从故障开始时，每一组由 144 种方案中选取最优的调度方案进行；但需要注意，这一组加工的物料数可能为 4，也可能为 3，这取决于出现故障的 CNC 类型。当故障被修复时，同理令系统完成当前一组的加工，从下一组加工开始恢复原先的调度方案。

Step4: 故障在一次加工过程中发生时，以秒为单位服从均匀分布，即随机在加工时间里的任意一秒开始出现故障；故障发生的时长也服从均匀分布，故随机选取 10-20 分钟内的任意一个加工时长。因此，对于发生故障的 CNC 正在加工的物料 k ，记录 k 及 k 的故障开始时间、结束时间。

Step5: 故障排除后恢复原先的调度过程，直到下一个随机故障出现，重复上述步骤，一直到 28800 秒结束，记录一道工序调度模型需要得到的结果。

Step6: 对不同可选 CNC 配比放置下的初始情况分别重复上述步骤，比较不同配比下故障对系统的影响。

5.3.1.4 模型的求解算法

基于原先的求解算法，更改相关的函数、参数，依照概率合理利用随机生成，对模型步骤依次进行编程求解，最终可以得到模型需要的相关结果。

5.3.2 任务 2

5.3.2.1 故障模型的结果

利用 MATLAB 求解一道工序随机故障模型(代码文件 Ques301.m 详见附录十六)。一道工序的情况下，第一组随机故障模型结果如下：

表 8 一道工序模型第一组随机故障模型结果表

加工物料序号	加工 CNC 编号	上料开始时间	下料开始时间	故障时的物料序号	故障 CNC 编号	故障开始时间	故障结束时间
1	1	0	588	66	2	5046	5647
2	2	28	641	213	6	16048	16968
.....				244	6	18797	19869
365	1	28049	28693	351	3	27095	28190

一道工序的情况下，第二组随机故障模型：

表 9 一道工序模型第一组随机故障模型结果表

加工物料序号	加工 CNC 编号	上料开始时间	下料开始时间	故障时的物料序号	故障 CNC 编号	故障开始时间	故障结束时间
1	1	0	610	78	6	6443	7413
2	2	30	670	201	2	16240	16886
.....				251	5	20715	21522
339	7	28067	28677	-	-	-	-

一道工序的情况下，第三组随机故障模型结果：

表 10 一道工序模型第一组随机故障模型结果表

加工物料序号	加工 CNC 编号	上料开始时间	下料开始时间	故障时的物料序号	故障 CNC 编号	故障开始时间	故障结束时间
1	1	0	572	65	1	4610	5308

2	2	27	624	138	1	10030	10890
			167	8	12304	13262
389	7	28171	28748	-	-	-	-

利用 MATLAB 求解两道工序随机故障模型(代码文件 Ques302.m 详见附录十七)。
两道工序的情况下，第二组随机故障模型结果如下：

表 11 两道工序模型第一组随机故障模型结果表

加工物 料序号	工序 1 的 CNC 编号	上料开 始时间	下料开 始时间	工序 2 的 CNC 序号	上料开 始时间	下料开 始时间
1	1	0	428	2	456	865
2	3	48	507	4	535	944
3	5	96	586	6	614	1023
					
238	5	27026	27468	2	27671	28077
239	3	27130	27747	4	28149	28558
240	7	27364	28238	8	28266	28675

表 12 两道工序模型第一组随机故障模型故障故障信息表

故障时的物料 序号	故障 CNC 编号	故障开 始时间	故障结束 时间
96	7	10489	11243
199	4	22782	23772
235	6	27513	28149

两道工序的情况下，第一组随机故障模型结果：

表 13 两道工序模型第二组随机故障模型结果表

加工物 料序号	工序 1 的 CNC 编号	上料开 始时间	下料开 始时间	工序 2 的 CNC 序号	上料开 始时间	下料开 始时间
1	1	0	428	2	456	865
2	3	48	507	4	535	944
3	5	96	586	6	614	1023
					
238	5	27026	27468	2	27671	28077
239	3	27130	27747	4	28149	28558
240	7	27364	28238	8	28266	28675

表 14 两道工序模型第二组随机故障模型故障信息表

故障时的物料 序号	故障 CNC 编号	故障开 始时间	故障结束 时间
41	3	5360	6192
119	2	16831	17984
124	6	18107	18768

两道工序的情况下，第三组随机故障模型结果：

表 15 两道工序模型第三组随机故障模型结果表

加工物 料序号	工序 1 的 CNC 编号	上料开 始时间	下料开 始时间	工序 2 的 CNC 序号	上料开 始时间	下料开 始时间
1	1	0	310	2	340	875
2	3	53	398	4	428	963
3	5	106	486	6	516	1051
.....						
198	1	27235	27770	2	27822	28357
199	7	27411	27946	8	27981	28516
200	5	27534	28069	6	28099	28634

表 16 两道工序模型第三组随机故障模型故障信息表

故障时的物料 序号	故障 CNC 编 号	故障开 始时间	故障结束 时间
140	8	17259	18006
198	7	24004	25088

5.3.2.2 故障模型的实用性

关于故障模型，我们需要通过模型的稳定性判断模型是否实用，即判断故障对模型产生的影响程度是否在某一限度内。首先依次选择加工物料总数、故障物料总数为样本。分别对随机生成修复故障时间的样本、设定 10 分钟修复故障时间的样本、设定 15 分钟的样本及设定 20 分钟的样本做平均值与标准差，标准差计算公式表示为：

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (u - x_i)^2}{n-1}} \quad (38)$$

利用 MATLAB 分别求解 4 种情况下 20 个样本的标准差。

对一道工序，得到的结果如下(代码文件 Ques303.m 详见附录二十)：

表 17 随机故障模型对一道工序 20 次样本分析结果表

		成品平 均数	成品数 标准差	故障发生 平均次数	故障数 标准差
随机	第一组	364.6	12.20	4.20	1.881
	第二组	330.6	15.57	3.50	1.638
	第三组	377.2	10.03	3.50	1.573
10min	第一组	365.0	9.55	4.25	1.860
	第二组	326.0	18.68	4.00	1.806
	第三组	376.1	13.41	4.20	2.726
15min	第一组	366.0	11.95	3.85	1.927
	第二组	327.7	19.55	3.60	1.759

	第三组	378.0	8.92	3.60	1.536
20min	第一组	362.1	9.82	4.15	1.755
	第二组	326.3	14.68	3.75	1.517
	第三组	366.6	13.73	4.45	1.572

对两道工序情况(针对 4:4 情况),结果如下(代码文件 Ques304.m 详见附录二十一):

表 18 随机故障模型对两道工序 20 次样本分析结果表

		成品平 均数	成品数 标准差	故障发生 平均次数	故障数 标准差
随机	第一组	240.8	12.20	2.30	1.881
	第二组	196.7	7.85	3.20	1.824
	第三组	221.9	7.67	3.15	1.755
10min	第一组	241.3	9.55	3.05	1.860
	第二组	202.6	3.39	2.65	1.309
	第三组	226.5	4.58	2.95	1.669
15min	第一组	242.6	11.95	2.35	1.927
	第二组	201.2	5.87	2.25	1.446
	第三组	222.1	7.72	2.80	1.765
20min	第一组	233.6	9.82	2.85	1.755
	第二组	196.4	9.69	2.70	1.689
	第三组	218.0	10.13	3.00	1.589

由此可见当样本修复时间在 10~15min 左右时,模型的稳定性更强,这表示在今后的加工中,可设法人为将修复故障时间控制在 10~15min 左右,避免出现故障严重影响加工的情况。

六、模型的灵敏度分析

对题目给定的 3 组数据,令 RGV 移动时间、CNC 加工时间、上下料时间、清洗时间均上下连续波动 5%,利用 MATLAB 绘制 4 个参数分别与物料成品数量之间的函数关系图。一道工序情况下的模型灵敏度分析图如下(代码文件 LingMin1.m 详见附录二十四):

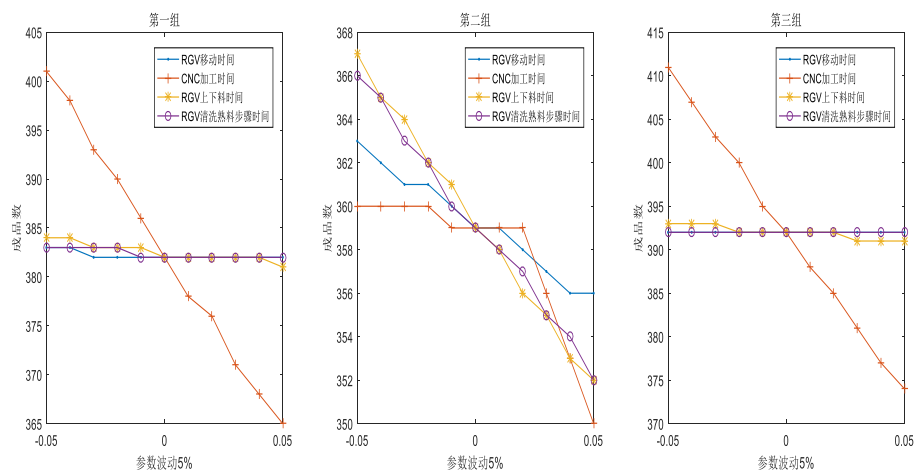


图 11 一道工序模型的灵敏度分析图

如图所示，对于第一、三组数据而言，两道工序加工的调度模型对 CNC 加工时间较为敏感，而对第二组数据而言，RGV 移动时间、CNC 加工时间、上下料时间、清洗时间的变化都会对最终能完成的成品数产生较大影响。因此，可以推断得出，一般情况下，模型对 CNC 加工时间都较为敏感，在实际加工过程中，需要谨慎考虑加工时间的设置，以保证加工效率。

绘制两道工序情况下的模型灵敏度分析图如下(代码文件 LingMin2.m 详见附录二十五):

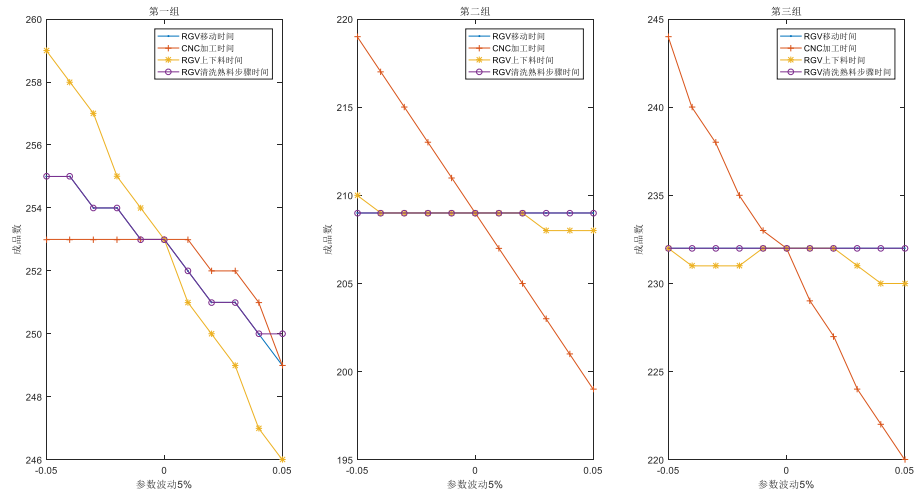


图 12 两道工序模型的灵敏度分析图

观察上图结果，对于第二、三组数据而言，一道工序加工的调度模型对 CNC 加工时间较为敏感，而对第一组数据而言，RGV 移动时间、CNC 加工时间、上下料时间、清洗时间的变化都会对最终能完成的成品数产生较大影响。因此，可以推断得出，一般情况下，模型对 CNC 加工时间都较为敏感，在实际加工过程中，需要谨慎考虑加工时间的设置，以保证加工效率。

上述两种情况表明，无论采用几道加工工序，CNC 单次加工时间的设定最为关键，而其它三个参数会因加工工序的数量不同而呈现不同的灵敏度，有些情况下敏感，有些情况下稳定。

七、模型的综合评价与推广

7.1 模型的综合评价

7.1.1 模型的优点

本文在考虑未来步骤对当前步骤的影响时，从多方面进行分析，确定了由未来几步确定当前一步可以时结果无限接近全局最优；既没有进行复杂的全局搜索，也没有进行不确定性较强的随机生成，但却得到了理想的最优结果。

本文在考虑两道工序的加工过程时，针对不同的 CNC 布局进行了全面的分析与验证，最终给出了不同情形下最佳的布局方式，增强了模型的实用性。

本文突出的特点在于考虑了更换刀具的问题。在实际问题中，更换刀具是常见的操作；针对题目，频繁的更换刀具确实可以使问题得到更优的结果；最终要的一点是，合理的换刀并没有违背题目的任何要求。基于上述三点，更换刀具对于两道工序的加工过程是有意义的，值得推广的。

本文在考虑随机生成故障时，对 RGV 的指令进行了分析，确定了故障出现时 RGV

是否应继续当前指令以及故障期间 RGV 指令发生了怎样的改变。这使得本文对故障的考虑更加实际、更加精细。

本文在检验模型的步骤中,使用方差来判断模型的稳定性,使用理想状态与实际状态的对比情况来判断模型的效率,进一步增强了模型的可靠性。

7.1.2 模型的缺点

本文在考虑不同配比的 CNC 时,忽略了 2:6、1:7 的情况;但是在实际的操作中,当两种 CNC 的加工时间差超过一定限度时,2:6、1:7 的配比将成为最优的放置方案。此处模型直接忽略了少数情况,考虑欠妥当。

本文在模型假设时假设了传送带的总能及时给到生料与熟料空位,但在实际情况中,传送带的运行是有一定的速度的,在加工系统运行时传送带会影响整个调度方案;模型的假设与实际略不相符,故存在缺陷。

7.2 模型的推广

本文主要利用动态规划对智能加工系统制定了高效的调度方案,其中运用最多的“未来多步确定一步”的思想可以推广到更多的调度模型中;对于问题本身,此模型在 RGV 位置增多, CNC 数量增多的情况下依然使用,可能仅仅需要对“未来多步”进行略微的调整。但是 RGV 的位置与 CNC 的数量不能无限的增加下去,当其达到一定值时, RGV 会应运动时间过长而效率降低,此时就需要在原先模型的基础上增加 RGV 数量,得到更高效的调度方案。

八、参考文献

- [1]查振元,李计星,绳润涛,等.智能平移轨道导引车的应用[J].机器人技术与应用,2017(5):42-43.
- [2]顾红,邹平,徐伟华.环行穿梭车优化调度问题的自学习算法[J].系统工程理论与实践,2013,33(12):3223-3230.
- [3]陈华.基于分区法的 2-RGV 调度问题的模型和算法[J].工业工程与管理,2014,19(6):70-77.
- [4]金芳,方凯,王京林,基于排队论的 AGV 调度研究[J].仪器仪表学报,2004,25(s1):844-846.
- [5]周广文,贾亚洲.专用计算机数控机床故障率实验研究[J].组合机床与自动化加工技术,2005(1):45-47.
- [6]乔非,吴启迪.SJ—FMS 中 RGV 的实时调度与故障调度[J].组合机床与自动化加工技术,1995(03):39-43.

九、附录

附录程序运行环境:

MATLAB 版本: 9.0.0.341360 (R2016a)

MATLAB 许可证编号: 123456

操作系统: Microsoft Windows 10 家庭中文版 Version 10.0 (Build 17134)

Java 版本: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode

附录一: oneStep.m

```
function [h,f,g,z] = oneStep( a,dT,T,t0,c )
```

```
%单步最优
```

```

z=[c;c;c;c];
for i=1:4
    e(i)=dT(abs(a-i)+1);%可能移动时间
    f(i)=i;%位置
    g(i)=t0+e(i);%移动后当前时间

    if g(i)>c(2*(i-1)+1)&&g(i)>c(2*(i-1)+2)
        %[cmin,cind]=min([c(2*(i-1)+1)+28+T(7),c(2*(i-1)+2)+31+T(7)]);
        cind=1;
        z(i,2*(i-1)+cind)=g(i)+T(5)+T(4);
        h(i)=0;%其余 CNC 等待时间
        for j=1:8
            if c(j)<g(i)+T(5)+T(7)&&j~=2*(i-1)+1
                h(i)=h(i)+g(i)+T(5)+T(7)-c(j);
            end
        end
    else
        [cmin,cind]=min([c(2*(i-1)+1)+T(5)+T(7),c(2*(i-1)+2)+T(6)+T(7)]);
        z(i,2*(i-1)+cind)=z(i,2*(i-1)+cind)+T(4+cind)+T(4);
        h(i)=0;%其余 CNC 等待时间
        for j=1:8
            if c(j)<cmin&&c(j)~=cmin-T(7)-T(4+cind)
                h(i)=h(i)+cmin-c(j);
            end
        end
    end
end
end
end

```

附录二： fStep.m

```

function e3 = fStep( f,dT,T,g,h,e )
%多步最优
%实践解得发现瞻望四步迈一步已可解得全局最优解
e1=[];f1=[];g1=[];h1=[];
for i=1:4
    [ze1,zf1,zg1,zh1] = oneStep( f(i),dT,T,g(i),h(i,:) );
    e1=[e1;ze1];f1=[f1;zf1];g1=[g1;zg1];h1=[h1;zh1];
    clear ze1 zf1 zg1 zh1
end
e2=[];f2=[];g2=[];h2=[];
for i=1:4
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f1(i,j),dT,T,g1(i,j),h1(4*(i-1)+j,:) );
        e2=[e2;ze1];f2=[f2;zf1];g2=[g2;zg1];h2=[h2;zh1];
    end
end

```

```

        clear ze1 zf1 zg1 zh1
    end
end
e3=[];f3=[];g3=[];h3=[];
for i=1:16
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f2(i,j),dT,T,g2(i,j),h2(4*(i-1)+j,:));
        e3=[e3;ze1];f3=[f3;zf1];g3=[g3;zg1];h3=[h3;zh1];
        clear ze1 zf1 zg1 zh1
    end
end

for i=1:4
    e1(i,:)=e1(i,:)+e(i);
end
for i=1:4
    for j=1:4
        e2(4*(i-1)+j,:)=e2(4*(i-1)+j,:)+e1(i,j);
    end
end
for i=1:4
    for j=1:4
        for k=1:4
            e3(16*(i-1)+4*(j-1)+k,:)=e3(16*(i-1)+4*(j-1)+k,:)+e2(4*(i-1)+j,k);
        end
    end
end
end
end

```

附录三： xunhuan.m

```
function [N,mm] = xunhuan( a,T,lj,t0,b,c )
```

```
%给迭代迈步循环提供函数
```

```
%初始路径移动
```

```
N=8;
```

```
for i=1:4
```

```
    %RGV 每一步移动
```

```
    dT=[0,T(1:3)];
```

```
    d=dT(abs(a-lj(i))+1);
```

```
    a=lj(i);
```

```
    t0=t0+d;
```

```
    %RGV 为单个奇数 CNC 上料
```

```
    M(2*(i-1)+1,1)=t0;
```

```
    t0=t0+T(5);
```

```
    b(2*(lj(i)-1)+1)=1;
```

```

c(2*(lj(i)-1)+1)=t0+T(4);
M(2*(i-1)+1,2)=2*(lj(i)-1)+1;

%RGV 为单个偶数 CNC 上料
M(2*(i-1)+2,1)=t0;
t0=t0+T(6);
b(2*(lj(i)-1)+2)=1;
c(2*(lj(i)-1)+2)=t0+T(4);
M(2*(i-1)+2,2)=2*(lj(i)-1)+2;
end

%多步最优
while t0<=8*3600
    [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
    e3 = fStep( f,dT,T,g,h,e );
    e3 = reshape(e3,64,4);
    [minValue,row] = min(e3);
    [minValue,col] = min(minValue);
    d=dT(abs(a-col)+1);
    t0=t0+d;
    t0
    a=col;
    N=N+1;
    i=a;
    if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
        %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;cind=1;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
        [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
        cmin=cmin-T(4+cind);
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    else
        [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    end
    M(N,1)=cmin;
    M(N,2)=2*(i-1)+cind;
    for i=length(M(:,2))-1:-1:1
        if M(i,2)==2*(a-1)+cind
            M(i,3)=cmin;
            break;
        end
    end
end

```

```
end  
t0=cmin+T(4+cind)+T(7);
```

```
end  
mm=M(N-8,3);  
end
```

附录四： find8.m

```
%找出八步  
clc,clear  
%RGV 移动 1,2,3 个单位所需时间  
T=[20,33,46];  
%CNC 加工完成一个一道工序的物料所需时间  
T=[T,560];  
%RGV 为 CNC1#, 3#, 5#, 7#一次上下料所需时间  
T=[T,28];  
%RGV 为 CNC2#, 4#, 6#, 8#一次上下料所需时间  
T=[T,31];  
%RGV 完成一个物料的清洗作业所需时间  
T=[T,25];
```

```
%RGV 当前时间  
t0=0;  
%RGV 当前位置[1,2,3,4]  
a=1;
```

```
%CNC 当前工作状态  
b=zeros(1,8);  
%CNC 预计加工完成时间  
c=zeros(1,8);
```

```
%初始路径选择  
start=[ones(6,1),perms([2,3,4])];  
lj=start(6,:);
```

```
%初始路径移动  
for i=1:4  
    %RGV 每一步移动  
    dT=[0,T(1:3)];  
    d=dT(abs(a-lj(i))+1);  
    a=lj(i);  
    t0=t0+d;
```

```
%RGV 为单个奇数 CNC 上料
```

```

t0=t0+T(5);
b(2*(lj(i)-1)+1)=1;
c(2*(lj(i)-1)+1)=t0+T(4);

%RGV 为单个偶数 CNC 上料
t0=t0+T(6);
b(2*(lj(i)-1)+2)=1;
c(2*(lj(i)-1)+2)=t0+T(4);
end

[e,f,g,h] = oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
e1=[];f1=[];g1=[];h1=[];
for i=1:4
    [ze1,zf1,zg1,zh1] = oneStep( f(i),dT,T,g(i),h(i,:) );
    e1=[e1;ze1];f1=[f1;zf1];g1=[g1;zg1];h1=[h1;zh1];
    clear ze1 zf1 zg1 zh1
end
e2=[];f2=[];g2=[];h2=[];
for i=1:4
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f1(i,j),dT,T,g1(i,j),h1(4*(i-1)+j,:) );
        e2=[e2;ze1];f2=[f2;zf1];g2=[g2;zg1];h2=[h2;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
e3=[];f3=[];g3=[];h3=[];
for i=1:16
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f2(i,j),dT,T,g2(i,j),h2(4*(i-1)+j,:) );
        e3=[e3;ze1];f3=[f3;zf1];g3=[g3;zg1];h3=[h3;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
e4=[];f4=[];g4=[];h4=[];
for i=1:64
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f3(i,j),dT,T,g3(i,j),h3(4*(i-1)+j,:) );
        e4=[e4;ze1];f4=[f4;zf1];g4=[g4;zg1];h4=[h4;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
e5=[];f5=[];g5=[];h5=[];
for i=1:256
    for j=1:4

```

```

        [ze1,zf1,zg1,zh1] = oneStep( f4(i,j),dT,T,g4(i,j),h4(4*(i-1)+j,:) );
        e5=[e5;ze1];f5=[f5;zf1];g5=[g5;zg1];h5=[h5;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
e6=[];f6=[];g6=[];h6=[];
for i=1:1024
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f5(i,j),dT,T,g5(i,j),h5(4*(i-1)+j,:) );
        e6=[e6;ze1];f6=[f6;zf1];g6=[g6;zg1];h6=[h6;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
e7=[];f7=[];g7=[];h7=[];
for i=1:4096
    for j=1:4
        [ze1,zf1,zg1,zh1] = oneStep( f6(i,j),dT,T,g6(i,j),h6(4*(i-1)+j,:) );
        e7=[e7;ze1];f7=[f7;zf1];g7=[g7;zg1];h7=[h7;zh1];
        clear ze1 zf1 zg1 zh1
    end
end
end

```

附录五： Ques1.m

```

%一道工序的三组数据的求解
clc,clear
%RGV 移动 1,2,3 个单位所需时间
T=[23,41,59];
%CNC 加工完成一个一道工序的物料所需时间
T=[T,580];
%RGV 为 CNC1#, 3#, 5#, 7#一次上下料所需时间
T=[T,30];
%RGV 为 CNC2#, 4#, 6#, 8#一次上下料所需时间
T=[T,35];
%RGV 完成一个物料的清洗作业所需时间
T=[T,30];
%1 T=[20,33,46,560,28,31,25];
    T=[23,41,59,580,30,35,30];
%3 T=[18,32,46,545,27,32,25];
%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

%CNC 当前工作状态

```



```

b=zeros(1,8);
%CNC 预计加工完成时间
c=zeros(1,8);

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

for i=1:6
    lj=start(i,:);
    [Mn(i),Ma(i)] = xunhuan( a,T,lj,t0,b,c );
    clear lj
end

%初始路径选择
start=perms([1,2,3,4]);

for i=1:24
    lj=start(i,:);
    [Mn(i),Ma(i)] = xunhuan( a,T,lj,t0,b,c );
    clear lj
end

i=6;clear M
lj=start(i,:);
%初始路径移动
N=8;
for i=1:4
    %RGV 每一步移动
    dT=[0,T(1:3)];
    d=dT(abs(a-lj(i))+1);
    a=lj(i);
    t0=t0+d;

    %RGV 为单个奇数 CNC 上料
    M(2*(i-1)+1,1)=t0;
    t0=t0+T(5);
    b(2*(lj(i)-1)+1)=1;
    c(2*(lj(i)-1)+1)=t0+T(4);
    M(2*(i-1)+1,2)=2*(lj(i)-1)+1;

    %RGV 为单个偶数 CNC 上料
    M(2*(i-1)+2,1)=t0;
    t0=t0+T(6);
    b(2*(lj(i)-1)+2)=1;

```

```

c(2*(lj(i)-1)+2)=t0+T(4);
M(2*(i-1)+2,2)=2*(lj(i)-1)+2;
end

%多步最优
while t0<=8*3600

[e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
e3 = fStep( f,dT,T,g,h,e );
e3 = reshape(e3,64,4);
[minValue,row] = min(e3);
[minValue,col] = min(minValue);
d=dT(abs(a-col)+1);
t0=t0+d;
t0
a=col;
N=N+1;
i=a;
if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
    %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;cind=1;
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
    [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
    cmin=cmin-T(4+cind);
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
else
    [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
end
M(N,1)=cmin;
M(N,2)=2*(i-1)+cind;
for i=length(M(:,2))-1:-1:1
    if M(i,2)==2*(a-1)+cind
        M(i,3)=cmin;
        break;
    end
end
t0=cmin+T(4+cind)+T(7);
end
[m,n]=size(M);
M(m-7:m,:)=[];
M(m-8,3)=M(m-8,1)+T(4+mod(M(m-8,2),2))+T(4);

```

```

if M(m-8,3)+T(4+mod(M(m-8,2),2))+T(7)>8*3600
    M(m-8,:)=[];
end
MM(:,1)=M(:,2);
MM(:,2)=M(:,1);
MM(:,3)=M(:,3);

```

附录六： f201.m

```

function [ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T )
QQ=perms([1,2,3,4]);
T6=10*3600+1;%记录最小值
for i=1:24
    for j=1:24
        T4=QQ(i,:);%前往哪个工序 1
        T5=QQ(j,:);%去往哪里工序 2
        d=dT(abs(T4(1)-T3)+1);%移动
        QT=t0+d;%移动到 T41
        QT=max(QT,T1(T4(1)));%T41 开始下料
        VT1(1,1)=QT;
        QT=QT+T(6);%T41 下料完毕
        d=dT(abs(T4(1)-T5(1))+1);%移动
        QT=QT+d;%移动到 T51
        QT=max(QT,T2(T5(1)));%T51 开始下料
        VT2(1,1)=QT;
        QT=QT+T(7);%T51 下料完毕
        QT=QT+T(8);%T51 清洗完毕

        for k=2:4
            d=dT(abs(T5(k-1)-T4(k))+1);%移动
            QT=QT+d;%移动到 T4k
            QT=max(QT,T1(T4(k)));%T4k 开始下料
            VT1(k,1)=QT;
            QT=QT+T(6);%T4k 下料完毕
            d=dT(abs(T4(k)-T5(k))+1);%移动
            QT=QT+d;%移动到 T5k
            QT=max(QT,T2(T5(k)));%T5k 开始下料
            VT2(k,1)=QT;
            QT=QT+T(7);%T5k 下料完毕
            QT=QT+T(8);%T5k 清洗完毕
        end
    end
    if QT < T6
        T6=QT;
        VTT1=VT1;
        VTT2=VT2;
    end
end

```

```

        TT4=T4;
        TT5=T5;
    end
    clear QT VT1 VT2
end
end
end

附录七: f202.m
function [ m,TTT ] = f202( aa,dT,T,t0 )
aa=[aa;aa];
QQ=perms([1,2,3,4]);
for i=1:24
    d=dT(abs(aa(4,1)-QQ(i,1))+1);
    tQ(i)=t0+d;
    for j=1:4
        if QQ(i,1)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,1))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,2)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,3))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,3)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
end
end

```

```

        end
    end
    d=dT(abs(QQ(i,4)-QQ(i,3))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,4)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
end
end
[tQmin,tQind]=min(tQ);
aa(5:8,1)=QQ(tQind,:);
aa(1:4,5)=QQ(tQind,:);
i=tQind;
d=dT(abs(aa(4,1)-QQ(i,1))+1);
tQ(i)=t0+d;
for j=1:4
    if QQ(i,1)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(5,2)=aa(j,3);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=aa(j,3)+T(6);
            aa(1,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(5,2)=tQ(i);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=tQ(i)+T(6);
            aa(1,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)

```

```

        aa(6,2)=aa(j,3);
        aa(2,3)=aa(6,2);
        aa(6,3)=aa(6,2)+T(6)+T(4);
        aa(2,6)=aa(j,3)+T(6);
        aa(2,7)=aa(j,3)+T(6)+T(7)+T(5);
        tQ(i)=aa(j,3)+T(6)+T(7);
    else
        aa(6,2)=tQ(i);
        aa(2,3)=aa(6,2);
        aa(6,3)=aa(6,2)+T(6)+T(4);
        aa(2,6)=tQ(i)+T(6);
        aa(2,7)=tQ(i)+T(6)+T(7)+T(5);
        tQ(i)=tQ(i)+T(6)+T(7);
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(7,2)=aa(j,3);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=aa(j,3)+T(6);
            aa(3,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(7,2)=tQ(i);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=tQ(i)+T(6);
            aa(3,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(8,2)=aa(j,3);
            aa(4,3)=aa(8,2);

```

```

        aa(8,3)=aa(8,2)+T(6)+T(4);
        aa(4,6)=aa(j,3)+T(6);
        aa(4,7)=aa(j,3)+T(6)+T(7)+T(5);
        tQ(i)=aa(j,3)+T(6)+T(7);
    else
        aa(8,2)=tQ(i);
        aa(4,3)=aa(8,2);
        aa(8,3)=aa(8,2)+T(6)+T(4);
        aa(4,6)=tQ(i)+T(6);
        aa(4,7)=tQ(i)+T(6)+T(7)+T(5);
        tQ(i)=tQ(i)+T(6)+T(7);
    end
end
end
end

while t0<8*3600
    [m,n]=size(aa);
    T1=aa(m-3:m,[1,3]);%一道工序结束时间
    T2=aa(m-7:m-4,[5,7]);%二道工序结束时间
    T1=sortrows(T1,1);
    T2=sortrows(T2,1);
    T1(:,1)=[];
    T2(:,1)=[];
    [Tmin,T3]=max(T2);%当前位置;

    [ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T );
    t0=T6;%当前时间
    for ii=1:4
        for jj=1:4
            if TT4(ii)==aa(m-4+jj,1)
                aa(m-4+jj,3)=VTT1(ii);
                aa(m-4+jj,5)=TT5(ii);
                aa(m-4+jj,6)=VTT2(ii);
                aa(m-4+jj,7)=aa(m-4+jj,6)+T(5)+T(7);
            end
            if TT5(ii)==aa(m-8+jj,5)
                aa(m-8+jj,7)=VTT2(ii);
            end
        end
    end
    end
    aa(m+1:m+4,1)=TT4';
    aa(m+1:m+4,2)=VTT1;
end

```

```

[m,n]=size(aa);
aa(m-3:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(aa(i,5)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end
end

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);
for i=1:m
    aa(i,4)=2*aa(i,1)-1;
    aa(i,8)=2*aa(i,5);
end
TTT=max(aa(:,7));
end

```

附录八： f203.m

```

function [ T6,VTT1,VTT2,TT4,TT5 ] = f203( T1,T2,T3,t0,dT,T )
Q1=perms([1,3,7]);
Q2=perms([2,4,5,6,8]);
T6=10*3600+1;%记录最小值
for i=1:6
    for j=1:120
        T4=Q1(i,1:3);%前往哪个工序 1
        T5=Q2(j,1:3);%去往哪里工序 2
        d=dT(abs(ceil(T4(1)/2)-T3)+1);%移动
        QT=t0+d;%移动到 T41

        QT=max(QT,T1(T4(1)));%T41 开始下料
        VT1(1,1)=QT;
        QT=QT+T(7-mod(T4(1),2));%T41 下料完毕
        d=dT(abs(ceil(T4(1)/2)-ceil(T5(1)/2))+1);%移动
        QT=QT+d;%移动到 T51
        QT=max(QT,T2(T5(1)));%T51 开始下料
        VT2(1,1)=QT;
    end
end

```



```

QT=QT+T(7-mod(T5(1),2));%T51 下料完毕
if T2(T5(1))==0
    QT=QT;
else
    QT=QT+T(8);%T51 清洗完毕
end

for k=2:3
    d=dT(abs(ceil(T5(k-1)/2)-ceil(T4(k)/2))+1);%移动
    QT=QT+d;%移动到 T4k
    QT=max(QT,T1(T4(k)));%T4k 开始下料
    VT1(k,1)=QT;
    QT=QT+T(7-mod(T4(k),2));%T4k 下料完毕
    d=dT(abs(ceil(T4(k)/2)-ceil(T5(k)/2))+1);%移动
    QT=QT+d;%移动到 T5k
    QT=max(QT,T2(T5(k)));%T5k 开始下料
    VT2(k,1)=QT;
    QT=QT+T(7-mod(T5(k),2));%T5k 下料完毕
    if T2(T5(k))==0
        QT=QT;
    else
        QT=QT+T(8);%T51 清洗完毕
    end
end
end
if QT < T6
    T6=QT;
    VTT1=VT1;
    VTT2=VT2;
    TT4=T4;
    TT5=T5;
end
clear QT VT1 VT2
end
end
end

```

附录九： f204.m

```

function [a,aa,t0] = f204( a,LJ,t0,dT,T )
%初始路径获取
for i=1:4
    if LJ(i)==3
        break;
    end
end
end

```

```

if i==1
    lj=[5,6,LJ(2:4)*2];
elseif i==4
    lj=[LJ(1:3)*2,5,6];
else
    lj=[LJ(1:i-1)*2,5,6,LJ(i+1:4)*2];
end
if i==1
    LJ=[3,3,LJ(2:4)];
elseif i==4
    LJ=[LJ(1:3),3,3];
else
    LJ=[LJ(1:i-1),3,3,LJ(i+1:4)];
end

aa(:,1)=LJ';
aa(:,4)=lj';
i=1;
d=dT(abs(aa(i,1)-a)+1);
t0=t0+d;
aa(i,2)=t0;
aa(i,3)=aa(i,2)+T(4)+T(7-mod(LJ(i),2));
t0=t0+T(7-mod(LJ(i),2));

for i=2:5
    d=dT(abs(aa(i,1)-aa(i-1,1))+1);
    t0=t0+d;
    aa(i,2)=t0;
    aa(i,3)=aa(i,2)+T(4)+T(7-mod(LJ(i),2));
    t0=t0+T(7-mod(LJ(i),2));
end
a=LJ(5);
end

```

附录十： f205.m

```

function [ T6,VTT1,VTT2,TT4,TT5 ] = f205( T1,T2,T3,t0,dT,T )
Q2=perms([1,3,7]);
Q1=perms([2,4,5,6,8]);
T6=10*3600+1;%记录最小值
for i=1:6
    for j=1:120
        T4=Q1(j,1:3);%前往哪个工序 1
        T5=Q2(i,1:3);%去往哪里工序 2
    end
end

```

```

d=dT(abs(ceil(T4(1)/2)-T3)+1);%移动
QT=t0+d;%移动到 T41

QT=max(QT,T1(T4(1)));%T41 开始下料
VT1(1,1)=QT;
QT=QT+T(7-mod(T4(1),2));%T41 下料完毕
d=dT(abs(ceil(T4(1)/2)-ceil(T5(1)/2))+1);%移动
QT=QT+d;%移动到 T51
QT=max(QT,T2(T5(1)));%T51 开始下料
VT2(1,1)=QT;
QT=QT+T(7-mod(T5(1),2));%T51 下料完毕
if T2(T5(1))==0
    QT=QT;%不需要清洗
else
    QT=QT+T(8);%T51 清洗完毕
end

for k=2:3
    d=dT(abs(ceil(T5(k-1)/2)-ceil(T4(k)/2))+1);%移动
    QT=QT+d;%移动到 T4k
    QT=max(QT,T1(T4(k)));%T4k 开始下料
    VT1(k,1)=QT;
    QT=QT+T(7-mod(T4(k),2));%T4k 下料完毕
    d=dT(abs(ceil(T4(k)/2)-ceil(T5(k)/2))+1);%移动
    QT=QT+d;%移动到 T5k
    QT=max(QT,T2(T5(k)));%T5k 开始下料
    VT2(k,1)=QT;
    QT=QT+T(7-mod(T5(k),2));%T5k 下料完毕
    if T2(T5(k))==0
        QT=QT;
    else
        QT=QT+T(8);%T51 清洗完毕
    end
end
end
if QT < T6
    T6=QT;
    VTT1=VT1;
    VTT2=VT2;
    TT4=T4;
    TT5=T5;
end
clear QT VT1 VT2
end
end

```

end

附录十一： f206.m

function [m,QQ] = f206(aa,dT,T,a,t0)

%设置 T1,T2,T3

[m,n]=size(aa);

for i=1:8

 for j=m-4:m

 if i==aa(j,5)

 T2(i,1)=aa(j,7);

 elseif aa(j,5)==0

 T2(i,1)=0;

 end

 end

end

for i=1:m

 for j=1:8

 if j==aa(i,4)

 T1(j,1)=aa(i,3);

 end

 end

end

T3=a;

[T6,VTT1,VTT2,TT4,TT5] = f205(T1,T2,T3,t0,dT,T);

t0=T6;

T3=ceil(TT5(3)/2);

[m,n]=size(aa);

for ii=1:3

 for jj=1:5

 if TT4(ii)==aa(m-5+jj,4)

 aa(m-5+jj,3)=VTT1(ii);

 aa(m-5+jj,4)=TT4(ii);

 aa(m-5+jj,5)=TT5(ii);

 aa(m-5+jj,6)=VTT2(ii);

 aa(m-5+jj,7)=aa(m-5+jj,6)+T(5)+T(7-mod(aa(m-5+jj,5),2));

 end

 end

end

aa(m+1:m+3,4)=TT4';

aa(m+1:m+3,2)=VTT1;

for i=m+1:m+3

```

        aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
    end

clear T1 T2
while t0<8*3600
    [m,n]=size(aa);
    for i=1:8
        for j=1:m
            if i==aa(j,5)
                T2(i,1)=aa(j,7);
            end
        end
    end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            T1(j,1)=aa(i,3);
        end
    end
end

a=ceil(TT5(3)/2);
T3=ceil(TT5(3)/2);

[ T6,VTT1,VTT2,TT4,TT5 ] = f205( T1,T2,T3,t0,dT,T );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=m:-1:1
        if TT4(ii)==aa(jj,4)
            aa(jj,3)=VTT1(ii);
            aa(jj,4)=TT4(ii);
            aa(jj,5)=TT5(ii);
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
            break;
        end
    end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
end

```

```

        end
    end

[m,n]=size(aa);
aa(m-2:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(ceil(aa(i,5)/2)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end
end

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);

for i=1:m
    for j=i+1:m
        if aa(i,5)==aa(j,5)
            aa(i,7)=aa(j,6);
            break;
        end
    end
end

[m,n]=size(aa);
QQ=max(aa(:,7));
end

```

附录十二： Ques201.m

%两道工序的 4,4 模型求解结果

clc,clear

%第一组作业参数

T=[20,33,46,400,378,28,31,25];

%第二组作业参数

%T=[23,41,59,280,500,30,35,30];

%第三组作业参数

%T=[18,32,46,455,182,27,32,25];

```

dT=[0,T(1:3)];
P=8*T(4)/(T(4)+T(5));
%固定刀具
D=[1,2,1,2,1,2,1,2];

%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

%CNC 预计加工完成时间
b=zeros(1,8);

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

%初始路径
newstart=start';
newstart=newstart(:);
for i=1:24
    if newstart(i,1)==1
        newstart(i,2)=0;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    else
        d=dT(abs(newstart(i)-newstart(i-1))+1);
        newstart(i,2)=newstart(i-1,2)+T(6)+d;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    end
end

for i=1:6
    N=4;
    aa=newstart(4*i-3:4*i,:);
    t0=aa(4,2)+T(6);
    [ mm(i),TTT(i) ] = f202( aa,dT,T,t0 );
    clear aa t0 N
end

N=4;i=6;
aa=newstart(4*i-3:4*i,:);
t0=aa(4,2)+T(6);

```

```

aa=[aa;aa];
QQ=perms([1,2,3,4]);
for i=1:24
    d=dT(abs(aa(4,1)-QQ(i,1))+1);
    tQ(i)=t0+d;
    for j=1:4
        if QQ(i,1)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,1))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,2)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,3))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,3)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,4)-QQ(i,3))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,4)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else

```



```

            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
[tQmin,tQind]=min(tQ);
aa(5:8,1)=QQ(tQind,:);
aa(1:4,5)=QQ(tQind,:);
i=tQind;
d=dT(abs(aa(4,1)-QQ(i,1))+1);
tQ(i)=t0+d;
for j=1:4
    if QQ(i,1)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(5,2)=aa(j,3);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=aa(j,3)+T(6);
            aa(1,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(5,2)=tQ(i);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=tQ(i)+T(6);
            aa(1,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(6,2)=aa(j,3);
            aa(2,3)=aa(6,2);
            aa(6,3)=aa(6,2)+T(6)+T(4);
            aa(2,6)=aa(j,3)+T(6);
            aa(2,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(6,2)=tQ(i);
            aa(2,3)=aa(6,2);

```

```

        aa(6,3)=aa(6,2)+T(6)+T(4);
        aa(2,6)=tQ(i)+T(6);
        aa(2,7)=tQ(i)+T(6)+T(7)+T(5);
        tQ(i)=tQ(i)+T(6)+T(7);
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(7,2)=aa(j,3);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=aa(j,3)+T(6);
            aa(3,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(7,2)=tQ(i);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=tQ(i)+T(6);
            aa(3,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(8,2)=aa(j,3);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=aa(j,3)+T(6);
            aa(4,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(8,2)=tQ(i);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=tQ(i)+T(6);

```

```

        aa(4,7)=tQ(i)+T(6)+T(7)+T(5);
        tQ(i)=tQ(i)+T(6)+T(7);
    end
end
end

while t0<8*3600
    [m,n]=size(aa);
    T1=aa(m-3:m,[1,3]);%一道工序结束时间
    T2=aa(m-7:m-4,[5,7]);%二道工序结束时间
    T1=sortrows(T1,1);
    T2=sortrows(T2,1);
    T1(:,1)=[];
    T2(:,1)=[];
    [Tmin,T3]=max(T2);%当前位置;

    [ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T );
    t0=T6;%当前时间
    for ii=1:4
        for jj=1:4
            if TT4(ii)==aa(m-4+jj,1)
                aa(m-4+jj,3)=VTT1(ii);
                aa(m-4+jj,5)=TT5(ii);
                aa(m-4+jj,6)=VTT2(ii);
                aa(m-4+jj,7)=aa(m-4+jj,6)+T(5)+T(7);
            end
            if TT5(ii)==aa(m-8+jj,5)
                aa(m-8+jj,7)=VTT2(ii);
            end
        end
    end
    end
    aa(m+1:m+4,1)=TT4';
    aa(m+1:m+4,2)=VTT1;
    aa(m+1:m+4,3)=aa(m+1:m+4,2)+T(4)+T(6);
    aa(m+1:m+4,4)=aa(m+1:m+4,1)*2-1;
end

[m,n]=size(aa);
aa(m-3:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(aa(i,5)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end

```

```

        end
    end

    bb=[];
    for i=1:m
        if aa(i,7)==0;
            bb=[bb,i];
        end
    end
    aa(bb,:)=[];
    [m,n]=size(aa);
    for i=1:m
        aa(i,4)=2*aa(i,1)-1;
        aa(i,8)=2*aa(i,5);
    end
end

```

附录十三： Ques202.m

%两道工序的 3,5 分配的结果求解

clc,clear

%第一组作业参数

%T=[20,33,46,400,378,28,31,25];

%第二组作业参数

T=[23,41,59,280,500,30,35,30];

%第三组作业参数

%T=[18,32,46,455,182,27,32,25];

dT=[0,T(1:3)];

P=8*T(4)/(T(4)+T(5));

%固定刀具

D=[1,2,1,2,2,2,1,2];

%RGV 当前时间

t0=0;

%RGV 当前位置[1,2,3,4]

a=1;

%CNC 预计加工完成时间

b=zeros(1,8);

%初始路径选择

start=[1,2,4,3];

d=dT(abs(start(1)-a)+1);

t0=t0+d;

```

aa(1,1)=start(1);
aa(1,2)=t0;
t0=t0+T(6);
aa(1,3)=t0+T(4);
aa(1,4)=1;

for i=2:3
    d=dT(abs(start(i)-start(i-1))+1);
    t0=t0+d;
    aa(i,1)=start(i);
    aa(i,2)=t0;
    t0=t0+T(6);
    aa(i,3)=t0+T(4);
end
aa(2,4)=3;
aa(3,4)=7;
a=4;

aa=[aa;aa];
aa(1:3,5)=aa(1:3,1);
for i=1:3
    d=dT(abs(aa(i,5)-a)+1);
    a=aa(i,5);
    t0=t0+d;
    t0=max(t0,aa(i,3));%上料开始
    aa(i,3)=t0;
    t0=t0+T(6);%上料结束
    aa(i,6)=t0;
    t0=t0+T(7);
    aa(i,7)=t0+T(5);
end
aa(4:6,2)=aa(1:3,3);
aa(4:6,3)=aa(4:6,2)+T(6)+T(4);
aa(1:3,5)=aa(1:3,5)*2;
%F203

for i=1:8
    for j=4:6
        if i==aa(j,4)
            T1(i,1)=aa(j,3);
        end
    end
end
end
for i=1:8

```

```

        for j=1:3
            if i==aa(j,5)
                T2(i,1)=aa(j,7);
            end
        end
    end
end

T3=a;

[ T6,VTT1,VTT2,TT4,TT5 ] = f203( T1,T2,T3,t0,dT,T );
a=ceil(TT5(3)/2);
T3=a;
[m,n]=size(aa);
for ii=1:3
    for jj=1:3
        if TT4(ii)==aa(m-3+jj,4)
            aa(m-3+jj,3)=VTT1(ii);
            aa(m-3+jj,4)=TT4(ii);
            aa(m-3+jj,5)=TT5(ii);
            aa(m-3+jj,6)=VTT2(ii);
            aa(m-3+jj,7)=aa(m-3+jj,6)+T(5)+T(7-mod(aa(m-3+jj,5),2));
        end
        if TT5(ii)==aa(m-6+jj,5)
            aa(m-6+jj,7)=VTT2(ii);
        end
    end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
aa(m+1:m+3,3)=VTT1+T(4)+T(6);

clear T1 T2
while t0<8*3600
    [m,n]=size(aa);
    for i=1:8
        for j=m-2:m
            if i==aa(j,4)
                T1(i,1)=aa(j,3);
            end
        end
    end
    for i=1:m-3
        for j=1:8
            if j==aa(i,5)

```

```

        T2(j,1)=aa(i,7);
    end
end
end
a=ceil(TT5(3)/2);
T3=a;

[ T6,VTT1,VTT2,TT4,TT5 ] = f203( T1,T2,T3,t0,dT,T );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=1:3
        if TT4(ii)==aa(m-3+jj,4)
            aa(m-3+jj,3)=VTT1(ii);
            aa(m-3+jj,4)=TT4(ii);
            aa(m-3+jj,5)=TT5(ii);
            aa(m-3+jj,6)=VTT2(ii);
            aa(m-3+jj,7)=aa(m-3+jj,6)+T(5)+T(7-mod(aa(m-3+jj,5),2));
        end
        if TT5(ii)==aa(m-6+jj,5)
            aa(m-6+jj,7)=VTT2(ii);
        end
    end
end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
aa(m+1:m+3,3)=VTT1+T(4)+T(6);
end

[m,n]=size(aa);
aa(m-2:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(ceil(aa(i,5)/2)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end

```

```

end
aa(bb,:)=[];
[m,n]=size(aa);

for i=1:m
    for j=i+1:m
        if aa(i,5)==aa(j,5)
            aa(i,7)=aa(j,6);
            break;
        end
    end
end
end

```

附录十四： Ques203.m

%两道工序的 5,3 分刀的求解

```
clc,clear
```

%第一组作业参数

```
%T=[20,33,46,400,378,28,31,25];
```

%第二组作业参数

```
%T=[23,41,59,280,500,30,35,30];
```

%第三组作业参数

```
T=[18,32,46,455,182,27,32,25];
```

```
dT=[0,T(1:3)];
```

```
P=8*T(4)/(T(4)+T(5));
```

%固定刀具

```
D=[2,1,2,1,1,1,2,1];
```

%RGV 当前时间

```
t0=0;
```

%RGV 当前位置[1,2,3,4]

```
a=1;
```

%CNC 预计加工完成时间

```
b=zeros(1,8);
```

%初始路径选择

```
start=perms([1,2,4,3]);
```

```
for ij=1:24
```

```
    LJ=start(ij,:);
```

```
    [a2,aa,t2] = f204( a,LJ,t0,dT,T );
```

```
    aa(5,7)=0;
```

```
    [ ma(ij),QQ(ij) ] = f206( aa,dT,T,a2,t2 );
```



```

clear aa
end

LJ=start(15,:);
[a,aa,t0] = f204( a,LJ,t0,dT,T );
aa(5,7)=0;
%设置 T1,T2,T3
[m,n]=size(aa);
for i=1:8
    for j=m-4:m
        if i==aa(j,5)
            T2(i,1)=aa(j,7);
        elseif aa(j,5)==0
            T2(i,1)=0;
        end
    end
end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            T1(j,1)=aa(i,3);
        end
    end
end
end

T3=a;
[ T6,VTT1,VTT2,TT4,TT5 ] = f205( T1,T2,T3,t0,dT,T );
t0=T6;
T3=ceil(TT5(3)/2);

[m,n]=size(aa);
for ii=1:3
    for jj=1:5
        if TT4(ii)==aa(m-5+jj,4)
            aa(m-5+jj,3)=VTT1(ii);
            aa(m-5+jj,4)=TT4(ii);
            aa(m-5+jj,5)=TT5(ii);
            aa(m-5+jj,6)=VTT2(ii);
            aa(m-5+jj,7)=aa(m-5+jj,6)+T(5)+T(7-mod(aa(m-5+jj,5),2));
        end
    end
end
end

```

```

end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
end

clear T1 T2
while t0<8*3600
    [m,n]=size(aa);
    for i=1:8
        for j=1:m
            if i==aa(j,5)
                T2(i,1)=aa(j,7);
            end
        end
    end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            T1(j,1)=aa(i,3);
        end
    end
end

a=ceil(TT5(3)/2);
T3=ceil(TT5(3)/2);

[ T6,VTT1,VTT2,TT4,TT5 ] = f205( T1,T2,T3,t0,dT,T );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=m:-1:1
        if TT4(ii)==aa(jj,4)
            aa(jj,3)=VTT1(ii);
            aa(jj,4)=TT4(ii);
            aa(jj,5)=TT5(ii);
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
            break;
        end
    end
end
end
end

```

```

aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
end
end

```

```

[m,n]=size(aa);
aa(m-2:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(ceil(aa(i,5)/2)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end
end

```

```

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);

```

```

for i=1:m
    for j=i+1:m
        if aa(i,5)==aa(j,5)
            aa(i,7)=aa(j,6);
            break;
        end
    end
end
end
[m,n]=size(aa);

```

附录十五： f301.m

```

function [ T6,VTT1,VTT2,TT4,TT5 ] = f301( T1,T2,T3,t0,dT,T,GZBH )
Q1=perms(setdiff([1,3,5,7],GZBH));
[Q1M,Q1N]=size(Q1);
Q2=perms(setdiff([2,4,6,8],GZBH));
[Q2M,Q2N]=size(Q2);
T6=10*3600+1;%记录最小值

```

```

for i=1:Q1M
    for j=1:Q2M
        T4=Q1(i,1:3);%前往哪个工序 1
        T5=Q2(j,1:3);%去往哪里工序 2

        d=dT(abs(ceil(T4(1)/2)-T3)+1);%移动
        QT=t0+d;%移动到 T41

        QT=max(QT,T1(T4(1)));%T41 开始下料
        VT1(1,1)=QT;
        QT=QT+T(7-mod(T4(1),2));%T41 下料完毕
        d=dT(abs(ceil(T4(1)/2)-ceil(T5(1)/2))+1);%移动
        QT=QT+d;%移动到 T51
        QT=max(QT,T2(T5(1)));%T51 开始下料
        VT2(1,1)=QT;
        QT=QT+T(7-mod(T5(1),2));%T51 下料完毕
        if T2(T5(1))==0
            QT=QT;%不需要清洗
        else
            QT=QT+T(8);%T51 清洗完毕
        end

        for k=2:3
            d=dT(abs(ceil(T5(k-1)/2)-ceil(T4(k)/2))+1);%移动
            QT=QT+d;%移动到 T4k
            QT=max(QT,T1(T4(k)));%T4k 开始下料
            VT1(k,1)=QT;
            QT=QT+T(7-mod(T4(k),2));%T4k 下料完毕
            d=dT(abs(ceil(T4(k)/2)-ceil(T5(k)/2))+1);%移动
            QT=QT+d;%移动到 T5k
            QT=max(QT,T2(T5(k)));%T5k 开始下料
            VT2(k,1)=QT;
            QT=QT+T(7-mod(T5(k),2));%T5k 下料完毕
            if T2(T5(k))==0
                QT=QT;
            else
                QT=QT+T(8);%T51 清洗完毕
            end
        end
    end
    if QT < T6
        T6=QT;
        VTT1=VT1;
        VTT2=VT2;
        TT4=T4;
    end
end

```

```

        TT5=T5;
    end
    clear QT VT1 VT2
end
end
end

```

附录十六: Ques301.m

%一道工序的发生故障的求解

```
clc,clear
```

%RGV 移动 1,2,3 个单位所需时间

```
T=[23,41,59];
```

%CNC 加工完成一个一道工序的物料所需时间

```
T=[T,580];
```

%RGV 为 CNC1#, 3#, 5#, 7#一次上下料所需时间

```
T=[T,30];
```

%RGV 为 CNC2#, 4#, 6#, 8#一次上下料所需时间

```
T=[T,35];
```

%RGV 完成一个物料的清洗作业所需时间

```
T=[T,30];
```

```
%1 T=[20,33,46,560,28,31,25];
```

```
%2 T=[23,41,59,580,30,35,30];
```

```
T=[18,32,46,545,27,32,25];
```

%RGV 当前时间

```
t0=0;
```

%RGV 当前位置[1,2,3,4]

```
a=1;
```

%CNC 当前工作状态

```
b=zeros(1,8);
```

%CNC 预计加工完成时间

```
c=zeros(1,8);
```

%初始路径选择

```
start=[ones(6,1),perms([2,3,4])];
```

```
i=6;clear M
```

```
lj=start(i,:);
```

%初始路径移动

```
N=8;
```

```
for i=1:4
```

```
    %RGV 每一步移动
```

```
    dT=[0,T(1:3)];
```

```
    d=dT(abs(a-lj(i))+1);
```

```

a=lj(i);
t0=t0+d;

%RGV 为单个奇数 CNC 上料
M(2*(i-1)+1,1)=t0;
t0=t0+T(5);
b(2*(lj(i)-1)+1)=1;
c(2*(lj(i)-1)+1)=t0+T(4);
M(2*(i-1)+1,2)=2*(lj(i)-1)+1;

%RGV 为单个偶数 CNC 上料
M(2*(i-1)+2,1)=t0;
t0=t0+T(6);
b(2*(lj(i)-1)+2)=1;
c(2*(lj(i)-1)+2)=t0+T(4);
M(2*(i-1)+2,2)=2*(lj(i)-1)+2;
end

GZ=[];
BXYQX=100;
%多步最优
while t0<=8*3600
    if randi(100)==55%约有百分之一的概率发生故障
        [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
        e3 = fStep( f,dT,T,g,h,e );
        e3 = reshape(e3,64,4);
        [minValue,row] = min(e3);
        [minValue,col] = min(minValue);%由 col 知当前本该移动到哪一位，但该位置其
实在之前的加工过程中发生故障
        d=dT(abs(a-col)+1);
        t0=t0+d;
        t0
        aaa=a;
        a=col;
        N=N+1;
        i=a;
        if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
            %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
            cmin=t0;cind=1;
        elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
            [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
            cmin=cmin-T(4+cind);
        else
            [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);

```

```

        cmin=t0;
    end
    GZN=N-8;
    GZBH=M(GZN,2);
    GZSJ=M(GZN,1)+T(6-mod(GZBH,2))+randi(T(4));
    GZZZ=GZSJ+600+randi(600);
    GZ=[GZ;GZN,GZBH,GZSJ,GZZZ] ;
    t0=t0-d;
    a=aaa;
    if t0 <GZSJ
        t0=t0+d;
        a=col;
    end
    M(GZN,1)=999999;
    M(GZN,3)=GZZZ;
    c(2*(i-1)+cind)=GZZZ;
    N=N-1;
    BXYQX=2*(i-1)+cind;
else
    [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
    e3 = fStep( f,dT,T,g,h,e );
    e3 = reshape(e3,64,4);
    [minValue,row] = min(e3);
    [minValue,col] = min(minValue);%由 col 知当前该移动到哪一位
    d=dT(abs(a-col)+1);
    t0=t0+d;
    t0
    a=col;
    N=N+1;
    i=a;
    if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
        %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;cind=1;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
        [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
        cmin=cmin-T(4+cind);
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    else
        [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    end
    M(N,1)=cmin;

```

```

M(N,2)=2*(i-1)+cind;
for i=length(M(:,2))-1:-1:1
    if M(i,2)==2*(a-1)+cind
        M(i,3)=cmin;
        break;
    end
end
if BXYQX==2*(i-1)+cind
    t0=cmin+T(4+cind);
    BXYQX=100;
else
    t0=cmin+T(4+cind)+T(7);
end
end
end

```

```

[m,n]=size(M);
M(m-7:m,:)=[];
M(m-8,3)=M(m-8,1)+T(4+mod(M(m-8,2),2))+T(4);
if M(m-8,3)+T(4+mod(M(m-8,2),2))+T(7)>8*3600
    M(m-8,:)=[];
end
MM(:,1)=M(:,2);
MM(:,2)=M(:,1);
MM(:,3)=M(:,3);

```

```

[m,n]=size(MM);
for i=1:m
    if MM(i,2)==999999;
        MM(i,2)=NaN;
        MM(i,3)=NaN;
    end
end
end

```

附录十七: Ques302.m

%两道工序的发生故障的求解

clc,clear

%第一组作业参数

%1 T=[20,33,46,400,378,28,31,25];

%第二组作业参数

%2 T=[23,41,59,280,500,30,35,30];

%第三组作业参数

T=[18,32,46,455,182,27,32,25];


```

dT=[0,T(1:3)];
P=8*T(4)/(T(4)+T(5));
%固定刀具
D=[1,2,1,2,1,2,1,2];

%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

%CNC 预计加工完成时间
b=zeros(1,8);

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

%初始路径
newstart=start';
newstart=newstart(:);
for i=1:24
    if newstart(i,1)==1
        newstart(i,2)=0;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    else
        d=dT(abs(newstart(i)-newstart(i-1))+1);
        newstart(i,2)=newstart(i-1,2)+T(6)+d;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    end
end

N=4;i=6;
aa=newstart(4*i-3:4*i,:);
t0=aa(4,2)+T(6);

aa=[aa;aa];
QQ=perms([1,2,3,4]);
for i=1:24
    d=dT(abs(aa(4,1)-QQ(i,1))+1);
    tQ(i)=t0+d;
    for j=1:4
        if QQ(i,1)==aa(j,1)

```

```

        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
end
[tQmin,tQind]=min(tQ);
aa(5:8,1)=QQ(tQind,:);
aa(1:4,5)=QQ(tQind,:);

```

```

i=tQind;
d=dT(abs(aa(4,1)-QQ(i,1))+1);
tQ(i)=t0+d;
for j=1:4
    if QQ(i,1)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(5,2)=aa(j,3);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=aa(j,3)+T(6);
            aa(1,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(5,2)=tQ(i);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=tQ(i)+T(6);
            aa(1,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(6,2)=aa(j,3);
            aa(2,3)=aa(6,2);
            aa(6,3)=aa(6,2)+T(6)+T(4);
            aa(2,6)=aa(j,3)+T(6);
            aa(2,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(6,2)=tQ(i);
            aa(2,3)=aa(6,2);
            aa(6,3)=aa(6,2)+T(6)+T(4);
            aa(2,6)=tQ(i)+T(6);
            aa(2,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);

```

```

tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(7,2)=aa(j,3);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=aa(j,3)+T(6);
            aa(3,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(7,2)=tQ(i);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=tQ(i)+T(6);
            aa(3,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(8,2)=aa(j,3);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=aa(j,3)+T(6);
            aa(4,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(8,2)=tQ(i);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=tQ(i)+T(6);
            aa(4,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
t0=tQ(i);
UTT=aa(:,6);
SI=length(UTT);

```

```

for ijk=1:SI
    if UTT(ijk)>100000
        UTT(ijk)=0;
    end
end
[nmn,nmi]=max(UTT);
clear UTT SI ijk
a=aa(nmi,5);
GZ=[];
BXYQX=100;
GZZZ=0;
while t0<8*3600
    if randi(100)>55.5&&randi(100)<63.5
        if randi(8)==4%约有百分之一的概率发生故障
            AA=randi(8);
            AAA=ceil(AA/4);%1 表示第一道工序故障，2 表示第二道工序故障
            AA=AA-4*(AAA-1);%第 AAA 道工序的第 AA 个元件故障
            [m,n]=size(aa);
            if AAA==1
                GZBH=aa(m-4+AA,4);
                for i=m-3:m
                    if aa(i,1)==AA
                        GZN=i;
                        GZSJ=aa(i,2)+T(6)+randi(T(4));
                        GZZZ=GZSJ+600+randi(600);
                        aa(i,3)=GZZZ;
                        aa(i,2)=999999;
                        break;
                    end
                end
            else
                GZBH=aa(m-8+AA,5)*2;
                for i=m-7:m-4
                    if aa(i,5)==AA
                        GZN=i;
                        GZSJ=aa(i,6)+T(7)+randi(T(5));
                        GZZZ=GZSJ+600+randi(600);
                        aa(i,7)=GZZZ;
                        aa(i,6)=999999;
                    end
                end
            end
            end
            GZ=[GZ;GZN,GZBH,GZSJ,GZZZ] ;
        end
    end
end

```

```

end
if t0<GZZZ
    clear T2 T1 TT4 TT5 VTT1 VTT2
    [m,n]=size(aa);
    for i=1:8
        for j=1:m
            if i==aa(j,5)*2
                T2(i,1)=aa(j,7);
            end
        end
    end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            T1(j,1)=aa(i,3);
        end
    end
end
end
UTT=aa(:,6);
SI=length(UTT);
for ijk=1:SI
    if UTT(ijk)>100000
        UTT(ijk)=0;
    end
end
end
[nmn,nmi]=max(UTT);
clear UTT SI ijk
T3=aa(nmi,5);

[ T6,VTT1,VTT2,TT4,TT5 ] = f301( T1,T2,T3,t0,dT,T,GZBH );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=m:-1:1
        if TT4(ii)==aa(jj,4)
            aa(jj,3)=VTT1(ii);
            aa(jj,4)=TT4(ii);
            aa(jj,1)=(aa(jj,4)+1)/2;
            aa(jj,5)=TT5(ii)/2;
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
            break;
        end
    end
end

```

```

        end
    end
    aa(m+1:m+3,4)=TT4';
    aa(m+1:m+3,2)=VTT1;
    for i=m+1:m+3
        aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
        aa(i,1)=(aa(i,4)+1)/2;
    end
    if t0>GZZZ&&AAA==1
        clear T2 T1 TT4 TT5 VTT1 VTT2
        [m,n]=size(aa);
        for i=1:8
            for j=1:m
                if i==aa(j,5)*2
                    T2(i,1)=aa(j,7);
                end
            end
        end
        end
        end

        for i=1:m
            for j=1:8
                if j==aa(i,4)
                    T1(j,1)=aa(i,3);
                end
            end
        end
        end
        UTT=aa(:,6);
        SI=length(UTT);
        for ijk=1:SI
            if UTT(ijk)>100000
                UTT(ijk)=0;
            end
        end
        end
        [nmn,nmi]=max(UTT);
        clear UTT SI ijk
        T3=aa(nmi,5);

    [ T6,VTT1,VTT2,TT4,TT5 ] = f301( T1,T2,T3,t0,dT,T,0 );
    t0=T6;%当前时间
    [m,n]=size(aa);
    for ii=1:3
        for jj=m:-1:1
            if TT4(ii)==aa(jj,4)
                aa(jj,3)=VTT1(ii);
            end
        end
    end

```

```

        aa(jj,4)=TT4(ii);
        aa(jj,1)=(aa(jj,4)+1)/2;
        aa(jj,5)=TT5(ii)/2;
        aa(jj,6)=VTT2(ii);
        aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
        break;
    end
end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
    aa(i,1)=(aa(i,4)+1)/2;
end
[m,n]=size(aa);
for i=1:m
    if abs(aa(i,7)-aa(GZN,6))<100
        aa(GZN,5:7)=aa(i,5:7);
    end
end
end
else
clear T1 T2 TZ2 TZ1 TT4 TT5 VTT1 VTT2
[m,n]=size(aa);
for i=1:8
    for j=1:m
        if i==aa(j,5)*2
            TZ2(i,1)=aa(j,7);
        end
    end
end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            %aa(i,3)=aa(i,2)+T(6)+T(4);
            TZ1(j,1)=aa(i,3);
        end
    end
end
end
T1=TZ1(find(TZ1~=0));
T2=TZ2(find(TZ2~=0));

```



```

    UTT=aa(:,6);
    SI=length(UTT);
    for ijk=1:SI
        if UTT(ijk)>100000
            UTT(ijk)=0;
        end
    end
    [nmn,nmi]=max(UTT);
    clear UTT SI ijk
    T3=aa(nmi,5);%当前位置;

    [ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T );
    t0=T6;%当前时间
    for ii=1:4
        for jj=m:-1:1
            if TT4(ii)==aa(jj,1)
                aa(jj,3)=VTT1(ii);
                aa(jj,5)=TT5(ii);
                aa(jj,6)=VTT2(ii);
                aa(jj,7)=aa(jj,6)+T(5)+T(7);
                break;
            end
        end
    end
    aa(m+1:m+4,1)=TT4';
    aa(m+1:m+4,4)=TT4'*2-1;
    aa(m+1:m+4,2)=VTT1;
    aa(m+1:m+4,3)=VTT1+T(6)+T(4);
end
end

[m,n]=size(aa);
aa(m-3:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(aa(i,5)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end

bb=[];
for i=1:m
    if aa(i,7)==0;

```

```

        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);
for i=1:m
    aa(i,4)=2*aa(i,1)-1;
    aa(i,8)=2*aa(i,5);
end

aaA=[aa(:,4),aa(:,2),aa(:,3),aa(:,5),aa(:,6),aa(:,7)];
[maa,naa]=size(aaA);
for i=1:maa
    if sum(aaA(i,:))>999999
        aaA(i,[2,3,5,6])=NaN;
    end
end

aaA(:,4)=aaA(:,4)*2;

```

附录十八： f302.m

```

function [m,m1,TMT,GZ] = f302( )
%RGV 移动 1,2,3 个单位所需时间
T=[23,41,59];
%CNC 加工完成一个一道工序的物料所需时间
T=[T,580];
%RGV 为 CNC1#, 3#, 5#, 7#一次上下料所需时间
T=[T,30];
%RGV 为 CNC2#, 4#, 6#, 8#一次上下料所需时间
T=[T,35];
%RGV 完成一个物料的清洗作业所需时间
T=[T,30];
%1 T=[20,33,46,560,28,31,25];
%2 T=[23,41,59,580,30,35,30];
    T=[18,32,46,545,27,32,25];
%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

%CNC 当前工作状态
b=zeros(1,8);
%CNC 预计加工完成时间
c=zeros(1,8);

```

```

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

i=6;clear M
lj=start(i,:);
%初始路径移动
N=8;
for i=1:4
    %RGV 每一步移动
    dT=[0,T(1:3)];
    d=dT(abs(a-lj(i))+1);
    a=lj(i);
    t0=t0+d;

    %RGV 为单个奇数 CNC 上料
    M(2*(i-1)+1,1)=t0;
    t0=t0+T(5);
    b(2*(lj(i)-1)+1)=1;
    c(2*(lj(i)-1)+1)=t0+T(4);
    M(2*(i-1)+1,2)=2*(lj(i)-1)+1;

    %RGV 为单个偶数 CNC 上料
    M(2*(i-1)+2,1)=t0;
    t0=t0+T(6);
    b(2*(lj(i)-1)+2)=1;
    c(2*(lj(i)-1)+2)=t0+T(4);
    M(2*(i-1)+2,2)=2*(lj(i)-1)+2;
end

GZ=[];
BXYQX=100;
%多步最优
while t0<=8*3600
    if randi(100)==55%约有百分之一的概率发生故障
        [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
        e3 = fStep( f,dT,T,g,h,e );
        e3 = reshape(e3,64,4);
        [minValue,row] = min(e3);
        [minValue,col] = min(minValue);%由 col 知当前本该移动到哪一位，但该位置其
实在之前的加工过程中发生故障
        d=dT(abs(a-col)+1);
        t0=t0+d;
        t0
    end
end

```

```

aaa=a;
a=col;
N=N+1;
i=a;
if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
    %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;cind=1;
elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
    [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
    cmin=cmin-T(4+cind);
else
    [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;
end
GZN=N-8;
GZBH=M(GZN,2);
GZSJ=M(GZN,1)+T(6-mod(GZBH,2))+randi(T(4));
%GZZZ=GZSJ+600+randi(600);*****
GZZZ=GZSJ+1200;
GZ=[GZ;GZN,GZBH,GZSJ,GZZZ] ;
t0=t0-d;
a=aaa;
if t0 <GZSJ
    t0=t0+d;
    a=col;
end
M(GZN,1)=999999;
M(GZN,3)=GZZZ;
c(2*(i-1)+cind)=GZZZ;
N=N-1;
BXYQX=2*(i-1)+cind;
else
    [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
    e3 = fStep( f,dT,T,g,h,e );
    e3 = reshape(e3,64,4);
    [minValue,row] = min(e3);
    [minValue,col] = min(minValue);%由 col 知当前该移动到哪一位
    d=dT(abs(a-col)+1);
    t0=t0+d;
    t0
    a=col;
    N=N+1;
    i=a;
    if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)

```

```

        %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;cind=1;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
        %[cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
        cmin=cmin-T(4+cind);
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    else
        %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
        cmin=t0;
        c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
    end
    M(N,1)=cmin;
    M(N,2)=2*(i-1)+cind;
    for i=length(M(:,2))-1:-1:1
        if M(i,2)==2*(a-1)+cind
            M(i,3)=cmin;
            break;
        end
    end
    if BXYQX==2*(i-1)+cind
        t0=cmin+T(4+cind);
        BXYQX=100;
    else
        t0=cmin+T(4+cind)+T(7);
    end
end
end
end

```

```

[m,n]=size(M);
M(m-7:m,:)=[];
M(m-8,3)=M(m-8,1)+T(4+mod(M(m-8,2),2))+T(4);
if M(m-8,3)+T(4+mod(M(m-8,2),2))+T(7)>8*3600
    M(m-8,:)=[];
end
MM(:,1)=M(:,2);
MM(:,2)=M(:,1);
MM(:,3)=M(:,3);

```

```

[m,n]=size(MM);
[m1,n1]=size(GZ);
for i=1:m
    if MM(i,2)==999999;

```

```

        MM(i,2)=NaN;
        MM(i,3)=NaN;
    end
end
TMT=m-m1;
end

```

附录十九： f303.m

```

function [m,m1,TMT,GZ] = f303( )
%第一组作业参数
%1 T=[20,33,46,400,378,28,31,25];
%第二组作业参数
%2 T=[23,41,59,280,500,30,35,30];
%第三组作业参数
    T=[18,32,46,455,182,27,32,25];

```

```

dT=[0,T(1:3)];
P=8*T(4)/(T(4)+T(5));
%固定刀具
D=[1,2,1,2,1,2,1,2];

```

```

%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

```

```

%CNC 预计加工完成时间
b=zeros(1,8);

```

```

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

```

```

%初始路径
newstart=start';
newstart=newstart(:);
for i=1:24
    if newstart(i,1)==1
        newstart(i,2)=0;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    else
        d=dT(abs(newstart(i)-newstart(i-1))+1);
        newstart(i,2)=newstart(i-1,2)+T(6)+d;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
    end
end

```

```

        newstart(i,4)=2*(newstart(i,1)-1)+1;
    end
end

N=4;i=6;
aa=newstart(4*i-3:4*i,:);
t0=aa(4,2)+T(6);

aa=[aa;aa];
QQ=perms([1,2,3,4]);
for i=1:24
    d=dT(abs(aa(4,1)-QQ(i,1))+1);
    tQ(i)=t0+d;
    for j=1:4
        if QQ(i,1)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,1))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,2)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
    d=dT(abs(QQ(i,2)-QQ(i,3))+1);
    tQ(i)=tQ(i)+d;
    for j=1:4
        if QQ(i,3)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else
                tQ(i)=tQ(i)+T(6)+T(7);
            end
        end
    end
end
end

```

```

d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
[tQmin,tQind]=min(tQ);
aa(5:8,1)=QQ(tQind,:);
aa(1:4,5)=QQ(tQind,:);
i=tQind;
d=dT(abs(aa(4,1)-QQ(i,1))+1);
tQ(i)=t0+d;
for j=1:4
    if QQ(i,1)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(5,2)=aa(j,3);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=aa(j,3)+T(6);
            aa(1,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(5,2)=tQ(i);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=tQ(i)+T(6);
            aa(1,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(6,2)=aa(j,3);
            aa(2,3)=aa(6,2);

```



```

        aa(6,3)=aa(6,2)+T(6)+T(4);
        aa(2,6)=aa(j,3)+T(6);
        aa(2,7)=aa(j,3)+T(6)+T(7)+T(5);
        tQ(i)=aa(j,3)+T(6)+T(7);
    else
        aa(6,2)=tQ(i);
        aa(2,3)=aa(6,2);
        aa(6,3)=aa(6,2)+T(6)+T(4);
        aa(2,6)=tQ(i)+T(6);
        aa(2,7)=tQ(i)+T(6)+T(7)+T(5);
        tQ(i)=tQ(i)+T(6)+T(7);
    end
end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(7,2)=aa(j,3);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=aa(j,3)+T(6);
            aa(3,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(7,2)=tQ(i);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=tQ(i)+T(6);
            aa(3,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(8,2)=aa(j,3);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=aa(j,3)+T(6);

```

```

aa(4,7)=aa(j,3)+T(6)+T(7)+T(5);
tQ(i)=aa(j,3)+T(6)+T(7);
else
aa(8,2)=tQ(i);
aa(4,3)=aa(8,2);
aa(8,3)=aa(8,2)+T(6)+T(4);
aa(4,6)=tQ(i)+T(6);
aa(4,7)=tQ(i)+T(6)+T(7)+T(5);
tQ(i)=tQ(i)+T(6)+T(7);
end
end
end
t0=tQ(i);
UTT=aa(:,6);
SI=length(UTT);
for ijk=1:SI
    if UTT(ijk)>100000
        UTT(ijk)=0;
    end
end
end
[nmn,nmi]=max(UTT);
clear UTT SI ijk
a=aa(nmi,5);
GZ=[];
BXYQX=100;
GZZZ=0;
while t0<8*3600
    if randi(100)>55.5&&randi(100)<63.5
        if randi(6)==4%约有百分之一的概率发生故障
            %BBQ=randi(600);BBQ=0;BBQ=300;BBQ=600;
            BBQ=600;%随机*****
            AA=randi(8);
            AAA=ceil(AA/4);%1 表示第一道工序故障，2 表示第二道工序故障
            AA=AA-4*(AAA-1);%第 AAA 道工序的第 AA 个元件故障
            [m,n]=size(aa);
            if AAA==1
                GZBH=aa(m-4+AA,4);
                for i=m-3:m
                    if aa(i,1)==AA
                        GZN=i;
                        GZSJ=aa(i,2)+T(6)+randi(T(4));
                        GZZZ=GZSJ+600+BBQ;
                        aa(i,3)=GZZZ;
                        aa(i,2)=999999;
                    end
                end
            end
        end
    end
end

```

```

                break;
            end
        end
    else
        GZBH=aa(m-8+AA,5)*2;
        for i=m-7:m-4
            if aa(i,5)==AA
                GZN=i;
                GZSJ=aa(i,6)+T(7)+randi(T(5));
                GZZZ=GZSJ+600+BBQ;
                aa(i,7)=GZZZ;
                aa(i,6)=999999;
            end
        end
    end
    end
    GZ=[GZ;GZN,GZBH,GZSJ,GZZZ] ;
end
end
if t0<GZZZ
    clear T2 T1 TT4 TT5 VTT1 VTT2
    [m,n]=size(aa);
    for i=1:8
        for j=1:m
            if i==aa(j,5)*2
                T2(i,1)=aa(j,7);
            end
        end
    end
    end
    for i=1:m
        for j=1:8
            if j==aa(i,4)
                T1(j,1)=aa(i,3);
            end
        end
    end
    end
    UTT=aa(:,6);
    SI=length(UTT);
    for ijk=1:SI
        if UTT(ijk)>100000
            UTT(ijk)=0;
        end
    end
    end
    [nmn,nmi]=max(UTT);

```

```

clear UTT SI ijk
T3=aa(nmi,5);

[ T6,VTT1,VTT2,TT4,TT5 ] = f301( T1,T2,T3,t0,dT,T,GZBH );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=m:-1:1
        if TT4(ii)==aa(jj,4)
            aa(jj,3)=VTT1(ii);
            aa(jj,4)=TT4(ii);
            aa(jj,1)=(aa(jj,4)+1)/2;
            aa(jj,5)=TT5(ii)/2;
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
            break;
        end
    end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
    aa(i,1)=(aa(i,4)+1)/2;
end
if t0>GZZZ&&AAA==1
    clear T2 T1 TT4 TT5 VTT1 VTT2
    [m,n]=size(aa);
    for i=1:8
        for j=1:m
            if i==aa(j,5)*2
                T2(i,1)=aa(j,7);
            end
        end
    end
    for i=1:m
        for j=1:8
            if j==aa(i,4)
                T1(j,1)=aa(i,3);
            end
        end
    end
    UTT=aa(:,6);

```

```

SI=length(UTT);
for ijk=1:SI
    if UTT(ijk)>100000
        UTT(ijk)=0;
    end
end
[nmn,nmi]=max(UTT);
clear UTT SI ijk
T3=aa(nmi,5);

[ T6,VTT1,VTT2,TT4,TT5 ] = f301( T1,T2,T3,t0,dT,T,0 );
t0=T6;%当前时间
[m,n]=size(aa);
for ii=1:3
    for jj=m:-1:1
        if TT4(ii)==aa(jj,4)
            aa(jj,3)=VTT1(ii);
            aa(jj,4)=TT4(ii);
            aa(jj,1)=(aa(jj,4)+1)/2;
            aa(jj,5)=TT5(ii)/2;
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7-mod(aa(jj,5),2));
            break;
        end
    end
end
aa(m+1:m+3,4)=TT4';
aa(m+1:m+3,2)=VTT1;
for i=m+1:m+3
    aa(i,3)=VTT1(i-m)+T(4)+T(7-mod(aa(i,4),2));
    aa(i,1)=(aa(i,4)+1)/2;
end
[m,n]=size(aa);
for i=1:m
    if abs(aa(i,7)-aa(GZN,6))<100
        aa(GZN,5:7)=aa(i,5:7);
    end
end
end
else
clear T1 T2 TZ2 TZ1 TT4 TT5 VTT1 VTT2
[m,n]=size(aa);
for i=1:8
    for j=1:m

```

```

        if i==aa(j,5)*2
            TZ2(i,1)=aa(j,7);
        end
    end
end

for i=1:m
    for j=1:8
        if j==aa(i,4)
            %aa(i,3)=aa(i,2)+T(6)+T(4);
            TZ1(j,1)=aa(i,3);
        end
    end
end

T1=TZ1(find(TZ1~=0));
T2=TZ2(find(TZ2~=0));

UTT=aa(:,6);
SI=length(UTT);
for ijk=1:SI
    if UTT(ijk)>100000
        UTT(ijk)=0;
    end
end

[nmn,nmi]=max(UTT);
clear UTT SI ijk
T3=aa(nmi,5);%当前位置;

[ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T );
t0=T6;%当前时间
for ii=1:4
    for jj=m:-1:1
        if TT4(ii)==aa(jj,1)
            aa(jj,3)=VTT1(ii);
            aa(jj,5)=TT5(ii);
            aa(jj,6)=VTT2(ii);
            aa(jj,7)=aa(jj,6)+T(5)+T(7);
            break;
        end
    end
end

aa(m+1:m+4,1)=TT4';
aa(m+1:m+4,4)=TT4'*2-1;
aa(m+1:m+4,2)=VTT1;

```

```

        aa(m+1:m+4,3)=VTT1+T(6)+T(4);
    end
end

[m,n]=size(aa);
aa(m-3:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(aa(i,5)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);
for i=1:m
    aa(i,4)=2*aa(i,1)-1;
    aa(i,8)=2*aa(i,5);
end

aaA=[aa(:,4),aa(:,2),aa(:,3),aa(:,5),aa(:,6),aa(:,7)];
[maa,naa]=size(aaA);
for i=1:maa
    if sum(aaA(i,:))>999999
        aaA(i,[2,3,5,6])=NaN;
    end
end

aaA(:,4)=aaA(:,4)*2;

[m,n]=size(aaA);
[m1,n1]=size(GZ);

TMT=m-m1;
end

```

附录二十： Ques303.m

%一道工序的修复时间，随机，10,15,20 的 20 组随机

clc,clear

G=[];

for i=1:20

 [m(i),m1(i),TMT(i),GZ] = f302();

 G=[G;GZ];

 clear GZ

end

A=[TMT;m1];

A=A';

B(1)=A(1,2);

for i=2:20

 B(i)=B(i-1)+A(i,2);

end

[mm,nn]=size(G);

C=zeros(1,mm);

C(B)=1;

G=[G,C'];

D=zeros(19,5);

for i=1:19

 D(i,1)=i+1;

end

for i=1:19

 [mm,nn]=size(G);

 G=[G(1:B(i),:);D(i,:);G(B(i)+1:mm,:)];

 B=B+1;

end

G=[[1,zeros(1,4)];G];

G(:,5)=[];

ZM=[mean(A(:,1)),mean(A(:,2))];

ZMM=std(A,0,1);

A=[A;ZM;ZMM];

附录二十一： Ques304.m

%两道工序的修复时间，随机，10,15,20 的 20 组随机

clc,clear


```

G=[];
for i=1:20
    [m(i),m1(i),TMT(i),GZ] = f304( );
    G=[G;GZ];
    clear GZ
end

A=[TMT;m1];
A=A';

B(1)=A(1,2);
for i=2:20
    B(i)=B(i-1)+A(i,2);
end

[mm,nn]=size(G);
C=zeros(1,mm);
C(B)=1;

G=[G,C'];
D=zeros(19,5);
for i=1:19
    D(i,1)=i+1;
end

for i=1:19
    [mm,nn]=size(G);
    G=[G(1:B(i),:);D(i,:);G(B(i)+1:mm,:)];
    B=B+1;
end
G=[[1,zeros(1,4)];G];

G(:,5)=[];

ZM=[mean(A(:,1)),mean(A(:,2))];
ZMM=std(A,0,1);

A=[A;ZM;ZMM];

附录二十二： f401.m
function m = f401( T )
%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]

```

```

a=1;

%CNC 当前工作状态
b=zeros(1,8);
%CNC 预计加工完成时间
c=zeros(1,8);

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

i=6;clear M
lj=start(i,:);
%初始路径移动
N=8;
dT=[0,T(1:3)];
for i=1:4
    %RGV 每一步移动
    d=dT(abs(a-lj(i))+1);
    a=lj(i);
    t0=t0+d;

    %RGV 为单个奇数 CNC 上料
    M(2*(i-1)+1,1)=t0;
    t0=t0+T(5);
    b(2*(lj(i)-1)+1)=1;
    c(2*(lj(i)-1)+1)=t0+T(4);
    M(2*(i-1)+1,2)=2*(lj(i)-1)+1;

    %RGV 为单个偶数 CNC 上料
    M(2*(i-1)+2,1)=t0;
    t0=t0+T(6);
    b(2*(lj(i)-1)+2)=1;
    c(2*(lj(i)-1)+2)=t0+T(4);
    M(2*(i-1)+2,2)=2*(lj(i)-1)+2;
end

%多步最优
while t0<=8*3600

    [e,f,g,h,]= oneStep( a,dT,T,t0,c );%a,t0,c 改变为 f,g,h.
    e3 = fStep( f,dT,T,g,h,e );
    e3 = reshape(e3,64,4);
    [minValue,row] = min(e3);
    [minValue,col] = min(minValue);

```

```

d=dT(abs(a-col)+1);
t0=t0+d;
a=col;
N=N+1;
i=a;
if t0>c(2*(i-1)+1)&& t0>c(2*(i-1)+2)
    %[cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;cind=1;
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
elseif t0<c(2*(i-1)+1)&& t0<c(2*(i-1)+2)
    [cmin,cind]=min([c(2*(i-1)+1)+T(5),c(2*(i-1)+2)+T(6)]);
    cmin=cmin-T(4+cind);
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
else
    [cmin,cind]=min([c(2*(i-1)+1),c(2*(i-1)+2)]);
    cmin=t0;
    c(2*(i-1)+cind)=cmin+T(4+cind)+T(4);
end
M(N,1)=cmin;
M(N,2)=2*(i-1)+cind;
for i=length(M(:,2))-1:-1:1
    if M(i,2)==2*(a-1)+cind
        M(i,3)=cmin;
        break;
    end
end
t0=cmin+T(4+cind)+T(7);
end
[m,n]=size(M);
M(m-7:m,:)=[];
M(m-8,3)=M(m-8,1)+T(4+mod(M(m-8,2),2))+T(4);
if M(m-8,3)+T(4+mod(M(m-8,2),2))+T(7)>8*3600
    M(m-8,:)=[];
end
MM(:,1)=M(:,2);
MM(:,2)=M(:,1);
MM(:,3)=M(:,3);
[m,n]=size(M);
end

```

附录二十三： f402.m
function m = f402(T)

```
dT=[0,T(1:3)];
```

```

P=8*T(4)/(T(4)+T(5));
%固定刀具
D=[1,2,1,2,1,2,1,2];

%RGV 当前时间
t0=0;
%RGV 当前位置[1,2,3,4]
a=1;

%CNC 预计加工完成时间
b=zeros(1,8);

%初始路径选择
start=[ones(6,1),perms([2,3,4])];

%初始路径
newstart=start';
newstart=newstart(:);
for i=1:24
    if newstart(i,1)==1
        newstart(i,2)=0;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    else
        d=dT(abs(newstart(i)-newstart(i-1))+1);
        newstart(i,2)=newstart(i-1,2)+T(6)+d;
        newstart(i,3)=newstart(i,2)+T(6)+T(4);
        newstart(i,4)=2*(newstart(i,1)-1)+1;
    end
end
end
N=4;i=6;
aa=newstart(4*i-3:4*i,:);
t0=aa(4,2)+T(6);

aa=[aa;aa];
QQ=perms([1,2,3,4]);
for i=1:24
    d=dT(abs(aa(4,1)-QQ(i,1))+1);
    tQ(i)=t0+d;
    for j=1:4
        if QQ(i,1)==aa(j,1)
            if tQ(i)<aa(j,3)
                tQ(i)=aa(j,3)+T(6)+T(7);
            else

```

```

            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end
[tQmin,tQind]=min(tQ);
aa(5:8,1)=QQ(tQind,:);
aa(1:4,5)=QQ(tQind,:);
i=tQind;
d=dT(abs(aa(4,1)-QQ(i,1))+1);
tQ(i)=t0+d;

```

```

for j=1:4
    if QQ(i,1)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(5,2)=aa(j,3);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=aa(j,3)+T(6);
            aa(1,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(5,2)=tQ(i);
            aa(1,3)=aa(5,2);
            aa(5,3)=aa(5,2)+T(6)+T(4);
            aa(1,6)=tQ(i)+T(6);
            aa(1,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,1))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,2)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(6,2)=aa(j,3);
            aa(2,3)=aa(6,2);
            aa(6,3)=aa(6,2)+T(6)+T(4);
            aa(2,6)=aa(j,3)+T(6);
            aa(2,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(6,2)=tQ(i);
            aa(2,3)=aa(6,2);
            aa(6,3)=aa(6,2)+T(6)+T(4);
            aa(2,6)=tQ(i)+T(6);
            aa(2,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,2)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,3)==aa(j,1)

```

```

        if tQ(i)<aa(j,3)
            aa(7,2)=aa(j,3);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=aa(j,3)+T(6);
            aa(3,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(7,2)=tQ(i);
            aa(3,3)=aa(7,2);
            aa(7,3)=aa(7,2)+T(6)+T(4);
            aa(3,6)=tQ(i)+T(6);
            aa(3,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
d=dT(abs(QQ(i,4)-QQ(i,3))+1);
tQ(i)=tQ(i)+d;
for j=1:4
    if QQ(i,4)==aa(j,1)
        if tQ(i)<aa(j,3)
            aa(8,2)=aa(j,3);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=aa(j,3)+T(6);
            aa(4,7)=aa(j,3)+T(6)+T(7)+T(5);
            tQ(i)=aa(j,3)+T(6)+T(7);
        else
            aa(8,2)=tQ(i);
            aa(4,3)=aa(8,2);
            aa(8,3)=aa(8,2)+T(6)+T(4);
            aa(4,6)=tQ(i)+T(6);
            aa(4,7)=tQ(i)+T(6)+T(7)+T(5);
            tQ(i)=tQ(i)+T(6)+T(7);
        end
    end
end
end

while t0<8*3600
    [m,n]=size(aa);
    T1=aa(m-3:m,[1,3]);%一道工序结束时间
    T2=aa(m-7:m-4,[5,7]);%二道工序结束时间
    T1=sortrows(T1,1);

```

```

T2=sortrows(T2,1);
T1(:,1)=[];
T2(:,1)=[];
[Tmin,T3]=max(T2);%当前位置;

[ T6,VTT1,VTT2,TT4,TT5 ] = f201( T1,T2,T3,t0,dT,T );
t0=T6;%当前时间
for ii=1:4
    for jj=1:4
        if TT4(ii)==aa(m-4+jj,1)
            aa(m-4+jj,3)=VTT1(ii);
            aa(m-4+jj,5)=TT5(ii);
            aa(m-4+jj,6)=VTT2(ii);
            aa(m-4+jj,7)=aa(m-4+jj,6)+T(5)+T(7);
        end
        if TT5(ii)==aa(m-8+jj,5)
            aa(m-8+jj,7)=VTT2(ii);
        end
    end
end
aa(m+1:m+4,1)=TT4';
aa(m+1:m+4,2)=VTT1;
aa(m+1:m+4,3)=aa(m+1:m+4,2)+T(4)+T(6);
aa(m+1:m+4,4)=aa(m+1:m+4,1)*2-1;
end

[m,n]=size(aa);
aa(m-3:m,:)=[];
[m,n]=size(aa);
for i=m:-1:m-15
    d=dT(abs(aa(i,5)-1)+1);%移动
    if aa(i,7)+T(6)+T(7)+d>8*3600
        aa(i,7)=0;
    end
end

bb=[];
for i=1:m
    if aa(i,7)==0;
        bb=[bb,i];
    end
end
aa(bb,:)=[];
[m,n]=size(aa);

```



```

for i=1:m
    aa(i,4)=2*aa(i,1)-1;
    aa(i,8)=2*aa(i,5);
end

```

```

end

```

附录二十四：LingMin1.m

%一道工序的灵敏度分析

```

clc,clear

```

%设置上下限为 5%的灵敏度分析

```

x=-0.05:0.01:0.05;

```

```

TT1=[20,33,46,560,28,31,25];

```

```

for i=1:11

```

```

    T1(i,:)= [TT1(1:3)+TT1(1:3)*x(i),TT1(4:7)];%RGV 移动速度

```

```

    T1(i+11,:)= [TT1(1:3),TT1(4)+TT1(4)*x(i),TT1(5:7)];%CNC 加工速度

```

```

    T1(i+22,:)= [TT1(1:4),TT1(5:6)+TT1(5:6)*x(i),TT1(7)];%RGV 上下料速度

```

```

    T1(i+33,:)= [TT1(1:6),TT1(7)+TT1(7)*x(i)];%RGV 清洗熟料步骤速度

```

```

end

```

```

for i=1:44

```

```

    m1(i) = f401(T1(i,:));

```

```

end

```

```

TT2=[23,41,59,580,30,35,30];

```

```

for i=1:11

```

```

    T2(i,:)= [TT2(1:3)+TT2(1:3)*x(i),TT2(4:7)];%RGV 移动速度

```

```

    T2(i+11,:)= [TT2(1:3),TT2(4)+TT2(4)*x(i),TT2(5:7)];%CNC 加工速度

```

```

    T2(i+22,:)= [TT2(1:4),TT2(5:6)+TT2(5:6)*x(i),TT2(7)];%RGV 上下料速度

```

```

    T2(i+33,:)= [TT2(1:6),TT2(7)+TT2(7)*x(i)];%RGV 清洗熟料步骤速度

```

```

end

```

```

for i=1:44

```

```

    m2(i) = f401(T2(i,:));

```

```

end

```

```

TT3=[18,32,46,545,27,32,25];

```

```

for i=1:11

```

```

    T3(i,:)= [TT3(1:3)+TT3(1:3)*x(i),TT3(4:7)];%RGV 移动速度

```

```

    T3(i+11,:)= [TT3(1:3),TT3(4)+TT3(4)*x(i),TT3(5:7)];%CNC 加工速度

```

```

    T3(i+22,:)= [TT3(1:4),TT3(5:6)+TT3(5:6)*x(i),TT3(7)];%RGV 上下料速度

```

```

    T3(i+33,:)= [TT3(1:6),TT3(7)+TT3(7)*x(i)];%RGV 清洗熟料步骤速度

```

```

end

```

```

for i=1:44

```

```

    m3(i) = f401(T3(i,:));

```

```

end

```

```

subplot(1,3,1)
plot(x,m1(1:11),'- ',x,m1(12:22),'+- ',x,m1(23:33),'*- ',x,m1(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第一组');
subplot(1,3,2)
plot(x,m2(1:11),'- ',x,m2(12:22),'+- ',x,m2(23:33),'*- ',x,m2(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第二组');
subplot(1,3,3)
plot(x,m3(1:11),'- ',x,m3(12:22),'+- ',x,m3(23:33),'*- ',x,m3(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第三组');

```

附录二十五：LingMin2.m

```

%两道工序的灵敏度分析
clc,clear
%设置上下限为 5%的灵敏度分析
x=-0.05:0.01:0.05;
TT1=[20,33,46,400,378,28,31,25];
for i=1:11
    T1(i,:)= [TT1(1:3)+TT1(1:3)*x(i),TT1(4:8)];%RGV 移动速度
    T1(i+11,:)= [TT1(1:3),TT1(4:5)+TT1(4:5)*x(i),TT1(6:8)];%CNC 加工速度
    T1(i+22,:)= [TT1(1:5),TT1(6:7)+TT1(6:7)*x(i),TT1(8)];%RGV 上下料速度
    T1(i+33,:)= [TT1(1:7),TT1(8)+TT1(8)*x(i)];%RGV 清洗熟料步骤速度
end
for i=1:44
    m1(i) = f402(T1(i,:));
end

TT2=[23,41,59,280,500,30,35,30];
for i=1:11
    T2(i,:)= [TT2(1:3)+TT2(1:3)*x(i),TT2(4:8)];%RGV 移动速度
    T2(i+11,:)= [TT2(1:3),TT2(4:5)+TT2(4:5)*x(i),TT2(6:8)];%CNC 加工速度
    T2(i+22,:)= [TT2(1:5),TT2(6:7)+TT2(6:7)*x(i),TT2(8)];%RGV 上下料速度
    T2(i+33,:)= [TT2(1:7),TT2(8)+TT2(8)*x(i)];%RGV 清洗熟料步骤速度
end
for i=1:44
    i
    m2(i) = f402(T2(i,:));
end

```

```

TT3=[18,32,46,455,182,27,32,25];
for i=1:11
    T3(i,:)= [TT3(1:3)+TT3(1:3)*x(i),TT3(4:8)];%RGV 移动速度
    T3(i+11,:)= [TT3(1:3),TT3(4:5)+TT3(4:5)*x(i),TT3(6:8)];%CNC 加工速度
    T3(i+22,:)= [TT3(1:5),TT3(6:7)+TT3(6:7)*x(i),TT3(8)];%RGV 上下料速度
    T3(i+33,:)= [TT3(1:7),TT3(8)+TT3(8)*x(i)];%RGV 清洗熟料步骤速度
end
for i=1:44
    i
    m3(i) = f402(T3(i,:));
end

subplot(1,3,1)
plot(x,m1(1:11),'- ',x,m1(12:22),'+- ',x,m1(23:33),'*- ',x,m1(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第一组');
subplot(1,3,2)
plot(x,m2(1:11),'- ',x,m2(12:22),'+- ',x,m2(23:33),'*- ',x,m2(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第二组');
subplot(1,3,3)
plot(x,m3(1:11),'- ',x,m3(12:22),'+- ',x,m3(23:33),'*- ',x,m3(34:44),'o- ');
xlabel('参数波动 5%');ylabel('成品数');
legend('RGV 移动速度','CNC 加工速度','RGV 上下料速度','RGV 清洗熟料步骤速度');
title('第三组');

```

附录二十六：附件二中的表格

Case_1_result.xls

Case_2_result.xls

Case_3_result_1.xls

Case_3_result_2.xls

两道工序 44 分刀 35 分刀 53 分刀结果表.xlsx

一道工序随机 20 次的结果.xlsx

两道工序随机 20 次的结果.xlsx