

基于排队论思想的智能 RGV 的动态调度优化策略

摘要

随着我国经济的迅速发展，各物料加工厂之间的竞争日趋激烈。智能 RGV 在各物料加工厂作业效率直接严重影响该厂的经济收益，因此，研究出合理的 RGV 调度优化模型是很有必要的。本文主要基于三种不同情况下建立合理的数学模型求解出不同情况下对应的 RGV 的动态优化调度策略。

情况一，只有一道工序的 RGV 的动态调度优化模型的确定。从排队论的思想及 0-1 整数规划出发，通过将 RGV 类比为服务台，CNC 类比为顾客进行模型的建立。以清洗完成后的熟料总量最大值为目标函数，以每班次连续作业时长、RGV 与各 CNC 的相对距离等为约束条件，建立单目标多约束的优化模型，利用计算机仿真法对 RGV 的调度模型进行遍历，得到三组不同数据下的 RGV 调度策略。以第一组数据为例，对该模型求解，得到该系统作业效率为 86.5%，清洗完成后的熟料个数为 365 个。详细调度策略见附录（二、三组数据的求解结果见支撑材料）。对应的求解算法见 4.1.8。最后，通过灵敏度分析，当加工时长变化 5% 时，系统作业效率误差不超过 0.8%，说明该模型具有很强的稳定性及实用性。

情况二，两道工序的 RGV 的动态调度优化模型的确定。由于各台 CNC 所载刀具未知，因此在情况一的基础上增加了蒙特卡洛模型进而对 CNC 的刀具加载种类进行随机取样。以清洗完成后的熟料总量最大值为目标函数，以刀具的总件数、每班次连续作业的时长等为约束条件，建立单目标多约束的优化模型，对该模型进行计算机仿真求解，得出不同数据所对应的目标函数值最大时各 CNC 刀具的加载种类与 RGV 调度策略。以第一组数据为例，编号 1、3、5、7 的 CNC 所载刀具在加工第一道工序时使用，编号 2、4、6、8 的 CNC 所载刀具在加工第二道工序时使用，并求得系统作业效率为 83.7%，对应的求解算法见 4.2.8。最后，通过灵敏度分析，当加工时长变化 5% 时，系统作业效率误差不超过 0.7%，说明该结果具有很强的稳定性。

情况三，在情况一、二的基础上，将 CNC 在加工过程中可能发生故障的因素考虑到模型中，增加新的约束情况。由于 CNC 发生故障的不确定性，本问采用随机取样法对发生故障的 CNC 进行多次模拟仿真，得出在存在故障的情况下系统的作业效率与 RGV 的调度策略。以情况一中的第一组数据为例，物料的最大数量为 356 个，故障概率为 0.01，根据期望，故障物料在三个左右，求得系统作业效率为 76.8%。

最后，本文对模型中的方法进行了科学的评价，并讨论了模型的优缺点，并对本文建立的模型进行了合理的推广。

关键词：智能 RGV 动态调度 排队论 0-1 规划 单目标多约束 蒙特卡洛模型

1 问题的背景与重述

1.1 问题的背景

近些年来,我国经济迅速发展,市场日趋复杂,竞争也日益激烈。为了在此环境生存下来,迎接来自国内外的诸多挑战和竞争,必须提高响应市场的能力,因此,就要求各企业用最低的成本、较高的机器使用率去加工产品。智能 RGV 的动态优化调度就为其中的一个例子,其已成为各物料场需要注意的重心。并且,调度优化是“在一定的时间内整合有限的资源去加工若干个工作件,以满足一个或多个既定目标”^[1]。所以,对智能 RGV 的动态调度做出研究和检验是非常有必要的。

1.2 问题的重述

一个智能加工系统由 8 台计算机数控机床(CNC)、1 辆轨道式自动引导车(RGV)、一条 RGV 直线轨道、1 条上料传送带、1 条下料传送带等附属设备组成。RGV 是一种无人驾驶、能在固定轨道上自由运行的智能车。它根据指令能自动控制移动方向和距离,并自带一个机械手臂、两只机械手爪和物料清洗槽,能够完成上下料及清洗物料等作业任务。

系统在工作时有以下具体三种情况:

(1) 物料只需一道加工工序时,每台 CNC 安装同样的刀具,物料可以在任一台 CNC 上加工完成;

(2) 物料需要两道加工工序时,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(其发生概率约为 1%)的情况,每次故障排除时间介于 10~20 分钟之间,故障排除后即刻加入作业序列。

要求在分别考虑一道工序和两道工序的物料加工作业情况下完成下列两项任务:

任务 1: 给出一般问题下的 RGV 动态调度模型以及相应的求解算法;

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,并将结果填入 EXCEL 表中。

2 模型的假设

(1) 假设处于相对面的两台 CNC 同段时间发出上下料信号,规定 RGV 优先处理台号为奇数的 CNC。

原因: 因为处于相对面的两台 CNC 在同段时间发出信号时,距离该段时间 RGV 所处的位置距离是一样的,为更好的建立模型,忽略该种情况的影响,做出了该假设。

(2) 假设忽略熟料在清洗槽中的实际清洗时间。

原因: 熟料在清洗槽中的实际清洗时间是很短的,远小于机械手将成料放到下料传送带上的时间,忽略该部分时间可更好的简化模型。

(3) 假设各传送带的速度都相同。

原因: 在上下料过程中,为使每台机器前的物料台最多只出现一个物料。

(4) 假设忽略 RGV 的机械臂上下料的旋转时间。

原因: 该部分时间与上下料时间相比是很短的,忽略该部分时间对模型的建

立及求解是很方便的。

(5) 假设每个物料台上至多至少均有一个物料。

原因：保证 RGV 工作的正常运行，利于简化计算。

(6) 假设 CNC 发生故障时立即故障。

原因：物料加工时间是一个时间段，在改时间段内，任意时刻都有可能发生故障，就会导致结果的多样性，作此假设，可以较为准确的求出模型的结果。

(7) 假设在情况一和情况二中的 CNC 在加工过程中不会发生故障。

原因：当 CNC 发生故障时，需人工进行处理，物料的加工也会受到影响，为使 RGV 在整体工作过程中所受的影响尽可能小，作此假设。

(8) 假设 RGV 匀速行驶。

原因：距离一定，当考虑速度是均匀的时候，可以简化该模型，便于研究计算。

3 主要符号说明

符号	符号说明
D_R	RGV 接收上下料信号时所处的位置
D_i	发出上下料信号的第 i 台 CNC 的位置
D_{Ri}	发出信号时的第 i 台 CNC 与该时刻 RGV 的位置间的距离
N	加工完成并且清洗完成的熟料的个数
Q_{yx}	有效的工作总时间
T	当前的工作时间
TT	工作总时间
x	表示加工物料第一道工序的 CNC 的总台数
y	表示加工物料第二道工序的 CNC 的总台数

*其他未标明符号在文中注明

4 RGV 的动态模型的建立

4.1 情况一：基于排队论思想的 RGV 动态调度模型

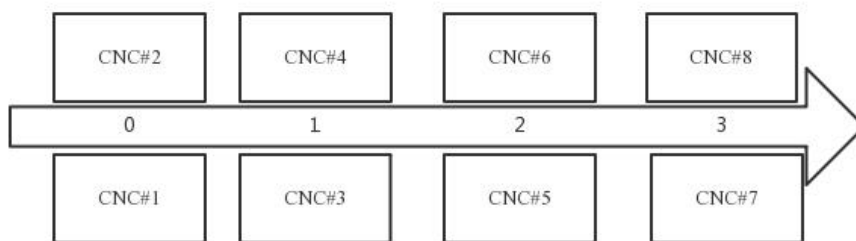
4.1.1 问题的分析

情况一要求 RGV 在符合所有约束条件的情况下对 8 个计算机数控机床进行上下料及清洗物料等作业任务，需要综合考虑以每班连续作业时间、RGV 第 j 次给 CNC 中的物料清洗完成的时刻即 RGV 第 $j+1$ 次给 CNC 上下料的时刻等约束条件。

首先，确定出该调度模型的目标函数，约束条件以及决策变量。然后建立以清洗完成后的熟料总量的最大值为目标函数的单目标多约束的优化模型，利用改进后的排队论的思想及动态调度规划和 0-1 规划模型的应用，完成题目中针对三种不同情况下的最大物料总量的目标。

4.1.2 RGV 的动态运动过程的分析

根据题目所给中所给的智能加工示意图，可将其简化如下：



图一 智能加工示意简化图

在上图中,规定初始 RGV 所在的位置为0,之后 RGV 的运动也只会 在 0,1,2,3 四个位置来回运动,每次移动的距离均可能发生改变,故有:

$$D_{Ri} = |D_i - D_R|, (i = 1, 2, 3, 4, 5, 6, 7, 8)$$

其中, D_R 表示 RGV 接收上下料信号时所处的位置, D_i 表示发出上下料信号的第 i 台 CNC 的位置, D_{Ri} 表示发出上下料信号的第 i 台 CNC 与同一时刻 RGV 的位置间的距离。

之后,考虑到 RGV 可能同时接受多个上下料信号,本文运用了先近后远的原则,找出了同段时间发出上下料信号的多台 CNC 距离该段时间 RGV 所处位置的 距离,找出距离最小值所对应的第 i 台 CNC,即 RGV 在下一时刻会前往到 该处进行上下料作业。

而且,给定变量 N ,初始值为 0,每当 RGV 清洗完成一个物料时,均有 $N = N + 1$,直至时间达到限定时间 N 才结束。

4.1.3 情况一下的 RGV 动态调度模型的确定

(1) 排队论的相关原理

排队论又称作随机服务系统理论,为了研究服务系统中排队现象的随机规律 而创立的学科,广泛应用于运输、生产、库存等各项资源分配管理的随机服务系 统中。

一般的排队系统由顾客源、队列、服务机构组成。顾客经由顾客源到达列队 时,依照一定的排队规则进行等待。等待结束后,服务机构会按照一定的排队顺 序来对顾客进行服务,当服务完毕时顾客随即退出队列,排队系统的一般表示如 下所示:



图二 排队论的简要流程图

对于排队论的输入过程,顾客到达的概率分布主要有定长输入、泊松流、K 阶爱尔兰分布,由于顾客到达的模式与加工的在制品完全不同,因此不做过多分 析。

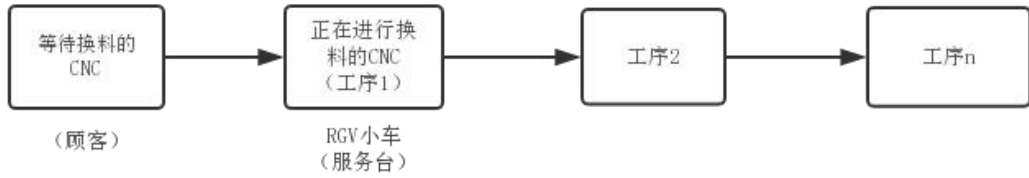
(2) RGV 动态调度引入排队论

排队论是探索排队系统在不同条件下产生的排队现象,从运筹学的角度,排

队论算法就是建立反映随机性的数学模型。

排队论算法适用于许多问题，车间生产系统也同样适用。所以在分析 RGV 的动态调度问题时，本文参考排队论的思想，引出了改进后的排队论的模型加以应用。

在本文中，将 RGV 类比成排队论模型中的服务台，将八台 CNC 类比成顾客，在此需要稍作说明，与原先的排队论不同的是，在 RGV 的动态调度过程中，八台 CNC 并不是只排队一次，而是在八个小时之内一直处于排队状态，只要物料一加工完成，就会向 RGV 发出上下料加工信号。另外，在改进后的排队论中，RGV 与各 CNC 的运动状态与排队论中的服务台和顾客的运动状态正好相反。所以，将排队论的基本模型引入到 RGV 动态调度系统中，利用改进后的排队论并使用运筹学方法可以计算出车间内的理想状态。



图三 基于排队论的 CNC 加工物料流程图

(3) 0—1 整数规划模型的建立

在考虑八台 CNC 是否需要 RGV 上下料时，本文考虑使用 0-1 整数规划模型，具体引入如下：

$$CNCi\# = \begin{cases} 1, & \text{当 } CNCi\# \text{ 向 } RGV \text{ 发出上下料信号时} \\ 0, & \text{当 } CNCi\# \text{ 正在加工无料时} \end{cases}$$

$$(i = 1, 2, 3, 4, 5, 6, 7, 8)$$

其中， $CNCi\#$ 表示第 i 台 CNC 是否发出上下料的信号，如果发出信号，则为 1，反之则为 0。

又考虑到 RGV 只会处于两种状态，一种是空闲状态，另一种就是工作状态，于是，使用的 0-1 整数规划如下：

$$Z_t = \begin{cases} 1, & RGV \text{ 处于工作状态} \\ 0, & RGV \text{ 处于空闲状态} \end{cases}$$

其中， Z_t 表示 RGV 的所处状态，刚开始时，RGV 是会收到八台 CNC 的上料信号，此时 Z_t 的值为 1，如果在帮第八台 CNC 上完料时第一台加工的物料都没熟， Z_t 会立即转为 0，直到出现熟料的 CNC 发出上下料信号 Z_t 才会转为 1，继续下一步的工作。

4.1.4 目标函数的确定

对于本文对智能 RGV 的研究是为了使其效率更高，更具有实用性，使得使用该机器的工厂能够获得最大利润，因此在满足一切条件与限制的情况下，希望使被加工并清洗的物料的个数尽可能的多，既有：

$$\max N,$$

其中 N 为各机器整个运行过程中加工完成并且清洗完成的熟料的个数。

4.1.5 决策变量和约束条件的确定

(1) 基于每组对智能加工系统的数据统计可知，每班次连续作业的时间有明确的限制条件，每班次连续作业时间为 8h，固有：

$$T_{\max} = 8h,$$

其中 T_{\max} 表示每班次连续作业的时长。

(2) 当一台 CNC 加工物料时，规定加工后一道物料必须在前一道物料加工完成及清洗完成之后（忽略报废的物料），即有：

$$S_{jq} < R_{(j+1)},$$

其中， S_{jq} 表示 RGV 第 j 次给 CNC 中的物料清洗完成的时刻， $R_{(j+1)}$ 表示 RGV 第 $j+1$ 次给 CNC 上下料的时刻。

(3) 每次上下料的过程中，必须保证一个物料只能被一台 CNC 加工，固有：

$$\sum_{j=1}^n \omega_{ji} = 1 \quad (i=1,2,3,4,5,6,7,8),$$

其中， ω_{ji} 表示当第 j 个物料被运送至第 i 台 CNC 时，其值为 1；否则，其值为零。

(4) 每当 CNC 需要上下料时，需保证每个上料传送带上都必须有物料，固有：

$$t_j > t_s.$$

其中， t_j 表示为八台 CNC 发出的第 j 个上下料信号的时刻， t_s 表示为发出第 j 个上下料信号前该台 CNC 所对应的上料台上料完的时刻^[2]。

4.1.6 最终优化模型的确立

由上文讨论可知，以各机器整个运行过程中加工完成并且清洗完成的熟料的个数的最大值为目标函数，以每班次连续作业时间、RGV 第 j 次给 CNC 中的物料清洗完成的时刻、RGV 第 $j+1$ 次给 CNC 上下料的时刻等为决策变量，并以每班次连续作业的时长等为约束条件，建立了如下的智能 RGV 的动态调度优化模型：

目标函数： $\max N$

$$\text{约束条件: } \begin{cases} T_{\max} = 8h \\ S_{jq} < R_{(j+1)} \\ \sum_{j=1}^n \omega_{ji} = 1 \\ t_j > t_s \end{cases}$$

其中 T_{\max} 表示每班次连续作业的时长， S_{jq} 表示 RGV 第 j 次给 CNC 中的物料清洗完成的时刻， $R_{(j+1)}$ 表示 RGV 第 $j+1$ 次给 CNC 上下料的时刻， ω_{ji} 表示当第

j 个物料被运送至第 i 台 CNC 时，其值为 1；否则，其值为零。

4.1.7 所建立的模型的工作效率的求解

对于 RGV 来说，其对于八个 CNC 的总的工作时间是确定的，即：

$$TT = \sum_{j=1}^n t_j + t_q \cdot n + t_\beta \cdot n + t_{yz} + t_{dz},$$

其中，TT 表示每班次连续作业的时长，其值为 64h， $\sum_{j=1}^n t_j$ 表示 n 个物料总加工所需要的时间， t_q 表示各物料的清洗时间， t_β 表示一次上下料的时间， t_{yz} 表示 RGV 移动所消耗的总时间， t_{dz} 表示总的等待时间。

而对于整个机器运行过程，只有被 CNC 加工所消耗的时间才算真正有效的时间，即

$$Q_{yx} = \sum_{i=1}^8 t_i \cdot n_i,$$

其中 Q_{yx} 表示有效的工作总时间， t_i 表示第 i 台 CNC 加工一个物料所需要的时间， n_i 表示第 i 台所加工的物料的总件数。

进而可得在该动态调度优化模型下的系统的作业效率为：

$$\eta = \frac{Q_{yx}}{TT}$$

4.1.8 求解的算法的确定

(1) 算法的主要思想

基于排队轮等待制度下的优先服务原则，RGV 在未作出判断时对接收到的所有信号采取优先处理距自身距离最近的 CNC 的原则。基于优先服务原则，采取计算机仿真的方法在该原则下对整个调度进行模拟，当得满足所有约束条件且目标函数值最大，停止计算，得出 RGV 调度优化策略。

(2) 算法步骤

经过对该智能加工系统进行的简要分析，决定采用基于改进的排队论模型及 0-1 整数规划模型的计算机仿真法对 RGV 的动态调度优化进行遍历仿真的具体步骤如下：

Step1:首先，建立一维坐标系并对 RGV 与 CNC 的坐标进行初始化。之后建立 CNC 的需求矩阵 Z 并将其元素均初始化为 1，对已工作时间 T 初始化为 0。

(Z 中元素为 1 代表对应的 CNC 需要 RGV 前来处理，为 0 则代表不需要。)

Step2:计算 RGV 与各个 CNC 之间的距离并生成距离矩阵 dri ，同时对接收到的信号进行处理，对所有需求为 1 的 CNC 与 RGV 现处位置的距离进行判断，筛选出 dri 中的距离最小值。并且， T 需时刻更新，若 T 在任意时刻达到作业时间，则跳转到 Step6。

Step3:RGV 产生移动指令，更新 RGV 的坐标位置，根据接收到的信号进行上下料等操作。并且， T 需时刻更新，若 T 在任意时刻达到作业时间，则跳转到 Step6。

Step4:在需求矩阵 Z 中，对已处理的 CNC 的需求进行更新，将 1 更新为 0。且在下一次接收到该 CNC 需求信号前将 RGV 与该 CNC 的距离视为无穷大，意为不需要到达，更新距离矩阵。并且， T 需时刻更新，若 T 在任意时刻达到作业时间，则跳转到 Step6。

Step5:接收到新一轮需求信号,将发出信号的 CNC 的需求更新为 1,并且, T 需时刻更新,若 T 在任意时刻达到作业时间,则跳转到 Step6。若 T 未饱和,计算新的距离矩阵开始新一轮工作,返回 Step2。

Step6:T 达到饱和,工作时间结束,结束循环。

4.2 情况二:基于排队论思想的 RGV 动态调度模型

4.2.1 问题的分析

情况二在情况一的的基础上将物料的加工作业情况由一道工序改为两道工序,并且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。在条件的改动下,要求 RGV 在符合所有约束条件的情况下对 8 个计算机数控机床进行不同工序的上下料及清洗物料等作业任务,需要综合考虑各约束条件。

首先,确定出条件改变后该调度模型的目标函数,约束条件以及决策变量。然后建立以物料总量的最大值为目标函数的单目标多约束的优化模型,利用改进后的排队论的思想及动态调度规划和 0-1 规划模型的应用,完成题目中针对情况二的最大物料总量的目标。

4.2.2 RGV 的动态运动过程的分析

由情况一可知,规定初始 RGV 所在的位置为 0,之后 RGV 的运动也只会 0,1,2,3 四个位置来回运动,每次移动的距离均可能发生改变,故有:

$$D_{Ri} = |D_i - D_R|, (i = 1, 2, 3, 4, 5, 6, 7, 8)$$

其中, D_R 表示 RGV 接收上下料信号时所处的位置, D_i 表示发出上下料信号的第 i 台 CNC 的位置, D_{Ri} 表示发出上下料信号的第 i 台 CNC 与同一时刻 RGV 的位置间的距离。

之后,考虑到 RGV 可能同时接受多个上下料信号,并且,收到的多个信号具体的指令也不一定相同,对于不同信号的处理方式,会在后面的模型的建立中做详细介绍。而且,给定变量 N ,初始值为 0,每当 RGV 清洗完成一个物料时,均有 $N = N + 1$,直至时间达到限定时间 N 才结束。

4.2.3 情况二下的 RGV 动态调度模型的确定

考虑到情况二较之情况一所增加的条件,于是决定在情况一所建立的模型之上,引入蒙特卡洛模型来随机确定八台 CNC 的分配问题。

4.2.3.1 蒙特卡洛模型的引入

(1) 蒙特卡洛模型的基本思想

为了求解数学、物理、工程技术以及管理等方面的问题,首先建立一个概率模型或随机过程,利用相应的参数,得到某些问题(如概率分布或数学期望等问题)的解;然后通过对模型或过程的观察或抽样试验来计算所求参数的统计特征,并用算术平均值作为所求解的近似值。

(2) 基于情况二下的 RGV 动态调度下使用的蒙特卡罗方法的步骤

①依据概率分布 $\phi(x)$ 不断生成随机一组数值,记为矩阵 A 。

由于随机数性质,每次生成的一组 A 都是不确定的,为区分起见,我们可以给生成的 A 赋予下标。如 A_i 表示生成的第 i 组 A 。生成了多少组 A ,再结合情况一下的模型的建立过程,就可以计算出多少组 N 的值。

②将这些 N_i 的值进行比较，筛选出最大的 N_i 。

③到达停止条件后退出

常用的停止条件有两种，一种是设定最多生成多个 N_i ，数量达到后即退出，另一种是检测计算结果与真实结果之间的误差，当这一误差小到某个范围之内时退出。

④误差分析

该方法得到的结果是随机变量，具有偶然性，因此，在给出点估计后，还需要给出此估计值的波动程度及区间估计。严格的误差分析首先要从证明收敛性出发，再计算理论方差，最后用样本方差来替代理论方差^[3]。

4.2.3.2 对 RGV 在接收多种信号的情况下的优先判断说明

首先需要说明的是，在分析以下五种情况时，只考虑在某一时刻 RGV 所接收的所有的信号，而忽略后来任一时刻的各 CNC 所发出的信号。因为当后来某一时刻有新信号加入时，可以重置该时刻从该时刻判断，又所有情况均在所分析的五种情形内，因此，在分析所有情形时，无需考虑新信号加入的情形，但其为动态调度，实际中需在任一时刻对所接收到的上下料信号进行重置。

对于某一时刻的 RGV 在接收多种上下料信号时存在如下三种主要关系：

(1) RGV 只收到一个上下料信号

该 RGV 只收到一个上下料信号又可分为：

①RGV 只收到一个完成第一道工序后的 CNC 发出的信号②RGV 只收到一个完成第二道工序后的 CNC 发出的信号。

在该两种情况下，只需 RGV 前往到该处完成上下料任务。

(2) RGV 只收到多个单一的上下料信号

该 RGV 只收到多个单一上下料信号又可分为：

①RGV 只收到多个完成第一道工序后的 CNC 发出的信号②RGV 只收到多个完成第二道工序后的 CNC 发出的信号。

在该两种情况下，只需 RGV 遵循如同情况一时的先近后远原则。

(3) RGV 收到混合上下料信号

该 RGV 收到多个混合的上下料信号又可分为：

①RGV 收到等数量的完成第一道工序后的 CNC 发出的信号和完成第二道工序后的 CNC 发出的信号②RGV 收到不等数量的完成第一道工序后的 CNC 发出的信号和完成第二道工序后的 CNC 发出的信号。

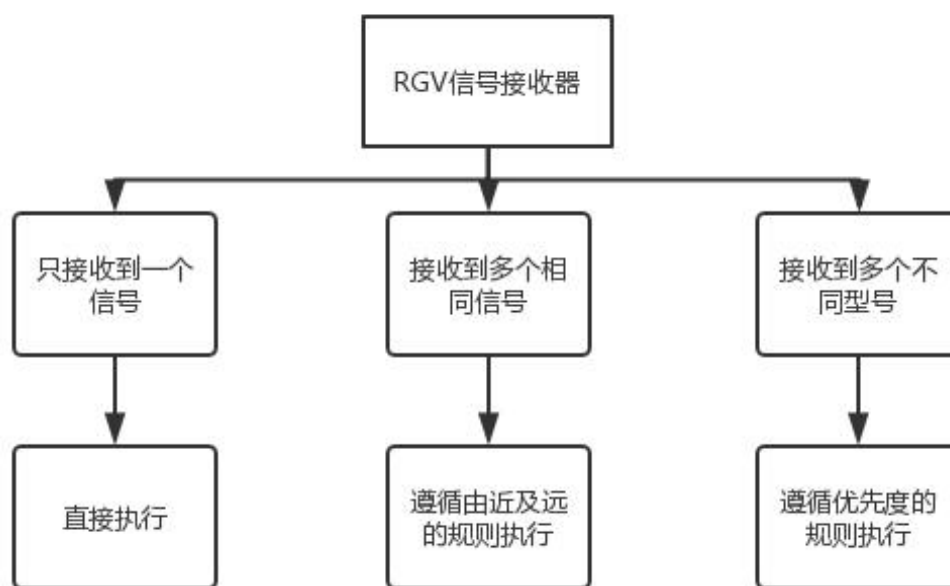
当收到等数量的完成第一道工序后的 CNC 发出的信号和完成第二道工序后的 CNC 发出的信号时，RGV 会先去到第一道工序完成后的 CNC 发出信号并距离此时 RGV 所在处最近的位置完成上下料工作，之后，需前往距离最近的完成第二道工序后的 CNC 发出的信号的位置为该 CNC 上下料并对熟料进行清洗工作，清洗完成后，重复以上行为，直至作业完成。

当收到不等数量的完成第一道工序后的 CNC 发出的信号和完成第二道工序后的 CNC 发出的信号时，又可具体分为两种情况：

①当 RGV 收到的完成第一道工序后的 CNC 发出的信号的数量大于收到的完成第二道工序后的 CNC 发出的信号的数量时，前期同（3）—①中步骤，直至所剩信号均为完成第一道工序后的 CNC 发出的信号，之后只需重复（1）—①或（2）—①中的步骤即可。

②当 RGV 收到的完成第一道工序后的 CNC 发出的信号的数量小于收到的完成第二道工序后的 CNC 发出的信号的数量时，前期同（3）—①中步骤，直至所剩信号均为完成第二道工序后的 CNC 发出的信号，之后只需重复（1）—②或（2）—②中的步骤即可。

根据以上判断，做出的简要流程图如下：



图四 RGV 接收信号流程图

4.2.4 目标函数的确定

本文对智能 RGV 的研究是为了使其效率更高，更具有实用性，使得使用该机器的工厂能够获得最大利润，因此在满足一切条件与限制的情况下，希望使被加工并清洗的物料的个数尽可能的多，既有：

$$\max N,$$

其中 N 为各机器整个运行过程中加工完成并且清洗完成的熟料的个数。

4.2.5 决策变量和约束条件的确定

在分析情况一的基础之上，由于新增的条件限制，情况二的约束条件较之情况一部分会有所不同。

（1）在对八台 CNC 进行不同工序的分配时，如何分配刀具是非常重要的，因为刀具的分配直接影响目标 N 的大小，设完成第一道工序的 CNC 有 x 台，完成第二道工序的 CNC 有 y 台，则其满足：

$$x + y = 8,$$

其中, x 表示加工物料第一道工序的 CNC 的总台数, y 表示加工物料第二道工序的 CNC 的总台数。

(2) 基于每组对智能加工系统的数据统计可知, 每班次连续作业的时间有明确的限制条件, 每班次连续作业时间为 8h, 固有:

$$T_{\max} = 8h,$$

其中 T_{\max} 表示每班次连续作业的时长。

(3) 在加工一个物料的总过程中, 加工完成第一道工序的时刻必须小于该物料刚开始加工第二道工序的时刻, 固有:

$$t'_{nyj} < t_{nej},$$

其中, t'_{nyj} 表示第 n 个物料加工完成第一道工序的时刻, t_{nej} 表示第 n 个物料开始加工第二道工序的时刻。

(4) 每次上下料的过程中, 必须保证一个物料的第一道工序或第二道工序只能被一台 CNC 加工, 固有:

$$\begin{cases} \sum_{j=1}^n \omega_{jx_i} = 1 \\ \sum_{j=1}^n \omega_{jy_i} = 1 \end{cases}.$$

其中, ω_{jx_i} 表示当第 j 个物料被运送至某台 CNC 进行第一道工序的加工时, 其值为 1; 否则, 其值为零。 ω_{jy_i} 表示当第 j 个物料被运送至某台 CNC 进行第二道工序的加工时, 其值为 1; 否则, 其值为零。

4.2.6 最终优化模型的确立

综上所述, 以各机器整个运行过程中加工完成并且清洗完成的熟料的个数的最大值为目标函数, 以各类型刀具所分配的数量、第 n 个物料加工完成第一道工序的时刻、第 n 个物料开始加工第二道工序的时刻等为决策变量, 并以刀具的总件数等为约束条件, 建立了如下的智能 RGV 的动态调度优化模型:

目标函数: $\max N$

$$\text{约束条件: } \begin{cases} x + y = 8 \\ T_{\max} = 8 \\ t'_{nyj} < t_{nej} \\ \sum_{j=1}^n \omega_{jx_i} = 1 \\ \sum_{j=1}^n \omega_{jy_i} = 1 \end{cases}$$

其中 T_{\max} 表示每班次连续作业的时长, t'_{nyj} 表示第 n 个物料加工完成第一道工序的时刻, t_{nej} 表示第 n 个物料开始加工第二道工序的时刻。

4.2.7 所建立的模型的工作效率的求解

对于 RGV 来说，其对于八个 CNC 的总的工作时间是确定的，即：

$$TT = \sum_{j=1}^n (t_{1j} + t_{2j}) + t_q \cdot n + (t_{\beta 1} + t_{\beta 2}) \cdot n + t_{yz} + t_{dz},$$

其中，TT 表示每班连续作业的时长，其值为 64h， $\sum_{j=1}^n (t_{1j} + t_{2j})$ 表示 n 个物料第一道工序加工与第二道工序加工完成所需要的总时间， t_q 表示各物料的清洗时间， $t_{\beta 1} + t_{\beta 2}$ 表示一个物料两次上下料的时间， t_{yz} 表示 RGV 移动所消耗的总时间， t_{dz} 表示总的等待时间。

而对于整个机器运行过程，只有被 CNC 加工所消耗的时间才算真正有效的时间，即

$$Q_{yx} = \sum_{i=1}^x t_{li} \cdot n_{li} + \sum_{j=1}^y t_{2j} \cdot n_{2j},$$

其中 Q_{yx} 表示有效的工作总时间， t_{li} 表示第 i 台 CNC 加工一个物料第一道工序所需要的时间， n_{li} 表示第 i 台所加工第一道工序的物料总件数。 t_{2j} 表示第 j 台 CNC 加工一个物料第二道工序所需要的时间， n_{2j} 表示第 j 台所加工第二道工序的物料总件数。

进而可得在该动态调度优化模型下的系统的作业效率为：

$$\eta = \frac{Q_{yx}}{TT}$$

4.2.8 求解的算法的确定

(1) 算法的主要思想

利用蒙特卡洛随机数取样的方式生成 8 个 0 或者 1 的数来模拟 CNC 的刀具分布，每次随机生成一种刀具分布便计算这种在分布情况下的最大清洗熟料量，往复循环多次，输出最大的一个最大清洗熟料量所对应的刀片分布方式与在这种方式下的 RGV 调度策略

(2) 算法步骤

Step1: 生成随机 CNC 刀具加载种类矩阵，其元素为 1 或者 2，1 代表该对应的 CN 加载的刀具为第一道工序服务，2 代表该对应的 CNC 加载的刀具为第二道工序服务

Step2: 建立一维坐标系并对 RGV 与 CNC 的坐标进行初始化。之后建立 CNC 的需求矩阵 Z 并将其元素均初始化为 1，对已工作时间 T 初始化为 0。（Z 中元素为 1 代表对应的 CNC 需要 RGV 前来处理，为 0 则代表不需要。）

Step3: 计算 RGV 与各个 CNC 之间的距离并生成距离矩阵 dri，同时对接收到的信号进行处理，对所有需求为 1 的 CNC 与 RGV 现处位置的距离进行判断，筛选出 dri 中的距离最小值。并且，T 需时刻更新，若 T 在任意时刻达到作业时间，则跳转到 Step6。

Step4: RGV 产生移动指令，更新 RGV 的坐标位置，根据接收到的信号进行上下料等操作。并且，T 需时刻更新，若 T 在任意时刻达到作业时间，则跳转到 Step6。

Step5:在需求矩阵 Z 中, 对已处理的 CNC 的需求进行更新, 将 1 更新为 0. 且在下次接收到该 CNC 需求信号前将 RGV 与该 CNC 的距离视为无穷大, 意为不需要到达, 更新距离矩阵。并且, T 需时刻更新, 若 T 在任意时刻达到作业时间, 则跳转到 Step6。

Step6:接收到新一轮需求信号, 将发出信号的 CNC 的需求更新为 1, 并且, T 需时刻更新, 若 T 在任意时刻达到作业时间, 则跳转到 Step6。若 T 未饱和, 计算新的距离矩阵开始新一轮工作, 返回 Step2。

Step7: T 达到饱和, 工作时间结束, 结束循环。

Step8:将上述步骤重复 100 次, 选出目标函数值最大的。

4.3 基于情况三引入故障参数的动态调度模型的建立

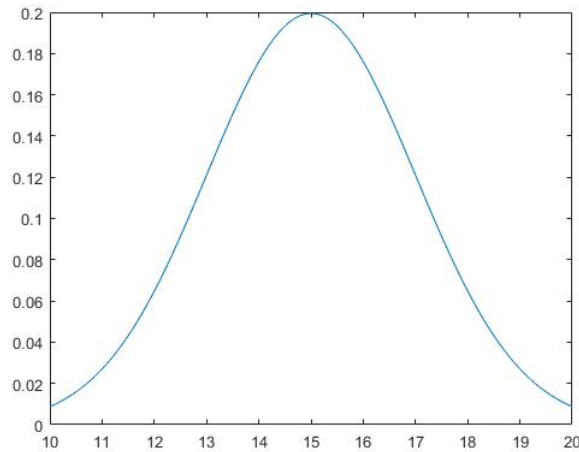
在情况一、二的基础上, 由于新增 CNC 故障的发生概率约为 1%的条件, 故情况三引入了故障参数 P , 并增加决策变量:

故障概率 $P=0.01$

故障排除时间 G

G 不是一个准确确定的分布时间, 为此本文参考了维修时间分布规律函数的估计^[4], 选择故障排除时间遵循正态分布:

$$G \sim N(\mu, \sigma^2)$$



图五 维修时间概率密度分布函数

将该故障概率引入仿真运算中, 运用计算机仿真法对该种情况下 RGV 的动态调度优化进行遍历仿真, 求得最终结果。

5 RGV 的动态调度模型的求解

5.1 基于情况一的动态模型的求解

5.1.1 模型的求解

以题目中表 1 的第一组数据为例, 本文利用计算机仿真, 将表一第一组数据的所有参数, 带入到上文所建立的优化模型, 其中:

表一 第一组的参数数据

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

注：时间单位均为秒（s）。

使用 MATLAB 编程将各参数值带入实现计算机仿真模拟复现并记录 RGV 的调度策略，以下是节选部分 RGV 调度策略的结果展示：

表二 基于情况一的 RGV 的调度策略（部分）

加工序号	CNC 编号	上料时间	下料时间	加工序号	CNC 编号	上料时间	下料时间
1	1	28	662	11	3	791	1425
2	2	59	718	12	4	847	1481
3	3	107	791	13	5	920	1554
4	4	138	847	14	6	976	1610
5	5	186	920	15	7	1049	1683
6	6	217	976	16	8	1105	1739
7	7	265	1049	17	1	1296	1930
8	8	296	1105	18	2	1352	1986
9	1	662	1296	19	3	1425	2059
10	2	718	1352	20	4	1481	2115

并得到：加工总数 $N=356$ ，作业效率 $\eta_1=86.5\%$

5.1.2 结果的的检验

对表一第一组数据得出的结果进行灵敏度分析，提高其 10%的加工时长，观测结果的改变，得到新的结果：加工总数 $N'=328$ ，作业效率 $\eta'=87.3\%$

对比 η 与 η' 的结果，两者相差不大，说明模型通过检验，拟合度良好，算法在一定程度上有效，且模型实用型强。

5.1.3 情况一的结果分析

观测完整的情况一的调度策略（见支撑材料），本文发现 CNC 的编号在情况一中呈现周期性变化。对此，本文基于该特殊现象做深度研究后得出结论，其原因是从初始时刻之后，小车按照一定顺序对八个 CNC 进行上料后，所有机器都处于加工状态，RGV 出现等待现象，直到再次以相同的顺序进行上下料作业，周而复始。并且，这一特殊现象从侧面佐证了此模型的实用型强的特点。

5.2 基于情况二的动态模型的求解

5.2.1 模型的求解

（1）刀具的分布策略的分析

以表中三组数据为例，利用蒙特卡洛算法进行无数次的随机取样，从样本中选取清洗完成后的熟料最多的样本，作为刀具分配方式。

利用 MATLAB 计算得出，使用第一组数据得出的两步工序的刀具比为 4:4，第二组数据为 3:5，第三组数据为 5:3。

表三 各工序所用时间与各刀具之比的关系

组数	各工序之比	刀具之比
第一组	400:378	4:4
第二组	280:500	3:5
第三组	455:182	5:3

观察表中数据得出刀具分布策略：当各工序所用时间接近 1:1 时，刀具采用平均分布的策略，当各工序所用时间差异过大时，刀具采用权重分布的策略，且权重比 \approx 各工序之比。

(2) 双工序下的 RGV 小车的调度策略

在情况一的基础上，对 RGV 小车的调度策略进行修正，得到双工序下的 RGV 小车的调度策略，使用 MATLAB 编程实现计算机仿真模拟复现并记录 RGV 小车的调度策略，以下是节选部分 RGV 小车调度策略的结果展示：

表四 情况二 RGV 小车的调度策略（部分）

加工序号	工序 1CNC 编号	上料	下料	工序 2CNC 编号	上料	下料
1	3	76	524	4	527	927
2	5	124	572	6	575	975
3	7	172	620	8	979	1379
4	1	502	976	2	975	1375
5	3	524	972	4	1023	1423
6	5	572	1020	6	1071	1471
7	7	620	1068	8	1071	1836

得到：加工总数 $N=245$ ，作业效率 $\eta=82.7\%$

5.2.2 结果的检验

对表第一组数据得出的结果进行灵敏度分析，提高其 10%的加工时长，观测结果的改变，得到新的结果：加工总数 $N'=220$ ，作业效率 $\eta'=81.0\%$

对比 η 与 η' 的结果，发现两者相差不大，说明模型通过检验，拟合度良好，算法在一定程度上有效，且模型实用性强。

5.2.3 情况二的结果分析

观测完整的情况二的调度策略（见支撑材料），本文发现 CNC 的编号在情况二中不再呈现周期性变化。对此本文对此特殊现象做深度研究后得出结论，其原因就是由于工作台刀具的不同导致 RGV 小车不能随意的加工，所以工作效率要比情况一低，而结果也证明了这种分析的正确性。这一特殊现象从侧面佐证了此模型适应能力强的特点。

5.3 基于情况三的动态模型的求解

5.3.1 情况三模型的求解

在情况一二的基础上，对 RGV 小车的调度策略进行修正，得到单双工序下的含有故障参数 RGV 小车的调度策略。使用 MATLAB 编程实现计算机仿真模拟复现并记录 RGV 小车的调度策略，以下是节选部分含有故障参数的 RGV 小车调度策略的结果展示：

表五 情况三第一组的故障（部分）

故障物料序号	故障 CNC 编号	故障开始时间	故障结束时间
44	4	3785	4685
209	8	18274	19174

5.3.2 情况三的检验

因为在情况三的情景下，物料故障具有一定随机性，每次模拟的最大物料数和作业效率各不相同。因此只要得出的数据和情况一二对比，在一定范围内，就认定为合理。

表五为某次程序运行所得出的部分结果，其中物料故障数为 2，在情况一中，物料的最大数量为 356 个，故障概率为 0.01，根据期望，故障物料在三个左右。对比发现，此次运行结果可靠。并得到加工总数 $N=316$ ，作业效率 $\eta=76.8\%$ 。

5.3.3 情况三的结果分析

观测完整的情况三的调度策略（见支撑材料），本文发现在引入故障参数后，大量具有重复性的精准良好结果均证明：较情况一二来说，系统的作业效率大大下降，这和实际情况相符。说明此模型在实际情况下，更具有一般性。

6 模型的评价与推广

6.1 模型的优点

- （1）模型利用类比推理的思想，以排队来类比生产作业，使复杂的模型简单化。
- （2）本文建立的模型引入了多种思想，对于解决实际问题有着一定的帮助，使模型简单易懂，容易求解。

6.2 模型的缺点

- （1）对于一个完全 NP 问题，拥有的可行解是无穷的，无法遍历的。模型有可能会陷入局部最优的情况。
- （2）系统作业的参数值组数只有三组，在一定程度上无法说明模型的实用型。
- （3）情况三模型存在随机性，模型自身无法检验并剔除极值。

6.3 模型的推广

首先，本文所建立的模型与实际情况相符合，具有一定的指导性。同时可针对模型进行更深层次的分析，如研究系统作业效率的影响因素，发成故障对整体作业的影响。

其次本文所提出的基于排队论的改良算法，并采用计算机仿真的方法，极大程度的再现了生产车间的真实情况。对于判断 RGV 每一步的决策有着一定程度的帮助。

最后对模型进行适当推广，本文建立的模型还可用于求解生产线上的生产分配问题、车间调度问题、流水线作业问题等。

参考文献:

[1]Baker K R.Introduction to Sequencing and Seheduling, New York:Wiley,1974.

[2]陈华, 孙启元. 基于 TS 算法的直线往复 2-RGV 系统调度研究[J].工业工程与管理.2015, 20 (05): 80-88.

[3]张磊, 蒙特卡洛模型, [https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%A8%A1%E5%9E%8B/9156350?fr=aladdin#reference-\[1\]-1252319-wrap](https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%A8%A1%E5%9E%8B/9156350?fr=aladdin#reference-[1]-1252319-wrap), 2018 年 9 月 15 日。

[4]曾海军, 陆中, 戎翔, 孙有朝. 基于遗传算法的维修时间分布参数非线性最小二乘估计[J]. 南京航空航天大学学报, 2013, 45 (06):859-864.

[5]陈杰. 基于排队论算法的车间在制品管理系统设计与实现[D]. 浙江理工大学, 2017.

[6]吴涛. 动态规划算法应用及其在时间效率上的优化[D]. 南京理工大学, 2008.

[7]张梦雨, 黄河, 杨以雄, 章娟. 基于 0-1 规划的成衣制造流水线平衡建模与仿真[J]. 毛纺科技, 2018, 46 (05):46-51.

附录一：基本核心代码，数据为第一组数据，本文的所有代码均以此为基础

<pre> clear;close;clc %xr RGV 的位置 %xi CNC(i) #的位置 %z CNC(i) #的需求指令，z=1 则 CNC(i) # 需要 RGV 前来上下料，zi=0，则 CNC(i) #不 需要 RGV 前来上下料 %manглу_cnc 处于工作状态下的 cnc %dri CNC(i) #与 RGV 之间的距离 %n 物料编号 %tsx 物料的上下料所需时间 %tsn 物料 n 的上料时刻 %tjwn CNC 对物料 n 的加工完成时刻 %txn 物料 n 的下料时刻 %y RGV 一次移动的单位 %ty RGV 移动时间，ty=t1 或 t2 或 t3 或 t4 %t0 RGV 移动 0 个单位所需要的时间，t0=0 %t1 RGV 移动 1 个单位所需要的时间 %t2 RGV 移动 2 个单位所需要的时间 %t3 RGV 移动 3 个单位所需要的时间 %tq 物料清洗时间 %TT 总工作时间 %tj CNC 加工时间 %T 循环更新时间 %cnc 物料的加工场所 %建立一维坐标系 x_rgv=0;%RGV 的位置初始化 x_cnc=[0 0 1 1 2 2 3 3];%对 CNC 的位 置进行初始化 z=[1 1 1 1 1 1 1 1]; n=0;t0=0;t1=20;t2=33;t3=46;tj=560 ; tsx1=28;tsx2=31;tq=25;ts=[];tx=[] ;cnc=[]; T=0;TT=8*3600; for i=1:8 manглу_cnc=find(z==0);%不需要 RGV 上 下料的 CNC, 正在加工。 dri=abs(x_cnc-x_rgv);%距离矩阵 dri(manглу_cnc)=inf;%将目前不需要 RGV 处理的 CNC 距离设置为空 y=min(dri);%RGV 将要移动的距离，选择最 小的 </pre>	<pre> switch(y)%选择移动时间 case 0 ty=t0; case 1 ty=t1; case 2 ty=t2; case 3 ty=t3; end bianhao=find(dri==y)%与 RGV 距离最小 的 CNC 的标号,可能有两个 cnc_i=min(bianhao)%移动到的 CNC x_rgv=x_cnc(cnc_i);%RGV 位置更新 A=mod(cnc_i,2); switch(A) case 1 tsx=tsx1; case 0 tsx=tsx2; end T=T+ty;n=n+1;cnc(n)=cnc_i; ts(n)=T+tsx;%物料 n 的上料时刻 z(cnc_i)=0;tjw(n)=ts(n)+tj; T=ts(n);%时间轴更新 end T=tjw(1);%重置时间 B=[]; for i=1:8 B(i)=tjw(i) end z(1)=1;N=[inf inf inf inf inf inf inf inf]; while(T<=TT) manглу_cnc=find(z==0);%不需要 RGV 上下料的 CNC, 正在加工。 dri=abs(x_cnc-x_rgv);%距离矩阵 dri(manглу_cnc)=inf;%将目前不需 要 RGV 处理的 CNC 距离设置为空 C=isequal(dri,N); while(C==1) T=min(B);F=find(B==T);z(F)=1; </pre>
---	---

<pre> manglu_cnc=find(z==0);%不需要 RGV 上 下料的 CNC,正在加工。 dri=abs(x_cnc-x_rgv);%距离矩阵 dri(manglu_cnc)=inf;%将目前不需 要 RGV 处理的 CNC 距离设置为空 break end y=min(dri);%RGV 将要移动的距离,选 择最小的 switch(y)%选择移动时间 case 0 ty=t0; case 1 ty=t1; case 2 ty=t2; case 3 ty=t3; end bianhao=find(dri==y);%与 RGV 距离最 小的 CNC 的标号,可能有两个 cnc_i=min(bianhao);%移动到的 CNC x_rgv=x_cnc(cnc_i);%RGV 位置更新 A=mod(cnc_i,2); switch(A) case 1 tsx=tsx1; case 0 tsx=tsx2; end T=T+ty;n=n+1;cnc(n)=cnc_i; h=n-8;ts(n)=T+tsx;%物料 n 的上料 时刻 tx(h)=ts(n);%物料 n-8 的下料时刻 z(cnc_i)=0;tjw(n)=ts(n)+tj; B(cnc_i)=tjw(n); T=ts(n)+tq;%清洗完成后时间轴更新 for i=1:8 if T>B(i) z(i)=1; end end end [m,n1]=size(tx);nn=n1-1;%货物个数 </pre>	<pre> number=[1:nn]';cnc=cnc(1,1:nn)'; ts=ts(1,1:nn)';tx=tx(1,1:nn)'; jieguo=[number,cnc,ts,tx]%最终结果 xiaolv=(nn*tj)/(28800*8)%效率 </pre>
--	---

附录二：刀具分配随机矩阵

```
G=unidrnd(2,1,8);%产生功能矩阵
G=[1 2 1 2 1 2 1 2]
D1=find(G==1);%下标矩阵
D2=find(G==2);z=G;%产生需求矩阵,1 是
第一道工序需求, 2 是第二道工序需求
[hang1,number_D1]=size(D1);
[hang2,number_D2]=size(D2);
```

附录三：故障审判

```
P=rand(1,1);
while(P<0.01)
    gzbh=gzbh+1;

    gzcnc(gzbh)=cnc_i;
    gzw1(gzbh)=n;

    gzks(gzbh)=ts(n)+(tjw(n)-ts(n)).*
    rand([1,1]);
    gzjs(gzbh)=gzks(gzbh)+900;
    tjw(n)=inf;
    P=rand(1,1)
end
```