

摘要

关键字： 决策模型

目录

一、 问题重述	2
1.1 问题背景	2
1.2 问题提出	2
二、 模型的假设	3
三、 符号说明	3
四、 问题分析	3
4.1 情况一、二分析	4
五、 绘制普通三线表格	4
附录 A 排队算法—matlab 源程序	6
附录 B 规划解决程序—lingo 源代码	6

一、 问题重述

1.1 问题背景

一个智能加工系统由 8 台计算机数控机床 (Computer Number Controller, CNC)、1 辆轨道式自动导引车 (Rail Guide Vehicle, RGV)、1 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。[1] 通常来说,一个工件成品的完成需要若干步,鉴于 RGV 小车在同一时间只能处理同一种任务,而同时最多可以有 8 台数控机床在运行,因此对 RGV 小车进行正确的调度将显著地提高该智能加工系统的加工效率。在本文中,我们将运用不同模型讨论在不同情形下,如何调度小车能够使该系统达到最高效的工作状态。

1.2 问题提出

根据以上的背景,我们在本文中需要针对下面的三种具体情况:

- 一道工序的物料加工作业情况,每台 CNC 安装同样的刀具,物料可以在任一台 CNC 上加工完成;

- 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；
- CNC 在加工过程中可能发生故障（据统计：故障的发生概率约为 1%）的情况，每次故障排除（人工处理，未完成的物料报废）时间介于 10 20 分钟之间，故障排除后即刻加入作业序列。要求分别考虑一道工序和两道工序的物料加工作业情况。

完成两项任务：

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：利用已给出的系统作业参数的 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。

二、模型的假设

基于赛题中给出的一些条件和生活常识，我们在本文中提出如下假设。此后在用到这些假设时，我们将不再另作声明。

- RGV 小车需要对某台数控机床上下料作业时，数控机床对应的传送带上一定有一个预备好的原料工件；
- RGV 小车不能同时进行两项或以上的工作 (包括移动)；
-

三、符号说明

符号	意义
alg	调度算法
n_t	在 t 时间内加工完的工件数
10	421.0
20	640.2

表 1 文中用到的符号和含义

四、问题分析

我们将针对本问题给出的三种情况分别予以分析，并提出可能的优化算法。

4.1 情况一、二分析

在情况一中，一个工件只需经过一步加工就可以成为成品，然后经过 RGV 小车清洗后放入下料传送带送出该系统。此时，一个工件从原料到成品需要 RGV 小车的如下操作：

1. 移动步骤：RGV 小车从原来的位置运行到 CNC 的位置；
2. 第一次上下料步骤：小车将上料传送带上的原料取下，将 CNC 中的成品置换为原料，若 CNC 原来处于空置状态，则只将原料放入 CNC 中；
3. 等待：等待 CNC 加工完成，此时小车可以进行其他任务；
4. 第二次上下料步骤：小车再次运行到 CNC 的位置，将 CNC 中的成品置换为原料，此时小车不能为已装载成品的状态；
5. 清晰步骤：小车清洗成品，并将成品置入下料传送带上。

而在情况二中，一个工件需要两步加工才能制成成品，因此工件由原料到成品需要多一次上下料步骤，即在第一轮加工完成后，将加工完成的工件放入第二步工序的 CNC 中加工。其他的步骤均是相同的。

显然，因为不存在任何随机因素，因此所有小车调度方案的数量是有限的，故情况一、二的最优解是一定存在并且确定的，意即：

$$\exists \text{alg s.t. } n_{t,\text{alg}} = \text{MAX}[n_t]$$

因此我们需要找到该最优的调度方案或接近最优的调度方案

五、 绘制普通三线表格

表格应具有三线表格式，因此常用 booktabs 宏包，其标准格式如表 2 所示。

表 2 标准三线表格

$D(\text{in})$	$P_u(\text{lbs})$	$u_u(\text{in})$	β	$G_f(\text{psi.in})$
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

其绘制表格的代码及其说明如下。

```
\begin{table}[!htbp]
\caption[标签名]{中文标题}
```

```

\begin{tabular}{cc...c}
\toprule[1.5pt]
表头第1个格 & 表头第2个格 & ... & 表头第n个格 \\
\midrule[1pt]
表中数据(1,1) & 表中数据(1,2) & ... & 表中数据(1,n) \\
表中数据(2,1) & 表中数据(2,2) & ... & 表中数据(2,n) \\
..... \\
表中数据(m,1) & 表中数据(m,2) & ... & 表中数据(m,n) \\
\bottomrule[1.5pt]
\end{tabular}
\end{table}

```

`table` 环境是一个将表格嵌入文本的浮动环境。`tabular` 环境的必选参数由每列对应一个格式字符所组成：`c` 表示居中，`l` 表示左对齐，`r` 表示右对齐，其总个数应与表的列数相同。此外，`@{文本}` 可以出现在任意两个上述的列格式之间，其中的文本将被插入每一行的同一位置。表格的各行以 `\\` 分隔，同一行的各列则以 `&` 分隔。`\toprule`、`\midrule` 和 `\bottomrule` 三个命令是由 `booktabs` 宏包提供的，其中 `\toprule` 和 `\bottomrule` 分别用来绘制表格的第一条（表格最顶部）和第三条（表格最底部）水平线，`\midrule` 用来绘制第二条（表头之下）水平线，且第一条和第三条水平线的线宽为 1.5pt，第二条水平线的线宽为 1pt。引用方法：“如表 `\ref{标签名}` 所示”。

参考文献

[1] 2018 年全国大学生数学建模竞赛 B 题, cumcm.cnki.net, 2018.9.13

附录 A 排队算法—matlab 源程序

```
kk=2; [mdd, ndd]=size(dd);
while ~isempty(V)
    [tmpd, j]=min(W(i, V)); tmpj=V(j);
    for k=2:ndd
        [tmp1, jj]=min(dd(1, k)+W(dd(2, k), V));
        tmp2=V(jj); tt(k-1, :)= [tmp1, tmp2, jj];
    end
    tmp= [tmpd, tmpj, j; tt]; [tmp3, tmp4]=min(tmp(:, 1));
    if tmp3==tmpd, ss(1:2, kk)= [i; tmp(tmp4, 2)];
    else, tmp5=find(ss(:, tmp4)~=0); tmp6=length(tmp5);
    if dd(2, tmp4)==ss(tmp6, tmp4)
        ss(1:tmp6+1, kk)= [ss(tmp5, tmp4); tmp(tmp4, 2)];
    else, ss(1:3, kk)= [i; dd(2, tmp4); tmp(tmp4, 2)];
    end; end
    dd= [dd, [tmp3; tmp(tmp4, 2)]]; V(tmp(tmp4, 3))= [];
    [mdd, ndd]=size(dd); kk=kk+1;
end; S=ss; D=dd(1, :);
```

附录 B 规划解决程序—lingo 源代码

```
kk=2;
[mdd, ndd]=size(dd);
while ~isempty(V)
    [tmpd, j]=min(W(i, V)); tmpj=V(j);
    for k=2:ndd
        [tmp1, jj]=min(dd(1, k)+W(dd(2, k), V));
        tmp2=V(jj); tt(k-1, :)= [tmp1, tmp2, jj];
    end
    tmp= [tmpd, tmpj, j; tt]; [tmp3, tmp4]=min(tmp(:, 1));
    if tmp3==tmpd, ss(1:2, kk)= [i; tmp(tmp4, 2)];
    else, tmp5=find(ss(:, tmp4)~=0); tmp6=length(tmp5);
    if dd(2, tmp4)==ss(tmp6, tmp4)
        ss(1:tmp6+1, kk)= [ss(tmp5, tmp4); tmp(tmp4, 2)];
    else, ss(1:3, kk)= [i; dd(2, tmp4); tmp(tmp4, 2)];
    end;
end
dd= [dd, [tmp3; tmp(tmp4, 2)]]; V(tmp(tmp4, 3))= [];
[mdd, ndd]=size(dd);
kk=kk+1;
```

```
end;  
S=ss;  
D=dd(1,:);
```