

# 基于图论和遗传算法对 RGV 动态调度问题的研究

## 摘 要

在智能加工系统中，轨道式自动引导车为计算机数控机床上下料的效率是瓶颈。本文针对智能 RGV 的动态调度问题，运用图论和遗传算法，构建出了一道工序的 RGV 动态调度模型和两道工序的 RGV 动态调度模型，根据我们的模型，我们使用 MATLAB 和 GAMS 软件，构建出了实用的算法与编码方式，得到了 RGV 动态调度的最优方案。

针对问题一，将整个系统看成一张图如上图 3，每个 CNC 是图上的顶点，每两个顶点对应的 CNC 之间的时间对应这两个顶点之间边的权重。我们希望从 1 号 CNC 出发，遍历所有的点得到一条最短路径，也就是所谓的 TSP 问题。关于 TSP 问题有很多解法，精确算法例如动态规划，启发式算法例如遗传算法、蚁群算法等。由于这里情况不太复杂，我们可以运用 GAMS 软件遍历该图的所有路径，从中选择一条整体用时最短的路径，即得到的最优路径。在最优路径下，一道工序一个班次（即 8 小时），共加工 395 个工件，工作效率 75.71%。

针对问题二，在第二种情况下情况比较复杂，我们运用遗传算法解得问题二的最优路径。我们选定 50 个个体作为模型的初始种群。每次 RGV 遍历过所有八个 CNC 后，在这个过程中 RGV 每一次移动所需时间的总和就是所要求的目标函数。因为模型中的适应度函数要求越大，所以取上述目标函数的倒数作为适应度函数。遗传操作的选择复制中，选用保留父代最优染色体替换子代最差染色体、轮盘赌选择的方式。选择整数交叉，最大程度的保留父代优秀的子路信息。最后我们求出最优路径是 1-3-5-7-1-4-2-6-8，在一个班次（8 个小时），共加工 116 个工件，工作率：93.75%。

针对问题三，通过随机数模拟 CNC 是否存在故障，随机产生一个 0-1 的随机数，若小于故障发生率，则说明 CNC 发生故障。因为排除故障所用时间在 10-20 分钟，即 600-1200 秒，所以再用取值在 600 到 1200 之间随机数模拟故障处理需要的时间，独立分出一块来对发生故障的计时。在计时完成之前，出现故障的 CNC 不参与排队。

**关键词：**RGV 动态调度；图论；遗传算法；TSP 问题；优化；甘特图

# 目 录

## 目录

基于图论和遗传算法对 RGV 动态调度问题的研究 .....	1
摘 要 .....	1
目 录 .....	2
§ 1 问题的重述 .....	3
一、背景知识 .....	3
二、相关数据 .....	4
三、具体问题 .....	4
§ 2 问题的分析 .....	3
一、研究现状综述 .....	4
二、对问题的总体分析和解题思路 .....	6
三、对具体问题的分析和对策 .....	7
§ 3 模型的假设 .....	7
§ 4 名词解释与符号说明 .....	7
一、名词解释 .....	7
二、符号说明 .....	7
§ 5 模型的建立与求解 .....	8
一、问题一的分析与求解 .....	8
二、问题二的分析与求解 .....	14
三、问题三的分析与求解 .....	15
§ 6 模型的评价与推广 .....	16
一、模型的评价 .....	16
二、模型的推广 .....	17
参考文献 .....	22
附录 .....	23

## § 1 问题的重述

### 一、背景知识

#### （一）背景介绍

一个智能加工系统由 8 台计算机数控机床、1 辆轨道式自动引导车、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 自带一个机械手臂、两只机械手爪和物料清洗槽，能根据指令能自动控制移动方向和距离并且完成上下料及清洗物料等作业任务。

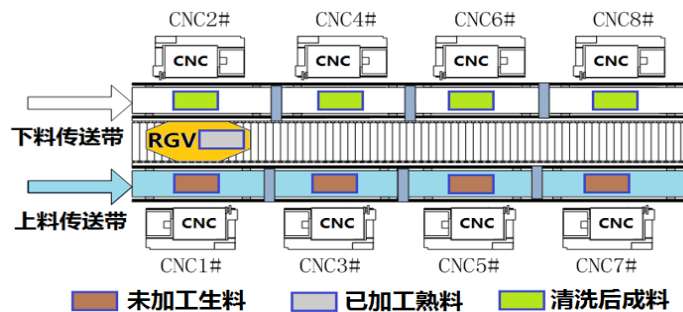


图 1：智能加工系统示意图

#### （二）问题引入

智能加工系统由 8 台 CNC、1 台带机械手和清洗槽的 RGV、1 条 RGV 直线轨道、1 条上料传送带和 1 条下料传送带等附属设备构成。

其作业流程如下：

- （1）智能加工系统通电启动之后，所有 CNC 都处于空闲状态。
- （2）工作正常情况之下，若某 CNC 处于空闲状态，则向 RGV 发出上料需求信号；否则，CNC 处于加工作业状态，在加工作业完成立刻向 RGV 发出需求信号。
- （3）RGV 在收到某 CNC 的需求信号后，它会自行确定该 CNC 的上下料作业次序，并依次按顺序为其上下料作业。
- （4）在 RGV 为某 CNC 完成一次上下料作业后，就会转动机械臂，将一只机械手上的熟料移动到清洗槽上方，进行清洗作业。
- （5）RGV 在完成一项作业任务后，则判别执行下一个作业指令。此时，如果没有接到其他的作业指令，则 RGV 就在原地等待直到下一个作业指令。
- CNC 完成一个物料的加工作业任务后，立刻向 RGV 发出需求信号。如果 RGV 没能即刻到达为其上下料，该 CNC 就会出现等待。
- （6）系统会周而复始地重复（3）至（5），直到停止作业，RGV 回到初始位置。

系统工作流程图如下图 1:

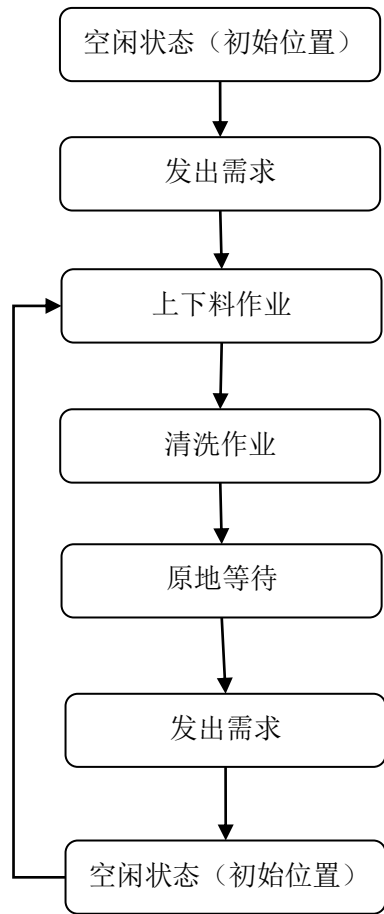


图 1 工作流程图

## 二、相关数据

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

### 三、具体问题

#### （一）第一种情况

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；

#### （二）第二种情况

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；

#### （三）第三种情况

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。

对三种情况分别提出以下任务：

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，并将具体的结果分别填入附件 2 的 EXCEL 表中。

## § 2 问题的分析

### 一、对问题的总体分析和解题思路

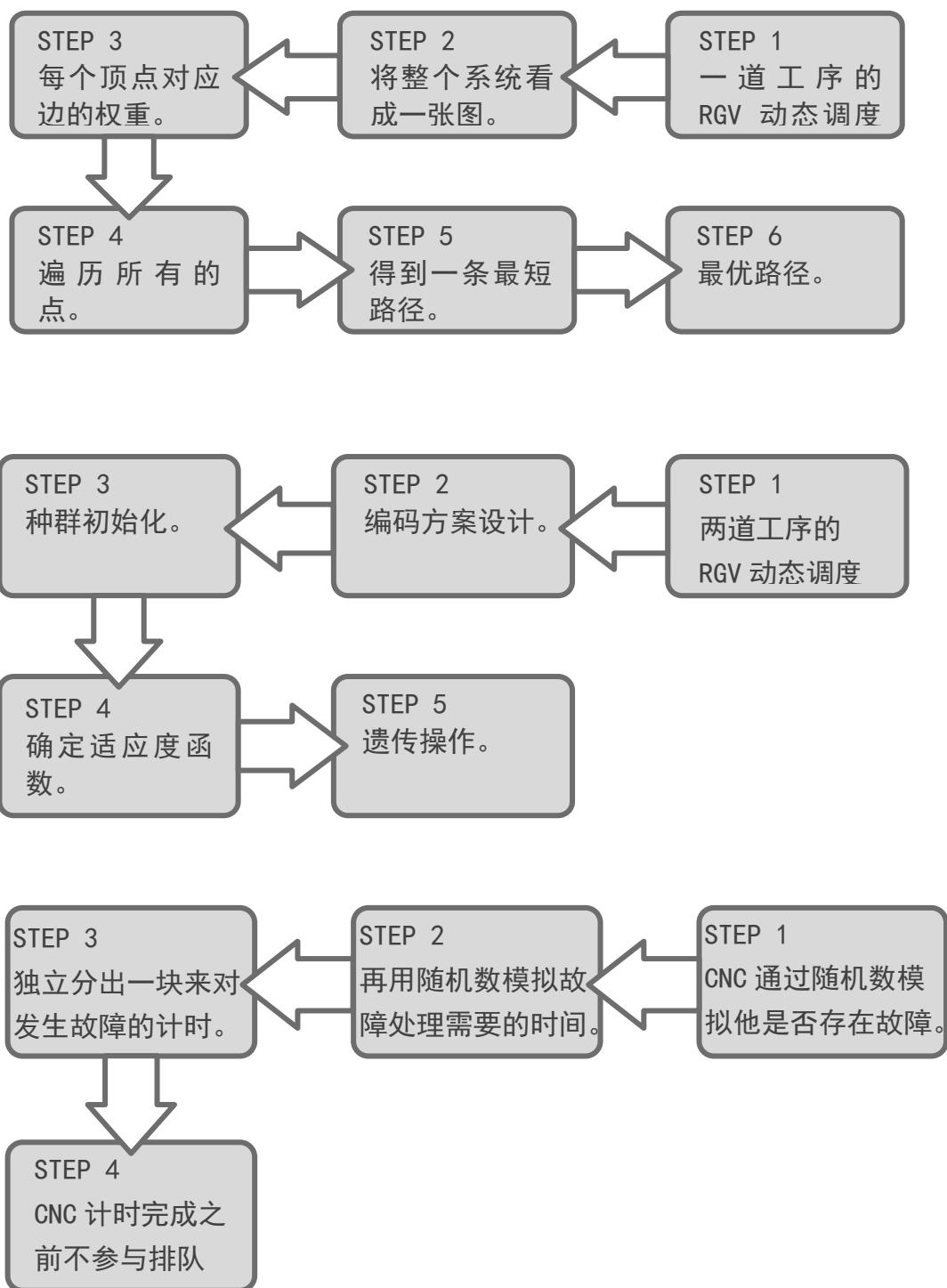


图 2 整体思路框图

### 三、对具体问题的分析和对策

#### （一）对情况一的分析和对策

本问题要求我们给出加工一道工序的物料作业情况下的 RGV 动态调度模型和相应的求解算法。

#### （二）对情况二的分析和对策

本问题要求我们给出加工两道工序的物料作业情况下的 RGV 动态调度模型和相应的求解算法。在第二种情况下我们需要分别考虑加工第一道工序和第二道工序的 CNC 个数与位置，并且第一道工序和第二道工序所需要的时间不同，因此情况比较复杂，我们运用遗传算法解得问题二的最优路径。

#### （三）对问题三的分析和对策

本问题要求我们 CNC 在加工过程中可能发生故障的概率约为 1%的情况，给出 RGV 的调度策略和系统的作业效率。需要考虑的是 CNC 在发生故障停止工作时，对原来调度方式的影响，已经在 CNC 排除故障创新加入作业序列后，又对调度方式的影响。

## § 3 模型的假设

- 一、假设熟料在清洗槽中的清洗时间为零。
- 二、假设在 CNC 的加工过程中故障的发生概率为 1%。
- 三、假设每次故障排除时间为介于 10~20 分钟之间的随机数。
- 四、假设 RGV 在工作过程中能一直正常工作，不会发生故障。

## § 4 名词解释

1. RGV 轨道式自动引导车 (Rail Guide Vehicle)，一种无人驾驶、能在固定轨道上自由运行的智能车。
2. CNC 计算机数控机床 (Computer Number Controller)
3. Hamilton 通路 通过图 G 的每个结点一次，且仅一次的通路（回路），就是 Hamilton 通路（回路）。

4Hamilton 图 .存在 Hamilton 回路的图就是 Hamilton 图。

5. 售货员问题——TSP 一售货员要到若干城市去售货，每座城市只经历一次，问如何安排行走路线，使其行走的总路程最短。

6. 遗传算法 模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

## § 5 模型的建立与求解

### 一、问题一的分析与求解

#### （一）对问题的分析

本问题要求我们给出加工一道工序的物料作业情况下的 RGV 动态调度模型和相应的求解算法。我们需要找到一条 RGV 遍历一遍所有 CNC 的路径，在这条路径中 RGV 移动所用时间最短，可以产生最大的时间效益，提高生产价值。

#### （二）对问题的求解

#### 模型 I ——一道工序的 RGV 动态调度模型

##### 1. 模型的准备

##### （1）模型原理

在这里我们可以用图论来进行模型的建立，将抽象的 RGV 调度模型以路径总时间最短为优化目标的图论模型。这里我们可以抽象为 Hamilton 圈模型中的旅行售货员问题——TSP：一售货员要到若干城市去售货，每座城市只经历一次，问如何安排行走路线，使其行走的总路程最短<sup>[1]</sup>。

想要求解这个模型之前，必须明白关于 Hamilton 图和 Hamilton 通路的概念：通过图 G 的每个结点一次，且仅一次的通路（回路），就是 Hamilton 通路（回路）。存在 Hamilton 回路的图就是 Hamilton 图<sup>[2]</sup>。

在实际求解 TSP 问题的时候首先需要判断其是不是 Hamilton 图（以下简称 H 图）。H 图判定至今没有平凡的充要条件，不过可以通过如下定理辅助判断。

必要条件：



$G$  是  $H$  图  $\Rightarrow$  对于  $V$  的每个非空真子集  $S$ , 均有  $\omega(G-S) \leq |S|$ , 即若去  $k$  个点, 得到连通分支数比  $k$  大, 则不是  $H$  图(逆否命题)。(显然有割点的图不是  $H$  图)

充分条件:

① 设  $G$  是  $n(n \geq 2)$  阶简单图,  $\delta \geq n/2 \Rightarrow G$  是  $H$  图

②  $G$  是简单图, 对于任意不相邻的顶点, 满足  $d(u)+d(v) \geq n$ ,  $G$  是  $H$  图  $\Leftrightarrow G+uv$  是  $H$  图

③  $G$  是  $H$  图  $\Leftrightarrow G$  的闭包是  $H$  图(若  $G$  的闭包是完全图, 则  $G$  是  $H$  图。但一个图的闭包不一定是  $H$  图)

闭包构造过程: 将度数之和  $\geq$  图的顶点个数的非邻接顶点对递归连接起来, 直到不再有这样的顶点对存在。

在确定是  $H$  图后, 需要从中找到一条最优的  $H$  圈, 在一个赋权完全图中, 找出一个有最小权的  $H$  图, 称这个圈为最优  $H$  圈。目前没有有效算法, 但可以通过如下近似算法求得近似值: 首先求出一个  $H$  圈, 通过替换边不断改善上界。通过求最小生成树获得其下界。若想要比较精确的得到最优的  $H$  圈, 可通过遍历所有  $H$  圈找到最优的圈。

## (2) 建模思路

我们将本模型的思路用流程图表示, 如图 2 所示。

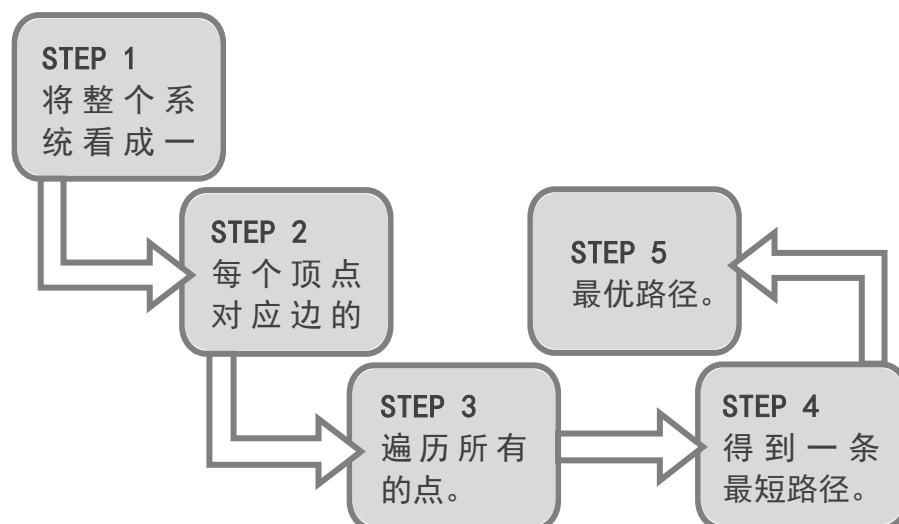


图 2 模型 I 思路图

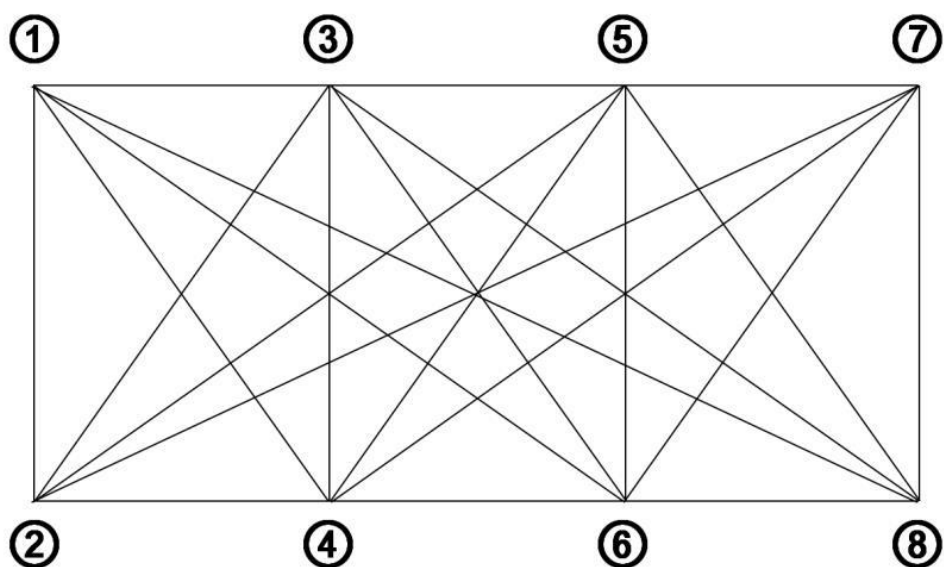


图 3 CNC 模型图

## 2. 模型的建立

经过分析，将整个系统看成一张图如上图 3，每个 CNC 是图上的顶点，每两个顶点对应的 CNC 之间的时间对应这两个顶点之间边的权重。我们希望从 1 号 CNC 出发，遍历所有的点得到一条最短路径，也就是所谓的 TSP 问题。通过一些证明我们可以得到这是 Hamilton 图，存在最优 Hamilton 圈。

## 3. 模型的求解

由于在本题模型中的点的个数比较少，我们可以运用 GAMS 软件遍历该图的所有路径，从中选择一条整体用时最短的路径，即得到的最优路径。

将三组不同的数据带入该模型，用 GAMS 软件求得每组对应的最佳路径，再分别用 matlab 模拟流程，即可得到上料时间、下料时间、加工工件数、工作效率等。在最优路径下，一道工序一个班次（即 8 小时），共加工 395 个工件，工作效率 75.71%。详细求解数据请看附录表格。

数据甘特图如下图 4（每单位 50s）：

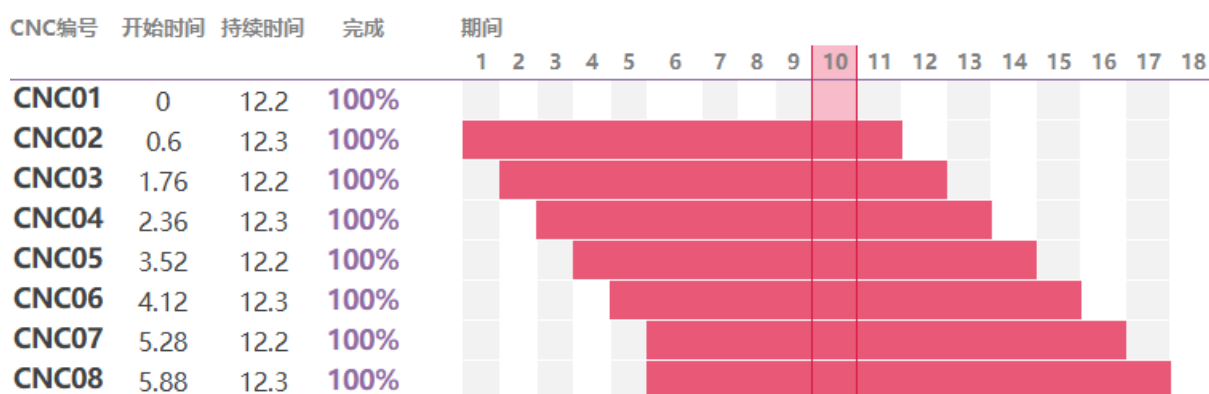


图 4 问题一甘特图

## 模型 II ——两道工序的 RGV 动态调度模型

### 二、问题二的分析与求解

#### 1. 对问题的分析

本问题要求我们给出加工两道工序的物料作业情况下的 RGV 动态调度模型和相应的求解算法。经过分析，将整个系统看成一张图，每个 CNC 是图上的顶点，每两个顶点对应的 CNC 之间的时间对应这两个顶点之间边的权重。所有 CNC 分为两种，一种是加工第一道工序的 CNC，另一种是加工第二道工序的 CNC，我们希望遍历所有的点得到一条最短路径，即得到的最优路径，在这条路径上我需要确定两种 CNC 分别的数量和位置。

#### 2. 对问题的求解

##### (1)模型的准备

##### ① 模型理论

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。遗传算法是从代表问题可能潜在的解集的一个种群开始的，而一个种群则由经过基因编码的一定数目的个体组成。每个个体实际上是染色体带有特征的实体。<sup>【3】</sup>染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（即基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此，在一开始需要实现从表现型到基因型的映射即编码工作<sup>【4】【5】</sup>。由于仿照基因编码的工作很复杂，我们往往进行简化，如二进制编码，初代种群产生之后，按照适者生存和优胜劣汰

的原理，逐代演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度大小选择个体，并借助自然遗传学的遗传算子进行组合交叉和变异，产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码，可以作为问题近似最优解。

## ②建模思路

我们将本模型的思路用流程图表示，如图 5 所示。

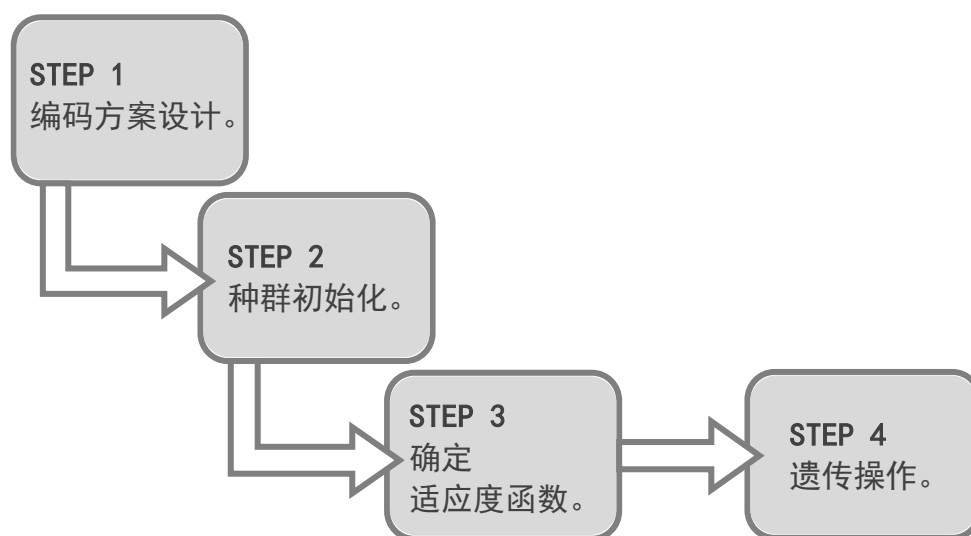


图 5 模型 II 思路图

## (2)模型的建立

### 2.1 编码方案设计

结合本文论文的特点，方案采用自然数编码方式。假设染色体的长度为  $m+n$ ，其中  $m$  代表 RGV 的数量， $n$  代表 CNC 的数量。RGV 的编码是 0，编码 1, 2, 3, ……， $n$  表示各个 CNC 被分配的自然数序号。

例如有一条染色体为 1, 3, 4, 6, 4, 2, 7, 5, 8。表示的路径：RGV 从 1 号 CNC 开始上料，然后依次到 3 号 CNC，4 号 CNC，6 号 CNC 上料，之后又回到 4 号 CNC 下料，然后又来到 2 号 CNC 上料，接着来到 7 号 CNC 上料，又返回到 5 号 CNC 上料，最后来到 8 号 CNC 上料。

### 2.2 种群初始化

首先将所有的 CNC 代码随机排成一排，形成一条初始染色体，反复上述过程产生  $N$

条个体构成初始种群。我们开始选取  $N=50$ ，这 50 个个体作为模型的初始种群。

### 2.3 适应度函数

每次 RGV 遍历过所有八个 CNC 后，在这个过程中 RGV 每一次移动所需时间的总和就是所要求的目标函数。因为模型中的适应度函数要求越大，所以取上述目标函数的倒数作为适应度函数。

### 2.4 遗传操作

#### (1) 选择复制

本文选用保留父代最优染色体替换子代最差染色体、轮盘赌选择的方式。详细的步骤如下：

Step1: 计算目前种群中各个染色体的适应度  $fit(i)$ ;

Step2: 计算种群中所有染色体的适应度之和

$$sumfit(i) = \sum_{i \in N} fit(i), i = 1, 2, \dots, n;$$

Step3: 计算目前种群中各个染色体的选择概率

$$p(i) = fit(i) / sumf, i = 1, 2, \dots, n;$$

Step4: 计算目前种群中各条染色体的累计概率

$$ps(i) = \sum_{i \in N} p(i), i = 1, 2, \dots, n。$$

随机产生一个在  $[0, 1]$  区域内的实数  $r$ ，若  $ps(i) > r$  就选择第一条染色体，否则选择使得  $ps(i-1) < r < ps(i)$  成立的第  $i$  条染色体。

(2) 自然数编码进行交叉的方法有很多。本文选择整数交叉，最大程度的保留父代优秀的子路信息。具体步骤如下：

Step1: 随机产生两个的父代染色体；例如

父亲染色体 1:

1	7	4	8
---	---	---	---

6	3	5	2
---	---	---	---

父亲染色体 2:

1	5	7	4
---	---	---	---

3	2	6	8
---	---	---	---

Step2: 父亲染色体的前半部分分别在子染色体中保留。

子代染色体 1:

1	7	4	8
---	---	---	---

子代染色体 2:

1	5	7	4
---	---	---	---

Step3: 在父亲染色体 2 中剔除与父亲染色体 1 前半部重复的部分，将剩下的部分按原顺序构成子代染色体 1 的后半部分。同理在父亲染色体 1 中剔除与父亲染色体 2 前半部重复的部分，将剩下的部分按原顺序构成子代染色体 2 的后半部分。即可得到子代染色体 1 和子代染色体 2。

子代染色体 1:

1	7	4	8
---	---	---	---

5	3	2	6
---	---	---	---

子代染色体 2:

1	5	7	4
---	---	---	---

3	2	6	8
---	---	---	---

### (3)模型的求解

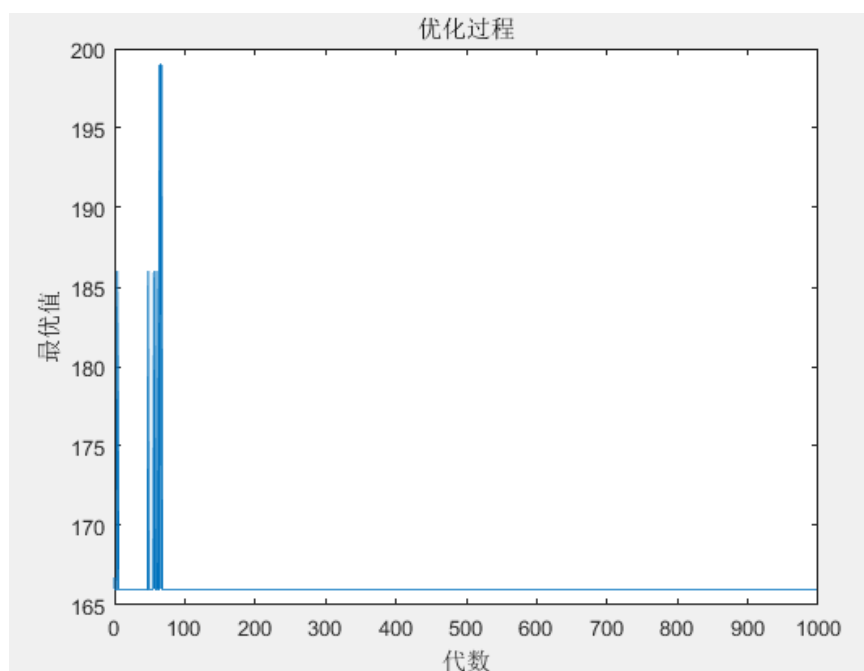


图 6 遗传算法

如上图 6，我们选择种群数是 300，然后迭代 1000 代，逐渐趋于平稳说明达到最优

值了，最后我们求出最优值是 1-3-5-7-1-4-2-6-8。别用 matlab 模拟流程，即可得到上料时间、下料时间、加工工件数、工作效率等。在一个班次（8 个小时），共加工 116 个工件，工作率：93.75%，详细求解数据请看附录表格。

数据甘特图如下图 7(1)和图 7（2）（每单位 50 秒）：

CNC编号	第一道工序 开始时间	第一道工序 持续时间	第二道工序 开始时间	第二道工序 持续时间	百分比 完成
CNC01	0	8.56	9.52	11.76	100%
CNC02	0.96	12.76	14.68	8.18	100%
CNC03	1.92	16.34	18.82	6.16	100%
CNC04	2.88	18.64	22.08	6.5	100%

图 7（1） 甘特图数据部分

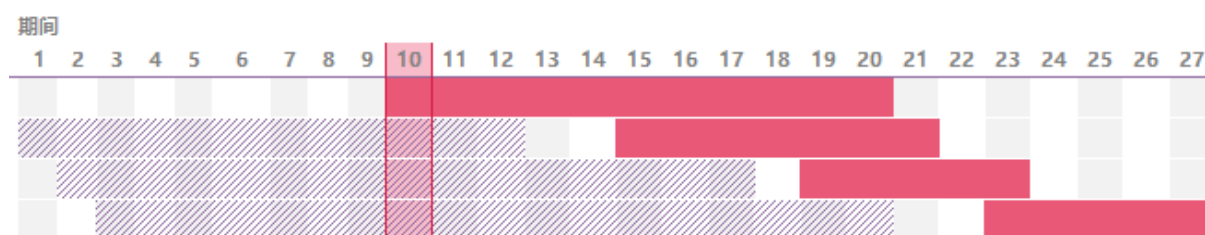


图 7（2） 甘特图图象部分

### 三、问题三的分析与求解

#### 1. 对问题的分析

本问题要求我们 CNC 在加工过程中可能发生故障的概率约为 1%的情况，给出 RGV 的调度策略和系统的作业效率。需要考虑的是 CNC 在发生故障停止工作时，对原来调度方式的影响，已经在在 CNC 排除故障创新加入作业序列后，又对调度方式的影响。

#### 2. 建模思路

我们将本模型的思路用流程图表示，如图 7 所示



图 7 问题建模思想

### 3. 对问题的求解

故障发生率的模拟：随机产生一个 0-1 的随机数，若小于故障发生率，则说明 CNC 发生故障。故障的同时进行记时。

修理时间的模拟：同时，产生一个 600-1200 的随机数，确定修理时间。修理过程中，剔除损坏工件进行排序。记时完成后，加入排序。

当物料加工只有一道工序时，在问一中我们是先求出最短路径再进行模拟。在本问中，还是需要建立在图论的基础上，需要对其过程进行改进：每到一个点，需要进行一次故障判断，若有 CNC 故障，该 CNC 对应图论模型中的点记为故障点并进行记时，但故障点不参与排序。依次反复进行模拟得到该情况的解。

当物料架加工有两道工序时，在问二中我们用遗传算法进行求解。需要将遗传算法进行改进，需考虑故障因素，在计算种群的适应度函数的时候，每经过一个基因，需要判断其他基因的损坏率（故障率），若损坏则进行剔除，并产生目标函数的增大从而导致适应度函数的减小。在考虑机器故障情况下的遗传算法得出的最优解会比不考虑故障情况下的值偏大。

## § 6 模型的评价与推广

### 一、模型的评价

#### （一）模型的优点：

1. 本文巧妙地运用流程图，将建模思路完整清晰的展现出来；
2. 利用 Excel 软件对数据进行处理并做出图表，快捷、简便、直观；
3. 本文采用图论建模，并且用遗传算法求解，计算精度高、运算时间短。



## （二）模型的缺点：

- (1)对于一些数据，我们对其进行了一些必要的处理，这会带来一定的误差；
- (2)模型中为使计算简便，使所得结果更理想化，忽略了一些次要影响因素。

## 二、模型的推广

### 1. RGV 环形轨道模型的推广

本文中任务一基于图论、排队论以及遗传算法的方法，得到了 RGV 动态调度模型和相应的求解算法，本模型还可以用于 RGV 环形轨道模型中，且模型精度较高，预测结果较为可靠。

### 2. 多辆轨道式自动引导车模型的推广

任务二根据系统作业参数的 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。这个模型还可以推广到多辆轨道式自动引导车模型，并且得到充分的验证。

## 参考文献

- [1]刘永强. 基于遗传算法的 RGV 动态调度研究[D]. 合肥工业大学, 2012.
- [2]甘剑锋, 周晓光. 基于排队论的自动化立体仓库 AGV 调度效率分析[J]. 计算机测量与控制, 2004(07):657-659.
- [3]高升. 基于电动汽车的带时间窗的路径优化问题研究[D]. 大连海事大学, 2015.
- [4]张腾飞, 马跃, 胡毅, 安涛, 王帅, 郭安. 基于遗传算法的多目标车间调度问题的研究[J]. 组合机床与自动化加工技术, 2016(05):43-45+50.
- [5]张富生. 基于遗传算法的车间动态调度研究[D]. 山东大学, 2013.
- [6]张俊杰, 杨艳丽, 曹岩, 白瑀, 蔺麦田. 图论中最短路径问题算法程序的开发[J]. 机电产品开发与创新, 2008(01):103-104.
- [7]王兵团. 数学模型基础[M], 北京: 北京交通大学出版社, 2005.
- [8]张杰, 郭丽杰. 运筹学模型及其应用[M], 北京: 清华大学出版社, 2012.
- [9]司守奎, 孙玺菁. 数学建模算法与应用[M]. 北京: 国防工业出版社, 2015
- [10]朱道立. 运筹学[M]. 北京: 高等教育出版社, 2006

## 附录

### 附：遗传算法程序

%RGV 移动一、二、三个单位所用时间

t1=20;

t2=33;

t3=46;

%CNC 的时间距离矩阵

```
C=[inf,0,t1,t1,t2,t2,t3,t3;  
    0,inf,t1,t1,t2,t2,t3,t3;  
    t1,t1,inf,0,t1,t1,t2,t2;  
    t1,t1,0,inf,t1,t1,t2,t2;  
    t2,t2,t1,t1,inf,0,t1,t1;  
    t2,t2,t1,t1,0,inf,t1,t1;  
    t3,t3,t2,t2,t1,t1,inf,0;  
    t3,t3,t2,t2,t1,t1,0,inf];
```

%变异概率

pm=0.05;

%迭代次数

maxgen=1000;

%种群数量

NIND=300;

%初始化种群

s=[];

for i=1:NIND

    ss=(randperm(numel(1:7)))+1;

    n=[];

    n(1)=1;

    n(2:4)=ss(1:3);

    n(5)=1;

    n(6:9)=ss(4:7);

    s(i,:)=n;

end

gen=0;

figure;

hold on;box on;

xlim([0,maxgen]);

```

title('优化过程');

xlabel('代数');

ylabel('最优值');

fit=fitness(s,C,NIND);
sumfit=sum(fit);
p=fit./sumfit;
ps=[];
for i=1:NIND
    ps(i)=sum(p(1:i));
end
while gen<maxgen
    ff=1./fit;
    line([gen-1,gen],[preff,min(ff)]);
    preff=min(ff);

%适应度函数

fit=fitness(s,C,NIND);
sumfit=sum(fit);
p=fit./sumfit;
ps=[];
for i=1:NIND
    ps(i)=sum(p(1:i));
end

%选择

seln=sel(ps);

%交叉

scro=[];
for i=1:2:NIND-1
    seln=sel(ps);
    [ z1,z2 ] = cro( s,seln );
    scro=[scro;z1];
    scro=[scro;z2];
end

%变异

```

```

for i=1:NIND
    if pm>rand
        R=randsrc(1,2,2:9);
        if R(1)==5
            break;
        end
        if R(2)==5
            break;
        end
        jj=scro(i,:);
        scro(i,R(1:2))=scro(i,R(2:-1:1));
        ji=scro(i,:);
    end
end

[fitmax,nmax]=max(fit);

%更新迭代次数

gen=gen+1;
s=scro;
end
[fitmax,nmax]=max(fit);
spath=s(nmax,:)
doo=1./fitmax

%

function [ seln ] = sel( ps )

%选择

for i=1:2
    r=rand;
    prand=ps-r;
    j=1;
    while prand(j)<0
        j=j+1;
    end
    seln(i)=j;
end

end

function [ fit ] = fitness( s,C,NIND )

%适应度函数

```

```

f=[];
for i=1:NIND
    ff=0;
    p=[];
    pp=s(i,:);
    p(1:5)=pp(1:5);
    p(6)=pp(6);
    p(7)=pp(2);
    p(8)=pp(7);
    p(9)=pp(3);
    p(10)=pp(8);
    p(11)=pp(4);
    p(12)=pp(9);
    p;
    for j=1:11
        ff=ff+C(p(j),p(j+1));
    end
    f=[f;ff];
end
fit=1./f;

end

function [ z1,z2 ] = cro( s,seln )

%交叉

f_n=s(seln(1),:);
m_n=s(seln(2),:);

%子代染色体1

p1=f_n;
p2=m_n;
d=f_n(2:4);
for i=1:3
    for j=1:9
        if p2(j)==d(i)
            p2(j)=0;
        end
    end
end
pp2=[];
for i=1:9

```

```

        if p2(i)>1
            pp2=[pp2,p2(i)];
        end
    end
    d;
    pp2;
    zz1=[d,pp2];
    z1=[];
    z1(1)=1;
    z1(2:4)=zz1(1:3);
    z1(5)=1;
    z1(6)=zz1(4);
    z1(7)=zz1(5);
    z1(8)=zz1(6);
    z1(9)=zz1(7);
    z1;

%子代染色体 2

p1=f_n;
p2=m_n;
d=m_n(2:4);
for i=1:3
    for j=1:9
        if p1(j)==d(i)
            p1(j)=0;
        end
    end
end
pp1=[];
for i=1:9
    if p1(i)>1
        pp1=[pp1,p1(i)];
    end
end
pp1;
zz2=[d,pp1];
z2=[];
z2(1)=1;
z2(2:4)=zz2(1:3);
z2(5)=1;
z2(6)=zz2(4);
z2(7)=zz2(5);
z2(8)=zz2(6);
z2(9)=zz2(7);

```

```
z2;
```

```
end
```