1. 基于窗口指派的准时制生产调度算法模型构建
   1. 问题描述

**参数介绍：（对应数值例子看能更好理解）**

一台机器：M

零件：

工件的正常加工时间：

实际加工时间：

工件的加工顺序：，例如

工件的实际完工时间：

工件的开始加工时间：

学习率：

恶化率：

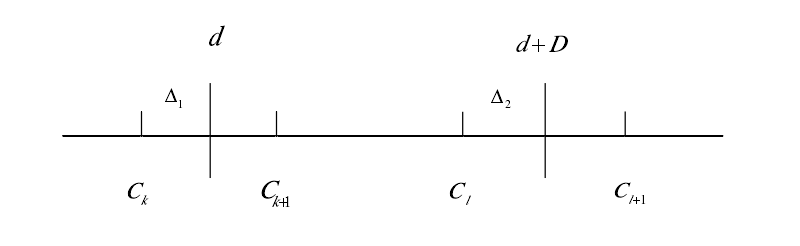
注：学习率恶化率可以理解为两个最后计算时代入的参数

：工件交货窗时的开始时间

：窗口大小

之前完工的工件为提前完工工件

以后完工的工件为延迟工件



工件的提前完工时间：

工件的延误完工时间：

，及分别为提前完工工件数，窗口中工件数和迟延完工工件数且，，，且的工件为提前完工5工件，在位置的工件为窗口工件，在位置的工件为迟延完工工件。

目标：分别为提前完工，延迟完工，窗时开始时间和窗时大小的每一时间单元的惩罚，目标是去找一个最优序使得每一个工件最优交货窗口总罚值最小。

决定最优的窗时开始时间，窗时大小及工件加工顺序使得目标函数最小化。

注：这里的罚值总和也叫时间成本，目标也就是使时间成本最小。

* 1. 基于窗口指派的准时制生产调度问题模型
  2. **定理2.1**

两个组数相乘之后求和，第一组中最大的数乘第二组中最小的数，第一组中第二大的数乘第二组中第二小的数，以此类推，求和之后为最小。

**定理2.2**

存在一个最优排序，从第一个工件开始加工到最后一个工件加工结束的过程中没有机器空闲。

注：时间成本最低当然不能让机器空着。

**定理2.3**

最优序列中，，正好与序列中某工件的完工时间相同，设其为工件序列中的第，个工件，，，窗时开始时间点会与第个位置工件的完工时间一致，会与第l个位置的工件的完工时间一致，即，。对这一最优顺序，其中，满足：

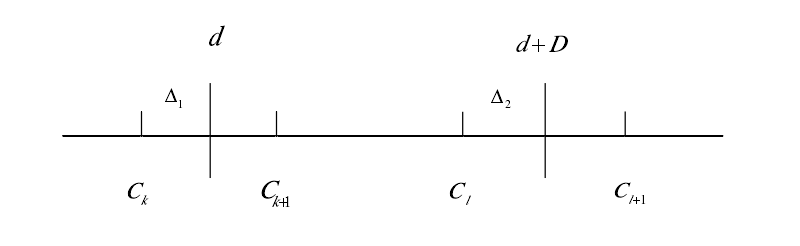




**证明：（过程看懂就行）**

目标函数

定义，



1. 提前完工工件的位置为的工件，这部分的提前成本总和为：

 注：根据目标函数





注：第一个工件的加工时间为，第二个工件的加工时间为，以此类推。

相对应的是第一个工件的完工时间为，第二个工件的完工时间，以此类推。







注：

（2）延迟完工工件的位置为的工件，这部分的延迟成本总和为：





1. 窗口开始时间的成本总和：



（4）窗口大小时间的成本总和：





 注：用来表示目标函数





注：把上面求得的四个部分的成本加和得到总成本目标函数。



其中，

，

。

由上可得，，和为常数且与和无关。并且。此时使总成本最小，分情况分析：

注：且，且，当且，当且四种情况

情况1：当且，只有且时，目标函数才能取得最小值。

这时总成本为，

且

情况2：当且，只有且时，目标函数才能取得最小值。

这时总成本为，

且

情况3：当且，只有且时，目标函数才能取得最小值。

这时总成本为，

且

情况4：当且，只有且时，目标函数才能取得最小值。

这时总成本为，

且

由以上四钟情况得到的不等式可以推出定理2.2结论：窗时开始时间点会与第个位置工件的完工时间一致，会与第个位置的工件的完工时间一致，即，，且，。

**定理2.4** 存在一个第一个工件从零时刻开始加工的最优排序。

注：不解释

**注：重点来了**

**定理2.5**  最优排序总成本可以表示为，

**证明：**由定理2.3得出位置权数的定义：

，。

注：





通过定理2.3我们可以得出结论，每部分的成本都是关于或者的线性函数，因此也是关于或者的线性函数。

综上所述，最优排序取得的目标函数为：



注：。位置权数很重要，三部分分别代表提前，窗口内，延迟的权数。

**定理2.6** 最优排序总成本可以表示为，其中，为与对应的位置函数表达方式。

注：一个是W,一个是w。

**证明：**工件的正常加工时间：

实际加工时间：

（我的错，实际加工时间里的a是学习率，参数重了）

工件的开始加工时间：

代公式：

，（工件加工间隙为0嘛）

，

，

，







将



代入，

目标函数可以被描述成：

















将提出，可得：











，

其中：













1. 基于窗口指派的准时制生产调度问题算法求解

由第二章的证明可知，最优排序取得的目标函数为：



为了使目标函数取得最小值，位置权数与正常加工时间的乘积应该尽可能小，所以，正常加工时间长的工件所对应的位置权数应该尽可能小，因此，进一步我们可以得出问题的算法。

* 1. 算法

第一步：计算出位置权数，，以及对应的位置函数表达方式。

第二步：位置权数与正常加工时间的乘积应该尽可能小，所以把最大正常加工时间工件匹配给最小的位置权数，把第二大正常加工时间的工件匹配给第二小的位置权数，以此类推。

第三步：若，则在位置的工件为提前完工的工件。

若，则在位置的工件为窗口内的工件。

若，则在位置的工件为延迟完工的工件。

第四步：计算出工件的实际完工时间，此时最优窗口的开始时间为，窗口大小为，进而求出总成本目标函数。

（注：窗口最优开始时间也就是提前完工工件完工之后开始，就是从1到，窗口大小就是从提前完工工件完工之后第一个工件开始，到窗口内工件完工时间结束的这段时间，就是到）

* 1. 数值例子

某共产将要生产7个工件，工件的正常加工时间为：

，，，，，，。

参数为：

，，，，，。

解：

第一步：计算出位置权数，如下表所示：

注：权数，n=7，

位置权数是w，位置函数是W,二者关系为

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **r** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 提前完工权数 | 35 | 46 | 57 | 68 | 79 | 90 | 101 |
| 窗口完工权数 | 49 | 49 | 49 | 49 | 49 | 49 | 49 |
| 延迟完工权数 | 126 | 108 | 90 | 72 | 54 | 36 | 18 |
| 位置权数值 | 35 | 46 | 49 | 49 | 49 | 36 | 18 |
| 位置函数值 | 43.6597 | 42.0439 | 37.6814 | 33.2827 | 30.0794 | 20.4881 | 9.6195 |

表3.1

第二步：匹配位置函数值

工件的正常加工时间为：

，，，，，，。

位置函数值为：

,,,,,,.

工件按正常加工时间从小到大排序为：



位置函数值按从大到小排序为：



按乘积最小原则匹配后的排序结果为：



第三步：把工件按生产完成时间进行分类：

若，则在位置的工件为提前完工的工件。

若，则在位置的工件为窗口内的工件。

若，则在位置的工件为延迟完工的工件。

我们由表3.1得到，工件，为提前完工的工件，，，为按时完工工件，，为延迟完工工件。

（注：最后取到的权数值，不是三个取最小嘛，如果是在提前那取的，就是提前工件；如果在窗时取的，就是窗时工件；如果在延后取的，就是延迟工件，见表3.1）

第四步：计算工件的实际完工时间，如下表所示：

注：

1.表3.2类似银行家算法

2.（跟一样，就这么理解就行）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **r** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| a | 3 | 4 | 6 | 9 | 14 | 18 | 20 |
| t | 0 | 3 | 6.319834 | 10.753976 | 16.857531 | 25.697503 | 36.528082 |
|  | 1 | 0.79996 | 0.70205 | 0.63994 | 0.59557 | 0.56161 | 0.53442 |
|  | 3 | 3.319834 | 4.434142 | 6.103555 | 8.839972 | 10.830579 | 11.664467 |

表3.2

综上所述我们得出：

，。

总的时间成本：

