

赛区评阅编号（由赛区组委会填写）：

---

## 2017 高教社杯全国大学生数学建模竞赛

### 承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D 中选择一项填写）：\_\_\_\_\_

我们的报名参赛队号（12 位数字全国统一编号）：\_\_\_\_\_

参赛学校（完整的学校全称，不含院系名）：\_\_\_\_\_ 武汉大学

参赛队员（打印并签名）：1. TZX

2. TJG

3. ZY

指导教师或指导教师组负责人（打印并签名）：\_\_\_\_\_

日期：\_\_\_\_\_ 2016 年 05 月 31 日

（此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号（由赛区组委会填写）：

---

**2017 高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅记录（可供赛区评阅时使用）：

评 阅 人						
备 注						

送全国评阅统一编号（由赛区组委会填写）：

全国评阅随机编号（由全国组委会填写）：

（此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页，即电子版论文的第一页为标题、摘要和关键词页。）

# 任务车间调度问题解析

## 摘要

本文根据所给车间作业问题对任务车间调度问题进行了描述与数学分析，主要针对不同工件不同加工工序的车间作业调度问题进行了数学建模，并通过编程实现了基于作业时间最短的情况下进行作业计划的求解。

**关键字：** 车间作业调度 数学建模

## 一、问题背景

随着市场竞争的日益激烈，面向顾客订货的、多品种、小批量生产已经成为 21 世纪的主导生产模式。在这种生产环境下，如何安排生产计划、如何进行调度成为企业有效进行生产的关键问题。每个企业都在寻求更好的生产与运作管理方案，以提高企业的生产、经营和管理效率，从而提高企业的核心竞争优势。

生产与运作管理的核心是车间作业调度问题能否高效地获得优化解，因此，车间调度策略一直是制造业研究的重点之一，对车间作业调度问题的研究具有重要的理论意义和现实意义。有效的调度方法与优化技术的研究和应用，对于制造企业提高生产效率、降低生产成本等方面起着重要作用，越来越受到学者们的关注。

## 二、问题重述

现有某墙纸生产厂接到三种类型墙纸的订单，第一种墙纸在蓝色背景上有黄色图案，第二种墙纸在绿色背景上有蓝色和黄色图案，第三种墙纸在黄色背景上有蓝色和绿色图案。在生产时，每种墙纸都是一个连续的纸卷，且将要通过三台机器，每台机器向墙纸上印刷不同的颜色。墙纸通过机器的顺序取决于墙纸的设计，对于第一种墙纸，先印刷蓝色背景，再印刷黄色图案；对于第二种墙纸，首先印刷绿色背景，然后先印刷蓝色图案，再印刷黄色图案；对于第三种墙纸，首先印刷黄色背景，然后先印刷蓝色图案，再印刷绿色图案。每一个工序的处理时间取决于需要向墙纸上印刷的对象。各种墙纸每种颜色的印刷时间如下表所示。

表 1 每种颜色印刷需时（单位：分钟）

机器	颜色	墙纸 1	墙纸 2	墙纸 3
1	绿色	—	10	17
2	蓝色	45	20	12
3	黄色	10	34	28

若每台机器同时只能处理一种墙纸，一种墙纸不能同时由多台机器进行处理，试制定墙纸在机器之间的印刷计划，以使订单尽快完成。

## 三、模型假设

1. 整个加工过程中，每台机器同时只能处理一种墙纸；
2. 一种墙纸不能同时由多台机器进行处理；

3. 任何一个墙纸的前一道工序加工完成后，方能进行后一道工序的加工；
4. 各墙纸必须按工艺路线以指定的次序在机器上加工；不考虑墙纸的优先权；
5. 每个墙纸的每道工序的开工时间一定大于或等于零；
6. 墙纸的加工时间事先给定，且在整个加工过程中保持不变；
7. 墙纸转换机器的耗时为 0；
8. 墙纸在印刷工序中不需要进行静置晾干等。

## 四、符号说明

符号	意义
$t_{ip}$	墙纸 $i$ 在机器 $p$ 上需要的时间；
$S_{ip}$	墙纸 $i$ 在机器 $p$ 上处理的开始时间；
$R_{ipq}$	$R_{ipq} = \begin{cases} 1, & \text{如果机器 } p \text{ 先于机器 } q \text{ 处理墙纸 } i, \\ 0, & \text{其他情况;} \end{cases}$ , 其中, $i, p, q \in \{1, 2, 3\}$
$I_{pij}$	$I_{pij} = \begin{cases} > 0, & \text{墙纸 } i \text{ 先于墙纸 } j \text{ 在机器 } p \text{ 上加工,} \\ \leq 0, & \text{其他情况;} \end{cases}$ , 其中, $p, i, j \in \{1, 2, 3\}$
$C_{ip}$	墙纸 $i$ 在机器 $p$ 上完成工序时的时刻；

## 五、问题分析

### 5.1 背景理解

该车间作业调度问题，可以描述为一个要完成的作业集合，每个作业由一个操作集组成，包含一个工序集合，各操作工序的完成需要占用机床或其它资源，并且必须按一些可行的工艺路线、工艺次序进行加工，每个机床固定的加工工序。研究的目的是确定一个调度过程，在约束条件下，该调度将每个工序分配到对应机器的某个时间段，使得完成所有作业所需的加工持续时间（制造周期）最小。

### 5.2 约束条件

**约束 1**，每台机器同时只能处理一种墙纸。

在机器  $p$  上，当墙纸  $i$  先于  $j$  处理时： $C_{ip} \leq S_{jp}$ ；当墙纸  $i$  后于  $j$  处理时： $C_{jp} \leq S_{ip}$ 。

**约束 2**，一种墙纸不同同时由多台机器进行处理。

对于墙纸  $i$ ，当机器  $p$  先于  $q$  处理时： $C_{ip} \leq S_{iq}$ ；当机器  $p$  后于  $q$  处理时： $C_{iq} \leq S_{ip}$ 。

**约束 3**，每种墙纸的处理时间一定非负。

即  $S_{ip} \geq 0$ 。

## 六、模型建立

目标函数为

$$\min \left( \max_{\substack{i=1,2,3 \\ p=1,2,3}} C_{ip} \right)$$

s.t.

$$\begin{cases} S_{ip} \geq 0, & i, p \in \{1, 2, 3\} \\ C_{ip} = S_{ip} + t_{ip}, & i, p \in \{1, 2, 3\} \\ I_{pij} = \begin{cases} > 0, & p \text{ 机器上 } i \text{ 比 } j \text{ 先处理} \\ \leq 0, & \text{其他情况} \end{cases}, & i, j, p \in \{1, 2, 3\} \\ R_{ipq} = \begin{cases} 1, & p \text{ 机器比 } q \text{ 机器先处理 } i \text{ 墙纸} \\ 0, & \text{其他情况} \end{cases}, & i, j, p \in \{1, 2, 3\} \\ (C_{ip} - S_{iq}) \times R_{ipq} < 0, & i, p, q \in \{1, 2, 3\} \\ (C_{ip} - S_{jp}) \times I_{pij} < 0, & i, j, p \in \{1, 2, 3\} \end{cases}$$

## 七、模型求解

用 C++ 编程解决这种规划问题相对较难，所以我们临时决定使用 Lingo 对模型进行求解。但由于不熟悉 Lingo 的使用（也由于免费版 Lingo 对变量数目有一定限制），程序无法运行。源码见附录 1。

## 八、基于另一假设的优化

这里假定了墙纸的加工过程开始后，必须完成一个阶段的工序，才能进入下一步处理（或者进入等待）。基于印刷机器一般为线扫描，这种假设一般是合理的。但如果这里的印刷机机器进行的是整面喷涂，加工时间其实是和颜料的深度有关，有或者机器可以中断工序、记忆进度、并在下次继续之前的工作，则可以将单个工序拆解为多段。在这一新的假设下，处理时间会进一步缩短，比如墙纸 1，需要在 2 号机器处理 45 分钟，进行蓝色背景的印刷，这个过程可以分为 10 分钟 + 17 分钟 + 18 分钟。这样，可以把三种墙纸的各阶段的处理时间看成序列，在三台机器之间进行分配。将整个处理流程看成一个状态机，通过编写 C++ 程序，结果显示总处理时间可以进一步优化至 87 分钟。程序输出结果如下（添加了必要的注释说明）：

```

// 初始状态，t1，t2，t3 表示三种墙纸，B，G，Y 分别表示
// 蓝色（Blue）、绿色（Green）、黄色（Yellow）的印刷任务

t1 (B, 45) (Y, 10)           // (B, 45) 表示有 45 分钟的
t2 (G, 10) (B, 20) (Y, 34) // 蓝色印刷任务
t3 (Y, 28) (B, 12) (G, 17)
      3/3: 10           // 三台机器都在工作，进行 10 分钟

t1 (B, 35) (Y, 10)
t2 (B, 20) (Y, 34)
t3 (Y, 18) (B, 12) (G, 17)
      2/3: 18           // 两台机器工作，18 分钟

t1 (B, 35) (Y, 10)
t2 (B, 02) (Y, 34)
t3 (B, 12) (G, 17)
      1/3: 2           // 一台机器，2 分钟

t1 (B, 35) (Y, 10)
t2 (Y, 34)
t3 (B, 12) (G, 17)
      2/3: 12           // 两台机器，12 分钟

t1 (B, 35) (Y, 10)
t2 (Y, 22)
t3 (G, 17)
      3/3: 17           // 三台机器，17 分钟

t1 (B, 18) (Y, 10)
t2 (Y, 05)
t3
      2/2: 5           // 两台机器，而且只剩
                        // 两种墙纸需要印刷，5 分钟

t1 (B, 13) (Y, 10)

```

```
t2
```

```
t3
```

```
1/1: 23
```

```
** Used 87 minutes. **
```

C++ 源程序见附录 2。

在新的假设下，三种墙纸的处理耗时从 97 分钟（? 待考证）降低到了 87 分钟，节约了 10.3% 的处理时间。

你可以在 <https://github.com/district10/TextureProblem> 查看论文  $\text{\LaTeX}$  源码和程序源码。