

计导课程设计—Job shop 管理游戏

一、任务概述

你是一间超级工厂的管理员 (BOSS)，每天都要在指定时间段内接受客户提交的 n 个产品 (Job) $\{J_i\}_{i=1}^n$ 加工订单，每个产品都会指定加工步骤 (Operation)，必须按照 $O_{i1}, O_{i2}, \dots, O_{in_i}$ 的顺序加工。其中 n_i 为产品 J_i 包含的加工步骤。每道工序 (步骤) 必须在指定的机器 (Machine) $\{M_j\}_{j=1}^m$ 上加工，且每道工序的加工时间固定已知。你的任务是通过计算，尽可能使完成所有订单的总加工时间最短，并将你的加工计划展示出来。

完成这一任务必须满足以下约束条件：

约束 1: 程序启动后首先接受产品加工请求输入，确认当天的需求后不再接受新的订单。对每个产品需满足加工工序约束，即只有其前一个工序加工完毕，才能加工其后一个工序。每个工序的加工时间已知，且所需机器固定(每台机器都不能被其他机器替代)，均作为程序的输入。

举例说明，假设你收到 3 个产品的订单，需要在 3 台机器上加工，第 1 个产品有 3 个操作 O_{11}, O_{12}, O_{13} ；第 2 个产品有 2 个操作 O_{21}, O_{22} ；第 3 个产品有 2 个操作 O_{31}, O_{32} 。每个产品的各工序所需机器和加工时间如表 1 所示。括号中的第一项表示工序的加工时间，第二项表示所需机器。例如，产品 1 的第一个工序必须在机器 M_1 上加工，加工时间为 7 个时间单元。表 1 为程序的输入。每个产品需满足加工工序约束。以第 2 个产品为例，只有当工序 O_{21} 加工完毕，才能开始加工 O_{22} 。

表 1. 要输入的参数

	产品 1	产品 2	产品 3	...
操作 1	(7, M_1)	(10, M_2)	(7, M_1)	...
操作 2	(12, M_2)	(17, M_1)	(22, M_2)	...
操作 3	(15, M_3)			...
...				...

约束 2: 每台机器同一时间只能加工一个操作，一旦开始加工一个操作就要加工完成，期间不允许中断。

二、程序输入

- 输入参数如下：
 - 1 产品的数目 n ;
 - 2 所用机器数目 m ;
 - 3 表 1 所示的各工序的加工时间，所用机器约束。
- 要求开发两个版本：命令行版本和动画版本。具体说明如下：

命令行版本，要求实现文件输入和键盘输入，文件和命令行同时输出。

动画版本，要求实现图形界面输入，文件和动画同时输出。

1. 文件输入 文件名为input.txt，文本文件格式为：

首行输入产品数和机器数，格式为：

<产品数目> <空格><机器数目><\n>

之后每一行代表一个产品的加工订单，‘-1’表示结束。

每行的订单输入格式规定如下：

<产品序号><空格><(><按顺序的工序所花时间><,><工序指定机器号><)><空格>...<\n>

例如上面表1的加工三个产品的订单，输入文件内容为：

3 3

1 (7,1) (3,2) (15,3)

2 (10,2) (17,1)

3 (7,1) (22,2)

-1

在软件系统开发期间，老师会提供几组测试数据（输入文件和参照方案）给同学们，便于大家测试。最后验收的时候，老师会现场给定新的测试数据，来验证各组程序的算法优劣。

2. 从键盘输入

命令行方式下，首先输入产品数和机器数，格式为：

<产品数目> <空格><机器数目><\n>

之后每一行代表一个产品的加工订单，输入‘-1’表示结束输入。

每行的订单输入格式规定如下：

<产品序号><空格><(><按顺序的工序所花时间><,><工序指定机器号><)><空格>...<ENTER>

例如上面表1的加工三个产品的订单输入，可以用键盘输入为：

3 3

1 (7,1) (3,2) (15,3)

2 (10,2) (17,1)

3 (7,1) (22,2)

-1

2.2在输出过程中，随机增加故障

3. 图形界面输入

图形窗口中设计总产品数目和机器数目的输入框，工序所花时间的输入框、工序指定机器号的输入框，工序确认的按钮，下一产品的按钮，以及订单收齐确认按钮。界面元素大致

如图1所示。

初始输入时，首先输入产品总数和机器总数，然后是产品1的加工要求，用户输入工序时间和机器号后，按工序确认可以产生一个工序操作，顺序产生多道工序后可以按“下一产品”按钮进入下一个产品的工序输入。依次操作直到所有产品的加工要求输入完成，按“订单确认”按钮启动程序计算。

The diagram shows a graphical user interface for a job shop scheduling game. It features several input fields and buttons. On the left, there are four input fields: '产品数' (Product Count) with value 5, '机器数' (Machine Count) with value 8, '工序时间' (Process Time) with value 10, and '指定机器' (Specify Machine) with value 1. To the right of these is a large rectangular area labeled '已输入的产品加工订单信息展示' (Display of input product processing order information). At the bottom, there are three buttons: '工序确认' (Process Confirmation), '下一产品' (Next Product), and '订单确认' (Order Confirmation).

图1. Job shop游戏动画版输入界面

三、程序输出

在满足以上约束条件的前提下，安排所输入订单的 n 个产品在 m 台机器上的加工，要求给出产品的各操作在机器上的加工顺序，以及各操作的起始加工时间。

要求开发两个版本：命令行版本和动画版本。具体说明如下：

命令行版本，要求实现文件输入和键盘输入，文件和命令行同时输出。

动画版本，要求实现图形界面输入，文件和动画同时输出。

1. 文件输出，文件名为output.txt，文本文件格式为：

一行代表一个机器的加工序列，最后一行输出 ‘End <最终结束时间>’ 表示方案的最终完成时间。

每行的机器加工序列输出格式规定如下：

<M><机器号><空格><(><起始时间,产品号-工序号,终止时间><)><空格>...<\n>

例如上面表1的加工三个产品的方案输出，文件内容为：

M1 (0,1-1,7) (7,3-1,14) (29,2-2,46)

M2 (7,1-2,19) (19,2-1,29) (29,3-2,51)

M3 (19,1-3,34)

End 51

通过时间轴，动态的打印出各个计划

2. 命令行输出

命令行方式下，一行代表一个机器的加工序列，最后一行输出 ‘End <最终结束时间>’

表示方案的最终完成时间。

每行的机器加工程序输出格式规定如下：

<M><机器号><空格><(><起始时间,产品号-工序号,终止时间><)><空格>...<n>

例如上面表1的加工三个产品的方案输出，命令行显示为：

M1 (0,1-1,7) (7,3-1,14) (29,2-2,46)

M2 (7,1-2,19) (19,2-1,29) (29,3-2,51)

M3 (19,1-3,34)

End 51

3. 图形界面输出

要求在图形窗口中用绘制的甘特图来表示操作在机器上的安排。

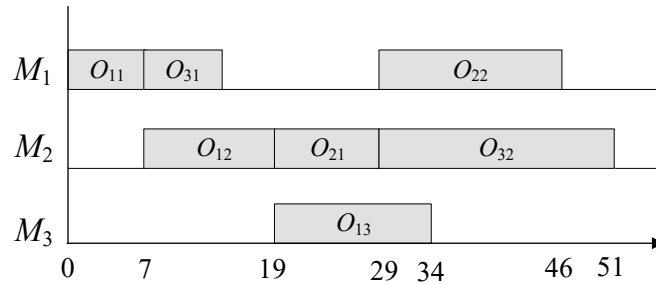


图 2. 加工顺序甘特图

例如，图 2 所示的甘特图给出了上文提及的 3 个产品在 3 台机器上加工的一个安排，其中给出了每台机器上的操作的加工排序以及每个操作的开始加工时间。图 2 所示的加工安排满足产品加工约束和表 1 所示的机器加工约束。例如，产品 1 的第 1 个操作加工完毕再加工第 2 个操作，第 2 个操作加工完毕再加工第 3 个操作。产品 1 的第 1、2、3 个操作分别在机器 1、2、3 上加工，加工时间分别是 7、12、15，满足表 1 中的约束。

4. 动画输出

四、核心调度算法

n 个产品在 m 台机器上加工，满足约束条件的加工安排有很多。我们设定最优解为所有产品加工完成所需时间最短，则需要找到一个加工安排，使得所有产品最快完成加工。

要想获得所有产品加工完成时间更短的方案，我们需要借助调度算法。以下是可选的调度算法。

(1) 调度规则算法

当一台机器加工完毕一个操作而变为空闲状态后，通常有多个操作请求在该机器上加工，此时，需要决定该机器下一步加工哪个操作。用调度规则可做出这一决策。调度规则在每次迭代中按照优先规则调度一个操作，直至生成一个完整调度。如果要用调度规则生成一个较好的安排，需要借助 Giffler & Thompson 算法。

(2) 邻域搜索算法

采用析取图问题建模，然后基于析取图模型构造解的邻域。邻域搜索的基本思想为：在每次迭代中，选取当前解的邻域中的最好解作为下一个解，只要目标函数值减少，该过程

就不断继续，直到找到局部最优点。典型的邻域搜索算法包括模拟退火、禁忌搜索算法。

(3) 演化算法

需要将问题的解表示为一个编码，用一个群体来表示一组解。群体经过若干世代的进化，最终产生的最优个体即为问题的最优解。

第 1 种算法相对较简单，对于小规模问题效果较好，但对较大规模问题效果不够理想；第 2、3 种算法较复杂，需要了解一些背景知识，对大规模问题的性能也较好。

建议同学们用关键词“车间调度”、“Job shop 调度”“Job shop scheduling”在学校图书馆电子数据库中查找并阅读相关文献，然后选择一种算法实现。

最终，所有参与此游戏开发的小组会将结果：程序运行时间、最优解方案加工时间两个时间值按一定权重进行评比，形成一个排行榜。所以选择算法时，要综合考虑最短的算法运行时间和解决方案最优，两者平衡。

五、实现要求

1. 以小组为单位完成上述任务要求，只要能完成输入订单的方案输出，满足约束规则，就可通过。

2. 必须完成两个版本：命令行版本和动画版本的开发，按时参加验收。为了保证验收，要求各小组的程序运行时间不能超过 10 分钟。

3. 必须在指定时间内提交概要设计文档和程序代码。

六、评价规则

所有参与此游戏开发的小组会将结果：程序运行时间、最优解方案加工时间两个时间值按一定权重进行评比，形成一个排行榜。

大作业成绩会基于这个排行榜得分。另外再综合软件的扩展功能、程序代码结构、小组协作度等附加项，形成最终分数。

新加的需求：

1. 根据加工计划，模拟加工过程
同时对命令行和动画版本生效
2. 加工过程中 允许有不定时的机器检修请求
(这个绝对是坑爹)

这样输入信息有两部分：

第一部分：

初始化信息：包括产品数目，机器数目加工订单表

检修命令：任意时刻 对任意机器的检修命令请求