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）加工订单，每个产品都会指定加工步骤（Operation），必须按照，，…，的顺序加工。其中为产品包含的加工步骤。每道工序（步骤）必须在指定的机器(Machine)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，，；第2个产品有2个操作，；第3个产品有2个操作，。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，只有当工序加工完毕，才能开始加工。

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。

**约束3:** 程序必须在**指定时间**内（老师在验收时现场指定）计算出一个最优方案并输出。若使用迭代算法会不断产生许多解，需要在指定时间到达时程序自动比较找出最优解作为最终方案进行输出。

1. 附加任务:模拟加工过程和接受随机检修指令任务

动画版本要求在给出加工计划的前提下，启动模拟加工过程，每隔一个周期时间就输出各机器的加工状态；并在模拟过程中随时接受键盘方式输入的机器检修指令（次数不限），并合理地将检修时间插入到已有的加工计划中。

1. 根据之前算出的加工计划，模拟加工过程，程序运行时间每秒对应**指定的加工时间比**（整数，默认为1，可修改，老师验收时现场指定）。

按照时间顺序，周期性（每隔2秒）动态展示整个加工过程，即每到一个时间点，动态输出每台机器当前的状态信息。

内容包括：折算为加工时间单位的时间值、机器号（从0开始）、机器状态（空闲、检修、加工）、当前工件号（从0开始）、工序号（从0开始）、当前状态已持续时长、当前状态需完成时长。

输出要求，具体见3.3图形界面输出说明。

2. 模拟加工过程中，必须实时接受键盘输入的机器检修指令。

任意时刻、任意机器均可提出设备检修请求，检修时间随检修需求确定（随机值）；如果指定检修的机器正在处理任务，则需要等到当前任务处理完毕后再进入检修期。

具体的检修指令格式，请看2.2说明。

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，文件和命令行同时输出。

动画版本，要求实现文件输入、键盘输入，文件和动画同时输出。

**1. 文件输入订单** 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。每行的订单输入格式规定如下：

<工序指定机器号><空格><工序加工时长><空格>...<\n>

注意：机器号从0开始。

例如，加工15个产品10道工序的订单，输入文件内容为：

15 10

8 14 4 75 3 12 2 38 0 76 5 97 9 12 1 29 7 44 6 66

5 38 3 82 2 85 4 58 6 87 9 89 0 43 1 80 7 69 8 92

9 5 1 84 0 43 6 48 4 8 7 7 3 41 5 61 8 66 2 14

2 42 1 8 0 96 5 19 4 59 7 97 9 73 8 43 3 74 6 41

6 55 2 70 3 75 8 42 4 37 7 23 1 48 5 5 9 38 0 7

8 9 2 72 7 31 0 79 5 73 3 95 4 25 6 43 9 60 1 56

0 97 2 64 3 78 5 21 4 94 9 31 8 53 6 16 7 86 1 7

3 86 7 85 9 63 0 61 2 65 4 30 5 32 1 33 8 44 6 59

2 44 3 16 4 11 6 45 1 30 9 84 8 93 0 60 5 61 7 90

7 36 8 31 4 47 6 52 0 32 5 11 2 28 9 35 3 20 1 49

8 20 6 49 7 74 4 10 5 17 3 34 0 85 2 77 9 68 1 84

1 85 5 7 8 71 6 59 4 76 0 17 3 29 2 17 7 48 9 13

2 15 6 87 7 11 1 39 4 39 8 43 0 19 3 32 9 16 5 64

6 32 2 92 5 33 8 82 1 83 7 57 9 99 4 91 3 99 0 8

4 88 7 7 8 27 1 38 3 91 2 69 6 21 9 62 5 39 0 48

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. 键盘输入检修指令**

动画版本中要求根据文件输入订单计算出加工计划后，启动模拟加工过程。在模拟加工过程结束之前，随时接受键盘输入的检修指令，不限次数，不限机器。

检修指令的格式为：

<C><机器号> <空格><检修时长><\n>

表示在当前时间发出对指定机器进行检修的指令，程序必须计算当前检修指令对应的加工时间，按程序时钟计算，公式=（当前clock-模拟开始clock）\*指定的加工时间比。

注意机器号从0开始，检修时长按订单加工时间单位。

### 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，文件和命令行同时输出。

动画版本，要求实现文件输入和键盘输入，文件和动画同时输出。

**1. 文件输出加工计划，**文件名为output.txt，文本文件格式为：

首先输出加工计划的每台机器加工序列，格式是一行代表一个机器的加工序列，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>

所有机器的加工序列输出完成后，接着输出计算方案的程序运行时间和方案的总完工时间，分两行，格式规定如下：  
 <Time Used:><计算出本方案的运行时间秒值><\n>

<End Time:><总完工时间><\n>

例如：上面2.1输入示例的加工15个产品10道工序的订单，调度方案输出文件内容为：

**M0 (0,6-0,97) (97,3-2,193) (193,0-4,269) (269,7-3,330) (330,2-2,373) (382,10-6,467) (467,11-5,484) (484,5-3,563) (563,1-6,606) (606,9-4,638) (638,8-7,698) (698,12-6,717) (912,14-9,960) (960,4-9,967) (967,13-9,975)**

**M1 (42,3-1,50) (50,11-0,135) (135,2-1,219) (219,14-3,257) (257,8-4,287) (287,12-3,326) (395,13-4,478) (478,0-7,507) (606,1-7,686) (699,4-6,747) (747,7-7,780) (780,10-9,864) (864,6-9,871) (871,9-9,920) (921,5-9,977)**

**M2 (0,3-0,42) (42,12-0,57) (57,8-0,101) (101,0-3,139) (139,13-1,231) (231,1-2,316) (316,6-1,380) (380,5-1,452) (452,4-1,522) (522,11-7,539) (539,10-7,616) (616,7-4,681) (681,9-6,709) (709,14-5,778) (801,2-9,815)**

**M3 (0,7-0,86) (89,0-2,101) (101,1-1,183) (183,8-1,199) (257,14-4,348) (348,10-5,382) (382,6-2,460) (484,11-6,513) (522,4-2,597) (597,2-6,638) (638,5-5,733) (733,12-7,765) (765,3-8,839) (839,9-8,859) (859,13-8,958)**

**M4 (14,0-1,89) (89,14-0,177) (199,8-2,210) (212,3-4,271) (271,10-3,281) (316,1-3,374) (374,11-4,450) (450,12-4,489) (489,9-2,536) (536,2-4,544) (544,6-4,638) (639,4-4,676) (681,7-5,711) (733,5-6,758) (758,13-7,849)**

**M5 (0,1-0,38) (135,11-1,142) (193,3-3,212) (231,13-2,264) (281,10-4,298) (298,0-5,395) (460,6-3,481) (563,5-4,636) (638,9-5,649) (649,2-7,710) (711,7-6,743) (743,8-8,804) (804,12-9,868) (868,4-7,873) (873,14-8,912)**

**M6 (0,13-0,32) (43,10-1,92) (92,12-1,179) (210,8-3,255) (255,4-0,310) (310,11-3,369) (374,1-4,461) (461,2-3,509) (536,9-3,588) (588,0-9,654) (735,6-7,751) (778,14-6,799) (799,5-7,842) (842,3-9,883) (883,7-9,942)**

**M7 (0,9-0,36) (86,7-1,171) (177,14-1,184) (184,10-2,258) (258,12-2,269) (271,3-5,368) (452,5-2,483) (483,13-5,540) (540,0-8,584) (584,2-5,591) (591,11-8,639) (676,4-5,699) (699,1-8,768) (768,6-8,854) (854,8-9,944)**

**M8 (0,0-0,14) (14,5-0,23) (23,10-0,43) (184,14-2,211) (211,11-2,282) (282,9-1,313) (313,13-3,395) (395,8-6,488) (488,3-7,531) (597,4-3,639) (639,12-5,682) (682,6-6,735) (735,2-8,801) (801,7-8,845) (845,1-9,937)**

**M9 (0,2-0,5) (171,7-2,234) (287,8-5,371) (371,3-6,444) (444,0-6,456) (461,1-5,550) (550,13-6,649) (649,6-5,680) (680,10-8,748) (748,9-7,783) (783,12-8,799) (799,14-7,861) (861,5-8,921) (921,4-8,959) (959,11-9,972)**

**Time Used: 43.290000s**

**End Time: 977**

**2. 命令行输出加工计划**

命令行方式下输出加工计划，格式同文件输出加工计划。

**3. 图形界面输出**

* 要求在图形窗口中显示五部分内容：

1. 输入的订单；
2. 计算出的调度方案，以甘特图形式画出；
3. 模拟加工过程的周期性输出；
4. 实时显示键盘输入的机器检修指令；
5. 加入了机器检修后的调度方案，以甘特图形式画出；

图形界面自行设计，但必须包括上述五部分内容。

图2给出了一种图形界面显示方案，其中模拟加工过程的周期性输出和插入检修后的甘特图绘制共用一块区域。

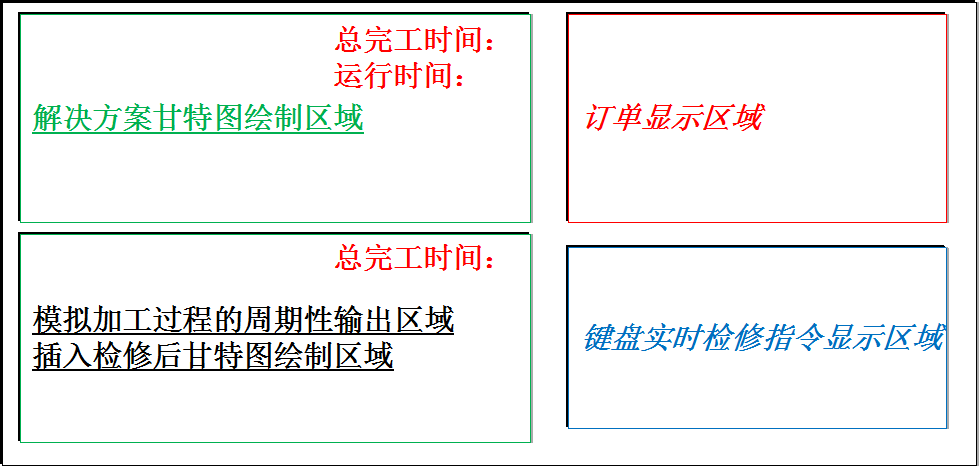


图2. jobshop图形界面设计

* 绘制的甘特图必须表示出所有加工操作在每台机器上的时间序列安排，如图3所示。甘特图中按照时间顺序(从左至右)，以图形方式输出。当有检修命令时，在对应机器的时间线上，画出“检修矩形区域”（颜色不同于“加工矩形区域”）。



46

51

34

29

19

7

0

图3. 加工计划的甘特图

例如，图3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。

**注意，机器号、产品号、工序号都应从0开始。**

* 增加的模拟加工过程信息输出，可以两种方式实现。一种是在窗口的指定区域内以文本形式输出周期性的机器状态信息；另一种方式是按模拟的时间推进，动态绘制甘特图的形式也能更直观地显示机器状态。各小组自行选定一种方式实现。

其中，文本形式的周期性机器状态信息，指定格式如下：

每隔2秒，输出所有机器的状态信息，每台机器一行。最后输出一个空行，表示本次打印结束。

每台机器的状态信息格式规定如下：

<折算为加工时间单位的时间值><空格><机器号><空闲 / 检修 / 加工><空格><当前产品号><-><工序号><空格><当前状态已持续时长><当前状态需完成时长><\n>

注意：机器状态若是空闲或检修，后续的产品号和工序号都不需要，但最好留出空白占位符，方便查看。

### 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

采用析取图为问题建模，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。