同顺序$M\times N$排序问题的动态规划方法

大数据学院 和泳毅 PB19010450

排序论（Scheduling Theory）是组合最优化理论中一个应用十分广泛的领域，而同顺序$m\times n$排序问题则是众多的排序模型中一个成果较多的模型 。1954年，S.M.Johnson[1]给出$m=2$情形的解法，之后许多作者试图把Johnson算法推广到$m\geq 3$的情形。但1976年Garey等人证明了$m\geq 3$情形是一个NP-C问题，这样找到一个解决该问题的好算法是难以实现的。近年来，$m\geq 3$的$m\times n$排序问题的研究，主要在如下几个方面：

1. 确定相邻零件的次序
2. 优先与消去准则
3. 分支定界算法
4. 特殊算法和近似算法

本文将在Johnson条件与分支定界算法的基础上，结合快速的动态规划的方法求解该问题。

一、问题描述

在排序问题中，服务机构或作业设施统称为“机器”，而被服务的顾客或工程建设任务统称为“零件”。同顺序$m\times n$排序问题是考虑如下的“流水作业系统”：n个零件$J_1,J_2,\cdots,J_n$在m台机器$M_1,M_2,\cdots,M_m$上加工，要求每个零件皆依同一顺序（约定为$M_1,M_2,\cdots,M_m$的顺序）通过m台机器，每台机器也依同一顺序（待求的零件顺序X）加工n个零件，并且每台机器一次只能加工一个零件。已知每个零件在各机器上的加工时间，需要确定一个零件加工顺序X，使得总工时$T(X)$（从M_1开工到M_m完工的时间）为最小。

定义1　设a_{ij}为机器M_i加工零件J_j的加工时间，设$X=(x_1,x_2,\cdots,x_n)$为零件下标$1,2,\cdots,n$的一个全排列，则序列X称为一个可行加工顺序。对于给定可行加工顺序X，可以将a_{ix_j}排成$m\times n$的工时矩阵$A(X)$：

$$A(X)=\begin{pmatrix} a_{1x_1} & a_{1x_2} & \cdots & a_{1x_n} \\ a_{2x_1} & a_{2x_2} & \cdots & a_{2x_n} \\ \vdots & \vdots & & \vdots \\ a_{mx_1} & a_{mx_2} & \cdots & a_{mx_n} \end{pmatrix}.$$

定义2　将按照X进行加工作业的工序流线图记为$G(X)$，如下图所示，其中顶点(i,x_j)表示机器M_i加工零件J_{x_j}的工序，工时a_{ix_j}是该顶点的权值。

$$
G(X):\quad
\begin{tikzcd}
(1,x_1) \arrow[r] \arrow[d] & (1,x_2) \arrow[r] \arrow[d] & \cdots \arrow[r] & (1,x_n) \arrow[d] \\
(2,x_1) \arrow[r] \arrow[d] & (2,x_2) \arrow[r] \arrow[d] & \cdots \arrow[r] & (2,x_n) \arrow[d] \\
\vdots \arrow[d] & \vdots \arrow[d] & & \vdots \arrow[d] \\
(m,x_1) \arrow[r] & (m,x_2) \arrow[r] & \cdots \arrow[r] & (m,x_n)
\end{tikzcd}
$$

定义3　设工序(i, x_j)的最早完工时刻为$f(i, x_j)$，并约定

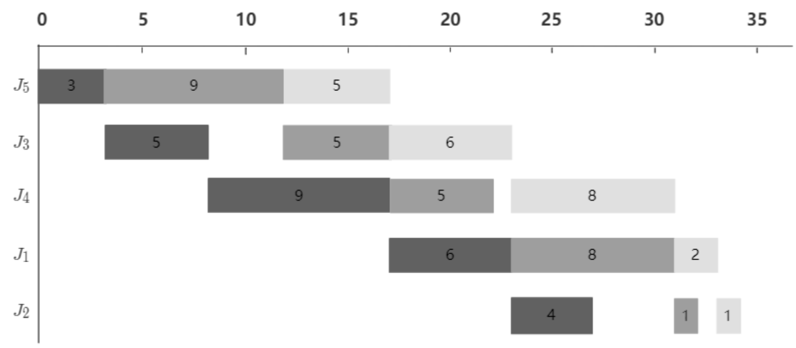
$$f(0, x_j) = f(i, x_0) = 0. \tag{1}$$

综上，所谓同顺序$m \times n$排序问题，就是寻求一个加工顺序X，使总工时$T(X) = f(m, x_n)$为最小。

例子

	J_1	J_2	J_3	J_4	J_5
M_1	6	4	5	9	3
M_2	8	1	5	5	9
M_3	2	1	6	8	5

给定以上零件及对应加工时间，最优总工时为34，最优加工顺序为J_5, J_3, J_4, J_1, J_2。即



二、算法原理

（一）基本方程与基本定理

方程1　机器M_i在加工零件J_{x_j}之前，必须先加工完零件$J_{x_{j-1}}$，并且要等到机器M_{i-1}加工完零件J_{x_j}，故有

$$f(i, x_j) = \max\{f(i, x_{j-1}), f(i-1, x_j)\} + a_{ix_j}. \tag{2}$$

求解最早完工时刻具有最优子结构，初始条件(1)与递推式(2)也是求有向无环图$G(X)$最长路径的动态规划基本方程，可以采用动态规划方法自底向上地求解总工时$T(X)=f(m,x_n)$。

Sum Time
输入：工时矩阵A与加工顺序X。
(0) 初始化$m\times n$的矩阵f，令$f(0,x_j)=f(i,x_0)=0$。$i:=1$。
(1) $j:=1$。若$i>m$，则迭代结束。
(2) 计算$f(i,x_j)=\max\{f(i,x_{j-1}),f(i-1,x_j)\}+a_{ix_j}$。
(3) $j:=j+1$，若$j>m$，则$i:=i+1$，回到步骤(1)。否则回到步骤(2)。
输出：最早完工时间$f(m,x_n)$。

以下进一步说明递推式(2)与有向无环图$G(X)$的关系。

定义4 将有向无环图$G(X)$中任何一条从$(1,x_1)$到(m,x_n)的路径称为对应于X的可行线。设l为一条对应于X的可行线，则称$\sum_{(i,j)\in l}a_{ij}$为对应于l的可行和。

引理[3] 设X为一加工顺序，则总加工时间为$T(X)$等于有向无环图$G(X)$的最大可行和：

$$T(X)=\max_{l}\sum_{(i,j)\in l}a_{ij}. \tag{3}$$

证： 先证$m=2$的情况。工时矩阵$A(X)$只有两行，对应到$G(X)$中显然有

$$\max_{l}\sum_{(i,j)\in l}a_{ij}=\max_{1\le s\le n}\{\sum_{j=1}^{s}a_{1x_j}+\sum_{j=s}^{n}a_{2x_j}\}.$$

当$n=1$时，总加工时间是$a_{11}+a_{21}$，该式成立。假设(3)式对n成立，下证(3)式对$n+1$成立。零件$J_{x_{n+1}}$在M_2上经历的过程只有两种：

（1）$J_{x_{n+1}}$在M_2上不需要等待，此时有

$$\begin{aligned}T(x_1,\cdots,x_{n+1})&=\sum_{j=1}^{n+1}a_{1x_j}+a_{2x_{n+1}}\\&\ge T(x_1,\cdots,x_n)+a_{2x_{n+1}},\end{aligned}$$

故有

$$T(x_1,\cdots,x_{n+1})=\max\{T(x_1,\cdots,x_n)+a_{2x_{n+1}},\sum_{j=1}^{n+1}a_{1x_j}+a_{2x_{n+1}}\}.$$

(2) $J_{x_{n+1}}$在M_2上需要等待，此时有

$$\begin{aligned} T(x_1,\cdots,x_{n+1}) &= T(x_1,\cdots,x_n)+a_{2x_{n+1}} \\ &\geq \sum_{j=1}^{n+1} a_{1x_j}+a_{2x_{n+1}}, \end{aligned}$$

同样可以得到

$$T(x_1,\cdots,x_{n+1})=\max\{T(x_1,\cdots,x_n)+a_{2x_{n+1}},\sum_{j=1}^{n+1} a_{1x_j}+a_{2x_{n+1}}\}.$$

由归纳法，即得

$$T(x_1,\cdots,x_{n+1})=\max_{1\leq s\leq n}\{\sum_{j=1}^{s} a_{1x_j}+\sum_{j=s}^{n+1} a_{2x_j},\sum_{j=1}^{n+1} a_{1x_j}+a_{2x_{n+1}}\}.$$

故当$m=2$时，引理得证。

再对m做归纳，假设(3)式对m成立。记J_{x_j}在机器M_m上的加工完毕时刻为T_j，有$T_1<T_2<\cdots<T_n$。T_j-T_{j-1}为J_{x_j}在M_m上的加工时间，有

$$T(x_1,\cdots,x_{n+1})=\max_{1\leq s\leq n}\{T_s+\sum_{j=s}^{n} a_{m+1x_j}\}.$$

由归纳假设，有

$$T_s=\max_{l_s}\sum_{(i,j)\in l_s} a_{ij},$$

其中l_s是连接a_{1x_1}和a_{mx_s}的一可行线，联立上两式得

$$T(x_1,\cdots,x_n)=\max_{1\leq s\leq n}\{\max_{l_s}\sum_{(i,j)\in l_s} a_{ij}+\sum_{j=s}^{n} a_{m+1x_j}\}.$$

因连接a_{1x_1}和a_{m+1x_s}的任一可行线都可以表示为$\{l_s,(m+1,x_s),\cdots,(m+1,x_n)\}$的形式，故上式即为$m+1$个机器时的(3)式。引理得证。

定义5 （1）设$N=\{1,2,\cdots,n\}$，若$\{s_1,s_2,\cdots,s_k\}\in N$，则称排列$S=(s_1,s_2,\cdots,s_k)$为$N$的一个部分序列。若$S=(s_1,s_2,\cdots,s_k)$和$S'=(s'_1,s'_2,\cdots,s'_k)$为两个部分序列，且$S\cap S'=\phi$，则$SS':=(s_1,s_2,\cdots,s_k,s'_1,s'_2,\cdots,s'_k)$也是一个部分序列，称为$S$和$S'$的连接。

（2）定义极值函数$t_{pq}(S)$表示工时矩阵

$$\begin{pmatrix} a_{ps_1} & a_{ps_2} & \cdots & a_{ps_k} \\ a_{p+1s_1} & a_{p+1s_2} & \cdots & a_{p+1s_k} \\ \vdots & \vdots & & \vdots \\ a_{qs_1} & a_{qs_2} & \cdots & a_{qs_k} \end{pmatrix} (1 \le p \le q \le m)$$

对应流线图的最大可行和，约定$t_{pq}(\phi) = 0$。

至此可得方程1即是计算完工时刻的递推式，也是求有向无环图$G(X)$最长路径的动态规划基本方程，并且总工时$T(X) = t_{1m}(X)$。

方程2[2] 若加工顺序$X = SS', S \cap S' = \phi$，则

$$t_{1m}(X) = \max_{1 \le p \le m} \{t_{1p}(S) + t_{pm}(S')\}. \qquad (4)$$

定理1[2] 设S, S', S''是任意互不相交的部分序列，$i, j \in N \setminus (S \cap S' \cap S'')$。

(1)若

$$t_{1p}(Sij) \le t_{1p}(Sji), \quad 1 \le p \le m, \qquad (5)$$

则$T(SijS') \le T(SjiS')$。

(2)若

$$t_{1p}(Sij) \le t_{1p}(Sj) + \min_{p \le q \le m} a_{qi}, \quad 1 \le p \le m, \qquad (6)$$

则$T(SijS'S'') \le T(SjS'iS'')$。

（3）若

$$t_{1p}(Si) \le t_{1p}(Sj) + \min_{p \le q \le r \le m} \{t_{qr}(i) - t_{qr}(j)\}, \quad 1 \le p \le m, \ (7)$$

则$T(SiS'jS'') \le T(SjS'iS'')$。

证：（1）由方程2与条件(5)得

$$t_{1m}(SijS') = \max_{1 \le p \le m} \{t_{1p}(Sij) + t_{pm}(S')\} \le \max_{1 \le p \le m} \{t_{1p}(Sji) + t_{pm}(S')\} = t_{1m}(SjiS')$$

(2)由方程2与条件(6)得

$$\begin{aligned} t_{1m}(SijS'S'') &= \max_{1 \le p \le m} \{t_{1p}(Sij) + t_{pm}(S'S'')\} \\ &\le \max_{1 \le p \le m} \{t_{1p}(Sj) + \min_{p \le q \le m} a_{qi} + t_{pm}(S'S'')\} \\ &\le \max_{1 \le p \le m} \{t_{1p}(Sj) + t_{pm}(S'iS'')\} \\ &= t_{1m}(SjS'iS'') \end{aligned}$$

(3)由方程2和条件(7)得

$$\begin{aligned}
t_{1m}(SiS'jS'') &= \max_{1\le p\le m}\{t_{1p}(Si)+t_{pm}(S'jS'')\}\\
&\le \max_{1\le p\le m}\{t_{1p}(Sj)+\min_{p\le q\le r\le m}\{t_{qr}(i)-t_{qr}(j)\}+t_{pm}(S'jS'')\}\\
&\le \max_{1\le p\le m}\{t_{1p}(Sj)+t_{pm}(S'iS'')\}\\
&= t_{1m}(SjS'iS'')
\end{aligned}$$

（二）$m=2$时的常规动态规划方法

定义6 设S是一个部分序列，在一般情况下，M_1开始加工S中的零件时，机器M_2还在加工其他零件，要等时间t后才可使用。这种情况下，完成S中零件所需的最短时间记为$T(S,t)$，流水作业调度问题的最优值为$T(N,0)=T(X)$。

定理2[9]（最优子结构）设$X=(x_1,x_2,\cdots,x_n)$是零件$J_i,i\in N$的一个最优加工顺序，则$\hat{X}=(x_2,\cdots,x_n)$是零件$J_i,i\in N\setminus\{x_1\}$的一个最优加工顺序。

证： 把顺序X的加工时间分为a_{1x_1}和T'。其中T'是机器M_1和M_2加工$\hat{X}=(x_2,\cdots,x_n)$的时间。令$S=N\setminus\{x_1\}$，假设$\hat{X}$不是零件集S的最优加工顺序，令$X'=(x_2',\cdots,x_n')$为其最优加工顺序。则机器M_2要等待a_{2x_1}的时间才能加工零件集S，于是最优时间为$T(S,a_{2x_1})$。

并且由于$(x_1,x_2',\cdots,x_n')$也是零件集N的一个加工顺序，其时间为$a_{1x_1}+T(S,a_{2x_1})$。X’是最优加工顺序，则$T(S,a_{2x_1})\le T'$，故有

$$a_{1x_1}+T(S,a_{2x_1})\le a_{1x_1}+T'$$

这与X是零件集N的最优加工顺序矛盾，即$\hat{X}$是零件集S的最优加工顺序。

方程3[9] 由上述最优子结构性质知

$$T(N,0)=\min_{1\le i\le n}\{a_{1x_i}+T(N\setminus\{x_i\},a_{2x_i})\}$$

一般情况下：

$$T(S,t)=\min_{x_i\in S}\{a_{1x_i}+T(S\setminus\{x_i\},a_{2x_i}+\max\{t-a_{1x_i},0\})\} \quad (8)$$

详细迭代过程如下：

FindMinTime

输入：工时矩阵A，零件集合S，等待时间t。

(0) 如果$S=\phi$，$T(S,t):=t$，算法结束。

(1) 记$\bar{S}_i=S\setminus\{x_i\}$，$x_i\in S$。

(2) 递归计算$T(S,t)=\min_{i\in I(S)}\{a_{1x_i}+\mathbf{FindMinTime}(A,\bar{S}_i,a_{2x_i}+\max\{t-a_{1x_i},0\})\}$。

输出：$T(S,t)$。

DP

输入：$2\times n$的工时矩阵A。

(0) $MAXNUM=2^n$，自定义最大等待时间$MAXTIME$。

(1) 构造零件的全组合序列$S_1,\cdots,S_{MAXNUM}$，且保证序列长度非递减，i:=1。

(2) $t:=0$。

(3) 计算$T(S_i,t)=\textbf{FindMinTime}(A,S_i,t)$。

(4) $t:=t+1$，如果$t\le MAXTIME$，回到步骤(3)；否则$i:=i+1$，如果$i\le MAXNUM$，回到步骤(2)，否则迭代结束。

输出：最优总工时$T(N,0)=T(S_{MAXNUM},0)$。

（三）确定相邻零件次序的方法

1. $m=2$时的最优加工顺序

命题1[1] 当$m=2$时，设$X=SijS'$，$X^*=SjiS'$。若

$$\min\{a_{1i},a_{2j}\}\le\min\{a_{1j},a_{2i}\},\tag{9}$$

则$T(X)\le T(X^*)$。

证： 由条件(9)得到

$$t_{12}(ij)=a_{1i}+\max\{a_{1j},a_{2i}\}+a_{2j}\le a_{1j}+\max\{a_{1i},a_{2j}\}+a_{2j}=t_{12}(ji).$$

而$t_{22}(ij)=t_{22}(ji)$，故

$$t_{12}(Sij)=\max_{1\le p\le 2}\{t_{1p}(S)+t_{p2}(ij)\}\le\max_{1\le p\le 2}\{t_{1p}(S)+t_{p2}(ji)\}=t_{12}(Sji).$$

又因$t_{11}(Sij)=t_{11}(Sji)$，所以条件(5)成立，从而$T(X)\le T(X^*)$。

若从动态规划方程角度考虑，设$X=(i,j,\cdots)$是部分零件集S在机器M_2等待时间为t时的任一最优加工顺序，由方程3得到

$$T(S,t)=a_{1i}+T(S\setminus\{i\},a_{2i}+\max\{t-a_{1i},0\})=a_{1i}+a_{1j}+T(S\setminus\{i,j\},t_{ij}),$$

其中

$$\begin{aligned}t_{ij}&=a_{2j}+\max\{a_{2i}+\max\{t-a_{1i},0\}-a_{1j},0\}\\&=a_{2j}+a_{2i}-a_{1j}+\max\{\max\{t-a_{1i},0\},a_{1j}-a_{2i}\}\\&=a_{2j}+a_{2i}-a_{1j}+\max\{t-a_{1i},a_{1j}-a_{2i},0\}\\&=a_{2j}+a_{2i}-a_{1j}-a_{1i}+\max\{t,a_{1i}+a_{1j}-a_{2i},a_{1i}\}.\end{aligned}$$

交换i和j的顺序后，时间为

$$T'(S,t)=a_{1i}+a_{1j}+T(S\setminus\{i,j\},t_{ji}),$$

其中

$$t_{ji} = a_{2j} + a_{2i} - a_{1j} - a_{1i} + \max\{t, a_{1i} + a_{1j} - a_{2j}, a_{1j}\}.$$

当i和j满足条件(9) $\min\{a_{1i}, a_{2j}\} \leq \min\{a_{1j}, a_{2i}\}$时，有

$$\max\{-a_{1j}, -a_{2i}\} \leq \max\{-a_{1i}, -a_{2j}\},$$

从而

$$a_{1i} + a_{1j} + \max\{-a_{1j}, -a_{2i}\} \leq a_{1i} + a_{1j} + \max\{-a_{1i}, -a_{2j}\}.$$

由此可得

$$\max\{a_{1i}, a_{1i} + a_{1j} - a_{2i}\} \leq \max\{a_{1j}, a_{1i} + a_{1j} - a_{2j}\},$$

对任意t有

$$\max\{t, a_{1i}, a_{1i} + a_{1j} - a_{2i}\} \leq \max\{t, a_{1j}, a_{1i} + a_{1j} - a_{2j}\}.$$

从而$t_{ij} \leq t_{ji}$，即$T(S,t) \leq T'(S,t)$。

由命题1可以得到如下的**Johnson**算法：

Johnson算法

输入：$2 \times n$的工时矩阵A。

(0) 初始化长度为n的空序列X，$front := 1$，$back := n$。

(1) 在工时矩阵A中找到最小元素（若不唯一则任择一），若此数位于a_{1i}, 进入步骤(2)，若此数位于a_{2j}, 进入步骤(3)。若所有元素都为$+\infty$，迭代终止。

(2) 令$X[front] = J_i$，$front := front + 1$，并置$a_{1i} = a_{2i} = +\infty$。返回步骤(1)。

(3) 令$X[back] = J_j$，$back := back - 1$，并置$a_{1j} = a_{2j} = +\infty$。返回步骤(1)。

输出：最优加工顺序X。

并且，从任一加工顺序出发，按上述算法，把每排定一个零件看作通过逐次对换相邻零件将其移至首位或末位。由命题1知每次对换时总工时不增加，因此最后得到的加工顺序必定总工时最小。

命题2[1]（Johnson条件）当$m = 2$，设$X = (1, 2, \cdots, n)$。若

$$\min\{a_{1i}, a_{2j}\} \leq \min\{a_{1j}, a_{2i}\}, \quad 1 \leq i \leq j \leq n \quad (10)$$

则X为最优加工顺序。

证： 以满足条件(10)的加工顺序X作为初始顺序，执行上述**Johnson**算法重新确定一个加工顺序。由命题1. 每次对换相邻零件时总工时不变，且保持条件(10)，最后得到最优加工顺序X_0，必有$T(X) = T(X_0)$, 即X也是最优加工顺序。

注意命题2作为最优解的条件是充分的，但不是必要的。

2. $m \geq 3$时确定相邻顺序的条件

命题3[3] 当$m \geq 3$时，设$X = SijS'$，$X^* = SjiS'$。若

$$\min\{a_{ui}, a_{vj}\} \leq \min\{a_{uj}, a_{vi}\}, \quad 1 \leq u < v \leq m \quad (11)$$

则$T(X) \leq T(X^*)$。

证： 对$1 \leq p \leq m$的p运用方程2，得

$$t_{1p}(Sij) = \max_{1 \leq r \leq p} \{t_{1r}(S) + t_{rp}(ij)\},$$
$$t_{1p}(Sji) = \max_{1 \leq r \leq p} \{t_{1r}(S) + t_{rp}(ji)\}.$$

只要将关于i, j的两列矩阵转置，即可看出$t_{rp}(ij)$和$t_{rp}(ji)$分别是如下矩阵的最大可行和：

$$\begin{pmatrix} a_{ri} & a_{r+1i} & \cdots & a_{pi} \\ a_{rj} & a_{r+1j} & \cdots & a_{pj} \end{pmatrix}, \begin{pmatrix} a_{pi} & a_{p-1i} & \cdots & a_{ri} \\ a_{pj} & a_{p-1j} & \cdots & a_{rj} \end{pmatrix}.$$

而条件(11)说明前一矩阵的排列满足命题2（Johnson条件），后一矩阵是前者的逆排序，所以$t_{rp}(ij) \leq t_{rp}(ji)$。因此，$t_{1p}(Sij) \leq t_{1p}(Sji)$，即条件(5)成立，从而$T(X) \leq T(X^*)$。

（四）结合动态规划的分支定界算法

分支定界算法思想和技巧在运筹学中有着广泛的应用，实质上它是不明 显的遍历方法，也就是把所研究的问题的全部可行解分成若干部分，在所有排列所形成的决策树中搜寻最优解，但利用估界作为剪枝准则，尽可能地多消去一些分支，可以加速计算过程。在本问题中，估计零件集$S \cup \{i\}$的总工时下界需要用到零件集S的信息，有着类似于动态规划的子问题结构。

定义7 设S是一个部分序列，令$B(S \cdots)$表示全部$(S \cdots)$形的加工顺序对应总加工时间的下界。S可以是空集。

参考文献[6]，一个完整的决策树共有n层节点，第0层为根节点，从根节点开始分支，产生第1层的n个节点，每个节点对应一个第一个零件已经排好的部分序列。从第1层的一个节点分支可产生第2层$n-1$个节点，每个节点 对应一个两个零件已经排好的部分序列。一般地，从第$r-1$层的一个节点分支可产生第r

层$n-r+1$个节点，第r层的每个节点对应一个前r个零件已经排好的部分序列。确定各节点的下界$B(S\cdots)$。由第$n-1$层的一个节点可以确定一个可行序列$\hat{X}$，其总工时$T(\hat{X})$作为最优加工顺序总工时的一个上界。对下界大于$T(\hat{X})$的节点进行剪枝。对下界不大于$T(\hat{X})$的节点进行分支。不断改进$T(\hat{X})$可以得到最优加工顺序。

下面给出文献[4]和文献[5]用动态规划方程2讨论的一种定界方法。

要估计$B(S\cdots)$的值，设$R=N\setminus S$，由方程2得

$$t_{1m}(SR)=\max_{1\le p\le m}\{t_{1p}(S)+t_{pm}(R)\}.$$

对给定的$p(1\le p\le m)$，可以通过几条特殊的可行线来估计$t_{pm}(R)$的下界。首先，按照工时矩阵A的后两行，根据a_{m-1j}和a_{mj}的值, $j\in R$，将R中的零件按照**Johnson**算法排成序列$(j_1,j_2,\cdots,j_k)$。

方程4[4] 令

$$\begin{aligned}&R_1=(j_1,j_2,j_3,\cdots,j_k), && R_2=(j_2,j_1,j_3,\cdots,j_k)\\ &R_3=(j_3,j_1,j_2,\cdots,j_k),\quad \cdots, && R_k=(j_k,j_1,j_2,\cdots,j_{k-1})\end{aligned}$$

给出方程

$$\begin{cases}b_p=t_{1p}(S)+\max\{\min_{1\le\alpha\le k}\{t_{pm-2}(j_\alpha)+t_{m-1m}(R_\alpha)\},t_{pp}(R)\},1\le p\le m-2,\\ b_{m-1}=t_{1m-1}(S)+t_{m-1m}(R_1),\\ b_m=t_{1m}(S)+t_{mm}(R_1).\end{cases}\tag{12}$$

那么$t_{1p}(S)+t_{pm}(R)\ge b_p\ (1\le p\le m)$，从而

$$B(S\cdots)=\max_{1\le p\le m}\{b_p\}$$

给出详细的估界算法：

Bound算法

输入：工时矩阵A，部分序列S，剩余序列R。

(0) 初始化序列$R_\alpha=j_\alpha(R\setminus j_\alpha)$，$1\le\alpha\le k$，记$A[i:j]$表示矩阵第$i$行至第$j$行构成的子矩阵。

(1) $p:=1$。

(2) 计算$a_p=\min_{1\le\alpha\le k}\{\textbf{Sum Time}(A[p:m-2],j_\alpha)+\textbf{Sum Time}(A[m-1:m],R_\alpha)\}$。

(3) 计算$b_p=\max\{a_p,\textbf{Sum Time}(A[p:p],R)\}+\textbf{Sum Time}(A[1:p],S)$。

(4) $p:=p+1$，如果$1\le p\le m-2$，回到步骤(2)，否则进入步骤(5)。

(5) 计算$b_{m-1}=\textbf{Sum Time}(A[1:m-1],S)+\textbf{Sum Time}(A[m-1:m],R_1)$。

(6) 计算$b_m=\textbf{Sum Time}(A[1:m],S)+\textbf{Sum Time}(A[m:m],R_1)$。

(7) 令$B(S\cdots):=max_{1\le p\le m}\{b_p\}$。

输出：估计下界$B(S\cdots)$。

接下来讨论最优加工顺序总工时的一个上界$T(\hat{X})$的计算，如果按照文献[6]的方法不断更新$T(\hat{X})$，计算量十分大，且效率不高。因此本人考虑采用文献[8]中对该问题给出的启发式算法$\mathbf{WSH}$法求得的近似解作为可行序列$\hat{X}$，并求得固定的$T(\hat{X})$。

WSH算法

输入：工时矩阵A。

(0) 初始化排序指标λ_j，$j = 1, 2, \cdots, n$。

(1) 计算$\lambda_j = \frac{\sum_{i=1}^{m} i \cdot a_{ij}}{\sum_{i=1}^{m} a_{ij}}$。

(2) 按λ_j非增的顺序排列所有零件，得到近似解顺序$\hat{X}$。

输出：近似解顺序$\hat{X}$。

最后完整地给出解决同顺序$M \times N$排序问题的结合动态规划的分支定界算法：

Branch&Bound算法

输入：$m \times n$工时矩阵A，部分序列S。

(0) 启发式最优上界$upper := \mathbf{WSH}(A)$，叶节点顺序集合$Leaf := \phi$。

(1) $X := N \setminus S$，N为全体零件集。

(2) $\bar{X}_i := X \setminus \{x_i\}$，$\bar{S}_i := Sx_i$，$\forall x_i \in X$。

(3) $X_i^* := \mathbf{Johnson}(A[m-1:m, \bar{X}_i])$，$\forall i \in I(X)$。

(4) 估计下界$lower_i := \mathbf{Bound}(A, X_i^*, \bar{S}_i)$，$\forall i \in I(X)$。

(5) 如果$lower_i \leq upper$，$\forall x_i \in X$，则如果$|\bar{X}_i| == 1$，$Leaf := Leaf \cup \{\bar{S}_i \bar{X}_i\}$，如果$|\bar{X}_i| > 1$，令$S := \bar{S}_i$，返回步骤(1)。

(6) 令$X^* = \arg\min_{X \in Leaf} \mathbf{Sum\ Time}(A, X)$，$T(X^*) = \mathbf{Sum\ Time}(A, X^*)$。

输出：最优加工顺序X^*，最优总工时$T(X)^*$。

三、数据集（案例）说明

（一）$m = 2$的测试数据集

1. A_1（$m = 2 \quad n = 5$）

	J_1	J_2	J_3	J_4	J_5
M_1	3	7	4	5	7
M_2	6	2	7	3	4

该数据集最优总工时为28，最优加工顺序为J_1, J_3, J_5, J_4, J_2。

2. A_2（$m=2 \quad n=5$）

	J_1	J_2	J_3	J_4	J_5
M_1	2	4	3	6	1
M_2	5	2	3	1	7

该数据集最优总工时为19，最优加工顺序为J_5, J_1, J_3, J_2, J_4。

（二）$m \geq 3$的测试数据集

1. A_3（$m=3 \quad n=6$）

	J_1	J_2	J_3	J_4	J_5	J_6
M_1	6	12	4	3	6	2
M_2	7	2	6	11	8	14
M_3	3	3	8	7	10	12

该数据集来自文献[6]，该作者使用分支定界法得到最优总工时为57。根据命题3的条件将零件排序，记$J_i < J_j$表示J_i应在J_j之前，则有

$$J_3 < J_5, J_6 < J_4, J_6 < J_1,$$
$$J_5 < J_1, J_4 < J_1, J_3 < J_1.$$

则初步排序为J_3, J_5, J_6, J_4, J_1或J_6, J_4, J_3, J_5, J_1。若将J_2放在最后则有$J_3, J_5, J_6, J_4, J_1, J_2$和$J_6, J_4, J_3, J_5, J_1, J_2$，并且前一序列为最优加工顺序。

2. A_4（$m=3 \quad n=5$）

	J_1	J_2	J_3	J_4	J_5
M_1	6	4	5	9	3
M_2	8	1	5	5	9
M_3	2	1	6	8	5

该数据集来自文献[7]，该作者使用基于Gupta启发式算法的分支定界法得到最优总工时为34，最优加工顺序为J_5, J_3, J_4, J_1, J_2。

3. A_5（$m=3 \quad n=4$）

	J_1	J_2	J_3	J_4
M_1	13	7	26	2
M_2	3	12	9	6
M_3	12	16	7	1

该数据集来自文献[11]，该作者使用分支定界法得到最优总工时为62。根据命题3的条件将零件排序，则有

$$J_2 < J_3.$$

若将J_4放在最后则有J_1, J_2, J_3, J_4和J_2, J_3, J_1, J_4，并且后一序列为最优加工顺序。

4. A_6（$m=3 \quad n=6$）

	J_1	J_2	J_3	J_4	J_5	J_6
M_1	1	8	11	3	5	12
M_2	8	5	2	9	5	7
M_3	4	10	7	2	4	7

该数据集来自文献[10]，该作者使用一种启发式的算法得到近似最优总工时为54。文献[3]使用消去法重新计算得到最优总工时为49，最优加工顺序为$J_1, J_2, J_4, J_6, J_5, J_3$。

5. A_7（$m=3 \quad n=10$）

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
M_1	1	5	7	8	3	7	9	8	6	3
M_2	2	9	6	9	2	10	7	9	1	1
M_3	9	7	8	9	3	4	7	4	3	1

该数据集来自文献[11]，该作者认为是3个机器10个零件的加工顺序中最难的问题。最优总工时为66，最优加工顺序为$J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}$。

四、程序输入输出说明

（一）输入

以数据集A_1为例：

	J_1	J_2	J_3	J_4	J_5
M_1	3	7	4	5	7
M_2	6	2	7	3	4

1. 输入零件的个数和机器的个数，以空格隔开，回车结束，如

```
5 2
```

2. 输入工时矩阵A，按机器的顺序（行序）输入。每行中各元素以空格隔开，每行以回车结束，如

```
3 7 4 5 7
6 2 7 3 4
```

（二）输出

1. 若输入的机器数为2（$m=2$），则输出两种算法的最优总工时与最优加工顺序，以数据集$A1$为例，程序输出：

   $m=2,$ 可采用动态规划方法和$Johnson$算法求解：
   【1】动态规划方法最优总工时：28.0
   【2】$Johnson$算法最优总工时：28.0
   最优加工顺序：
   (1) $J1\ J3\ J5\ J4\ J2$

2. 若输入的机器数大于等于3（$m\geq 3$），则输出结合动态规划的分支定界法的最优总工时与最优加工顺序，以数据集A_3为例，程序输出：

   $m>=3,$ 采用结合动态规划的分支定界法求解：
   最优总工时： 57.0
   最优加工顺序：
   (1) $J3\ J5\ J6\ J2\ J4\ J1$
   (2) $J3\ J5\ J6\ J4\ J1\ J2$
   (3) $J3\ J5\ J6\ J4\ J2\ J1$

五、程序测试结果

（一）$m=2$的测试结果

1. A_1

算法	最优总工时	运行时间
DP	28	3.326ms
Johnson	28	0.045ms

最优加工顺序为J_1,J_3,J_5,J_4,J_2。

2. A_2

算法	最优总工时	运行时间
DP	19	3.306ms
Johnson	19	0.050ms

最优加工顺序为J_5,J_1,J_3,J_2,J_4。

（二）$m \geq 3$的测试结果

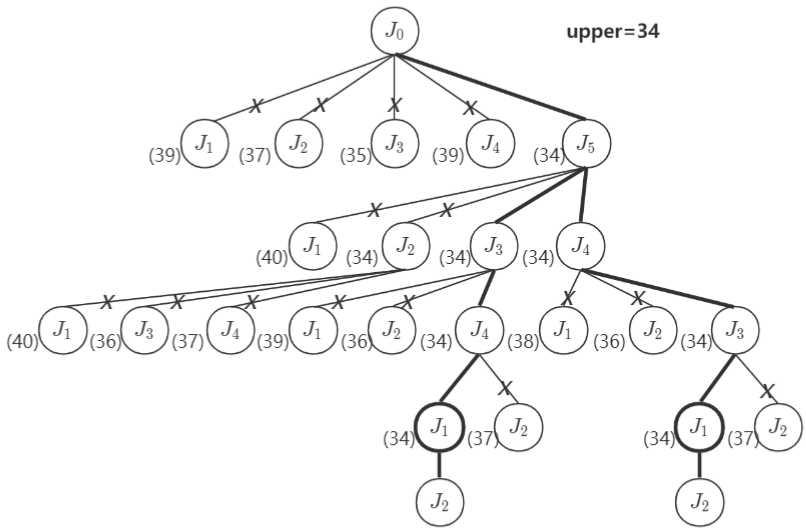
1. A_3

运行时间：53ms，最优总工时57。最优加工顺序：

$$J_3, J_5, J_6, J_2, J_4, J_1 \quad J_3, J_5, J_6, J_4, J_1, J_2 \quad J_3, J_5, J_6, J_4, J_2, J_1$$

2. A_4

使用该例表示测试分支定界法节点生成过程：



运行时间：5.5ms，最优总工时34。最优加工顺序：

$$J_5, J_3, J_4, J_1, J_2 \quad J_5, J_4, J_3, J_1, J_2$$

3. A_5

运行时间：6.5ms，最优总工时62。最优加工顺序：

$$J_2, J_3, J_1, J_4$$

4. A_6

运行时间：54ms，最优总工时49。最优加工顺序：

$$J_1, J_2, J_4, J_6, J_3, J_5 \quad J_1, J_2, J_4, J_6, J_5, J_3$$

5. A_7

运行时间：12.45s，最优总工时66。最优加工顺序：

$$
\begin{array}{ll}
J_1,J_2,J_3,J_4,J_5,J_6,J_7,J_8,J_9,J_{10} & J_1,J_2,J_3,J_4,J_5,J_6,J_7,J_8,J_{10},J_9 \\
J_1,J_2,J_3,J_4,J_5,J_8,J_7,J_6,J_9,J_{10} & J_1,J_2,J_3,J_4,J_5,J_8,J_7,J_6,J_{10},J_9 \\
\multicolumn{2}{c}{\cdots} \\
J_1,J_2,J_8,J_5,J_3,J_4,J_7,J_6,J_9,J_{10} & J_1,J_2,J_8,J_5,J_3,J_4,J_7,J_6,J_{10},J_9
\end{array}
$$

共140个最优加工顺序。

六、分析总结

$m=2$的情形下，常规的动态规划方法和引入Johnson条件改进后的**Johnson**算法都能正确的求解该问题，得到最优总工时。但常规的动态规划方法不易存储中间过程，本文没有实现输出最优加工顺序的功能，但**Johnson**算法可以输出最优加工顺序。并且对常规动态规划方法是指数级的时间复杂度，**Johnson**算法则是线性的时间复杂度。可见改进后无论是在实现难度还是运行时间上都有了很大的进步。

$m\geq 3$的情形下，采用动态规划方程推导出对分支定界法节点下界的估计，并用启发式算法给出最优上界，精确的上下界能极大地提高分支定界法的运行效率，将更多节点进行剪枝。并且该算法除了可以输出最优总工时，还可以输出多个最优加工顺序，在复杂数据集上也有不错的表现。

七、参考文献

[1] Johnson S M . Optimal two- and three-stage production schedules with setup times included[J]. Naval Research Logistics Quarterly, 1954, 1.

[2] 林诒勋. 同顺序M×N排序问题的动态规划方法[J]. 数学进展, 1986(04):337-346.

[3] 越民义, 韩继业. n个零件在m台机床上的加工顺序问题(I)[J]. 中国科学, 1975, 5(5):462.

[4] 越民义, 韩继业. 同顺序m×n排序问题的一个新方法[J]. 科学通报, 1979(18):7-10.

[5] 越民义, 韩继业. 排序问题中的一些数学问题(续)[J]. 数学的实践与认识, 1976(03):61-72.

[6] Lomnicki Z A . A "Branch-and-Bound" Algorithm for the Exact Solution of the Three-Machine Scheduling Problem[J]. OR, 1965, 16(1):89-100.

[7] 谢金华, 叶春明, 马良,等. Flow shop排序问题Fm|prmu|Cmax的改进分枝定界法[J]. 现代制造工程, 2008(3):3.

[8] 沈英俊. 一种求解同顺序 Flowshop排序问题的新的启发式算法[J]. 北京航空航天大学学报, 1998, 000(001):83.

[9] 王晓东. 计算机算法设计与分析(第3版)[M]. 电子工业出版社, 2007.:72-75

[10] Gupta, Jatinder N D . Heuristic Algorithms for Multistage Flowshop Scheduling Problem[J]. A I I E Transactions, 1972, 4(1):11-18.

[11] Ignall E,Schrage L.Application of the Branch and Bound Technique to SomeFlow-Shop Scheduling Problems[J].0Operations Research,1965,13(3):400-412.