由于 θ 的均值是 φ, $\varphi=\sin^{-1}\sqrt{p}$, l 的均值等于 0;由于 θ 的方差是 $1/4n$, l 的方差

$$E[l^2]=E[n(\theta-\varphi)^2]=1/4.$$

这表明了 l 渐近地服从正态分布 $N(0, 1/4)$. 我们留意, l 的渐近分布既不依赖于 p, 又不依赖于 n.

由此推出，实测点 P 在半径 $\sqrt{n}$ 的 1/4 圆周上以均值 P_0 和标准离差 1/2 渐近地服从正态分布. 这里，若 $|\theta-\varphi|$ 足够小, 则弧长 l 可用弦 P_0P 来代替，甚至可用由 P 到分割线 OP_0 的垂直距离来代替. 可见,从实测点 P 顺着半径 $\sqrt{n}$ 的 1/4 圆周 (近似垂直地) 测出的 1/2 的长度相当于正态分布的一 σ. 根据这个事实就可以做种种统计推断, 在统计分析纸上以 1/2 为单位刻划 σ 尺就是这个原因. （续完）

参　考　资　料

[1] 中国科学院数学研究所统计组编,常用数理统计方法,科学出版社,北京，1973.

[2] Fisher R. A., On the dominance ratio, *Proc. Roy. Soc., Edinburgh*, **42** (1922), 321—341

[3] Fisher R. A. and Mather, K., The inheritance of style-length in Lythrum salicaria, *Ann. Eugenics*, **12** (1943), 1—23.

[4] Mosteller F. and Tukey, J. W., The uses and usefulness of binomial probability paper, *Jour. Amer. Statist. Assoc.*, **44** (1949), 174—212.

[5] Masuyama M., *Graphical Method of Statistical Inference*, Maruzen Co., Tokyo, 1954.

[6] 增山元三郎,推计纸の使い方(改订增补版),日本规格协会,东京，1959.

[7] 吉村功,推计纸の使い方ポケットブック,日本规格协会,东京，1966.

排序问题中的一些数学问题(续)

中国科学院数学研究所　越民义　韩继业

三、关于同顺序的两台和三台机器的排序问题

我们现来介绍关于另件在两台和三台机器的加工顺序问题. 机器的数目从一台变成多台,情况就要复杂得多.根据加工工艺的要求,一个另件将按一定的次序通过这些机器,接受它们的加工;另件不相同,通过机器的次序可能也不同. 我们把多台机器的排序问题分为“同顺序”和“不同顺序”两种. 所谓同顺序的多台机器的排序问题, 是指每个另件都是先在第一台机器上加工,然后到第二台机器上加工,……,最后送到最末一台机器加工,而且在每台机器上加工另件的顺序都一致. 不同顺序的排序问题将在下一章再介绍. 在本文中,我们只把“完成全部加工工作所需的时间”做为衡量加工顺序优劣的指标,而不讨论其他的目标函数.

下面先用一个例子来说明同顺序的排序问题.设有两台机器,我们记为 M_1 和 M_2; 有五个另件,我们记为 J_1, J_2, J_3, J_4, J_5. 每个另件都是先在 M_1 上加工,再送到 M_2 上加工. 另件在机器上的加工时间由下表给出:

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

即 J_1 在 M_1 上的加工时间为 3（小时），在 M_2 上的加工时间为 6，等等．如果我们按照顺序 $(J_1\ J_2\ J_3\ J_4\ J_5)$ 来加工另件，那么整个加工过程可用图 1 表示出来．

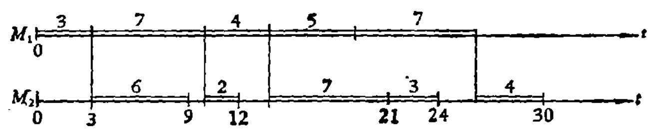

图　1

现在我们将另件加工的顺序改变为 $(J_1\ J_3\ J_5\ J_4\ J_2)$，那么整个过程如图 2 所示．

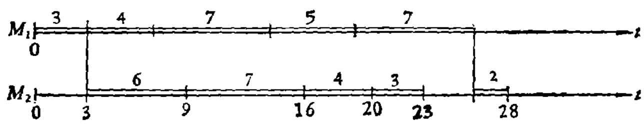

图　2

根据图 1 和图 2 可以看出，若采用前一顺序，则从 M_1 开始加工 J_1 起到 M_2 加工完 J_5 为止所需时间（即完成全部加工工作所需的时间）为 30 小时；若采用后一顺序，则全部加工时间为 28 小时，比前一顺序节约 2 小时．

由此可知，另件的加工顺序不同，完成全部工作的时间就可能不同．为了能多快好省地进行生产，我们希望能在事先将另件进行安排，使得完成全部工作的时间为最短．完成全部工作的时间为最短的顺序，我们称之为“最优顺序”．最优顺序一般不一定只有一个．我们现将问题的一般提法表述如下：

设有 n 个另件 $J_1, J_2, \cdots, J_n$ 要在 m 台机器 $M_1, M_2, \cdots, M_m$ 上加工．这些另件都是先在 M_1 上加工，再到 M_2 上加工，…，最后到 M_m 上加工；而且另件在每台机器上的加工顺序都一致．这些另件在每台机器上的加工时间事先都已知道．设每台机器在同一时间只能加工一个另件；且另件一旦在某台机器上开始加工，则这台机器就不间断地加工完这另件．我们要问：如何将另件安排一个顺序，使得在按照这一顺序进行加工时，完成全部加工工作的时间为最短．

上面这一问题，我们称之为同顺序的 $m \times n$ 排序问题．1954 年约翰逊对于两台机器的情形提出了一种寻求最优顺序的简单易行的解法，即“约翰逊方法”．对于三台以上机器的情形，虽然从事研究的人很多，并且有若干解法被提出来，但这些方法或非常复杂，或只适用于个别情况．因此这问题有待于进一步研究．

§1. 可行线与可行和[1]

这一节的目的在于给出一些基本概念和基本定理．我们知道，将 n 个另件安排一个加工顺序也就是给定 $(1\ 2\ \cdots\ n)$ 的某一个排列．比如说，在上面的例子中按照 $J_1\ J_3\ J_5\ J_4\ J_2$ 的顺序加工，也就是相应地给定了排列 $(1\ 3\ 5\ 4\ 2)$．现在，我们用记号 $\omega = (\omega_1\ \omega_2\ \cdots\ \omega_n)$ 表示 $(1\ 2\ \cdots\ n)$ 的一个排列．设 J_i 在 M_k 上的加工时间为 a_{ki}．若另件的加工顺序是 $(J_{\omega_1}\ J_{\omega_2}\ \cdots\ J_{\omega_n})$，我们将 a_{ki} 排成矩阵

$$A(\omega)=\begin{bmatrix} a_{1\omega_1} & a_{1\omega_2} & \cdots & a_{1\omega_n} \\ a_{2\omega_1} & a_{2\omega_2} & \cdots & a_{2\omega_n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m\omega_1} & a_{m\omega_2} & \cdots & a_{m\omega_n} \end{bmatrix} \tag{1.1}$$

对于上述例子来说，当 $\omega=(1\ 3\ 5\ 4\ 2)$ 时，

$$A(1\ 3\ 5\ 4\ 2)=\begin{bmatrix} 3 & 4 & 7 & 5 & 7 \\ 6 & 7 & 4 & 3 & 2 \end{bmatrix}$$

定义 1. 可行线. 将矩阵 $A(\omega)$ 中 $a_{1\omega_1}$ 和 $a_{m\omega_n}$ 用一条折线连接起来，此折线只能以一些 a_{ki} 为顶点，并且只能向右或向下延伸. 这种折线我们称之为 $A(\omega)$ 的一条可行线.

在上例中，连接数字 3, 6, 7, 4, 3, 2 或连接 3, 4, 7, 5, 3, 2 的折线就是 $A(1\ 3\ 5\ 4\ 2)$ 的可行线.

对每一 ω，令 $\{l(\omega)\}$ 表示相应的全体可行线所成之集合.

定义 2. 可行和. 设 $l(\omega)$ 为 $A(\omega)$ 的一条可行线，则称此线上的所有 a_{ki} 加起来所得之和为相应的可行和. 也就是可行和为 $\sum\limits_{(ki)\in l(\omega)} a_{ki}$. 当 ω 固定时，符号 $\max\limits_{l(\omega)} \sum\limits_{(ki)\in l(\omega)} a_{ki}$[1)]表示相应于 $A(\omega)$ 的最大可行和.

在上述例子中，若 $\omega=(1\ 3\ 5\ 4\ 2)$，则相应于 $A(1\ 3\ 5\ 4\ 2)$ 的所有可行和为

$$3+6+7+4+3+2=25,$$
$$3+4+7+4+3+2=23,$$
$$3+4+7+5+3+2=24,$$
$$3+4+7+5+7+2=28.$$

于是最大可行和为 28. 可行和这一概念甚为重要，它与机器完成全部加工工作的时间有如下关系:

定理 1. 设 $\omega=(\omega_1, \cdots, \omega_n)$ 为一加工顺序，则从 M_1 开始加工 J_{ω_1} 起到 M_m 加工完 J_{ω_n} 为止这段时间之长 $T(\omega)$ 等于矩阵 $A(\omega)$ 的最大可行和，即

$$T(\omega)=\max_{l(\omega)} \sum_{(ki)\in l(\omega)} a_{ki}. \tag{1.2}$$

此定理的证明可查阅[1]中第一节的引理.

§ 2. 两台机器的最优加工顺序[2]

下面介绍寻求最优顺序的约翰逊方法.

(2.1) 解法.

第一步. 在矩阵 $A(\omega)$ 的上下两行中找出最小的数字(若最小数字不止一个，可任择其一). 若此数字在上行，则将相应的 J_i 排在最前位置; 若此数字在下行，则将相应的 J_i 排在最后位置. 转入下一步.

在上述例子中最小数字是 2，在下行，故将相应的 J_2 排在最后，得 $(J_1\ J_3\ J_4\ J_5\ J_2)$.

第二步. 将已排定位置的另件从 $A(\omega)$ 中划去，然后对于剩下的另件重复第一步. 如此继续下去，直到把全部另件都排完. 此时就得到最优顺序.

1) 一般而言，设 S 为一集合，若对于 S 中任一元素 a，相应地有一函数值 $f(a)$，则我们用符号 $\max\limits_{a\in S} f(a)$ 表示函数 $f(x)$ 在 S 上的极大值. 在不引起混淆的情况下，我们常简写作 $\max\limits_{a} f(a)$.

在本例中,在划去 J_2 之后,上下两行中的最小数字为 3, 相应的另件为 J_1 和 J_4. 按照第一步, J_1 应排在最前面; J_4 应排在最后面(注意, 在 J_2 之前). 现在只剩下 J_3 和 J_5. 它们的加工时间中最小数字是 4,相应的 J_3 应排在前面(J_1 之后);相应的 J_5 应排在后面(J_4 之前). 最后得到最优顺序(1 3 5 4 2).

(2.2) 最优性证明.

今后我们将另件在 M_1 上的加工时间记为 a,在 M_2 上的加工时间记为 b,则得下表

	J_1	J_2	$\cdots$	J_{i-1}	J_i	J_{i+1}	$\cdots$	J_n
M_1	a_1	a_2	$\cdots$	a_{i-1}	a_i	a_{i+1}	$\cdots$	a_n
M_2	b_1	b_2	$\cdots$	b_{i-1}	b_i	b_{i+1}	$\cdots$	b_n

设 l 是连接 $a_1\ a_2 \cdots a_i\ b_i\ b_{i+1}\ \cdots\ b_n$ 的一可行线, 相应的可行和为 $a_1 + a_2 + \cdots + a_i + b_i + b_{i+1} + \cdots + b_n = \sum_{r=1}^{n} b_r + (a_1 - b_1) + \cdots + (a_{i-1} - b_{i-1}) + a_i$. 由定理 1 可知, 对应于排列 $\omega = (1, \cdots, i-1, i, i+1, \cdots, n\}$ 而言,完成全部工作的时间为

$$T(\omega) = \sum_{r=1}^{n} b_r + \max_{1 \leqslant r \leqslant n} \left\{ \sum_{j=1}^{r-1} (a_j - b_j) + a_r \right\}, \tag{2.1}$$

今后, 我们常令 A 表示 $\sum_{r=1}^{n} a_r$, B 表示 $\sum_{r=1}^{n} b_r$, C 表示 $\sum_{i=1}^{n} c_i$ 现在我们将 J_i 和 J_{i+1} 交换位置,即得到一新的排列 $\omega' = \{1, \cdots, i-1, i+1, i, \cdots, n\}$,

	J_1	$\cdots$	J_{i-1}	J_{i+1}	J_i	J_{i+2}	$\cdots$	J_n
M_1	a_1	$\cdots$	a_{i-1}	a_{i+1}	a_i	a_{i+2}	$\cdots$	a_n
M_2	b_1	$\cdots$	b_{i-1}	b_{i+1}	b_i	b_{i+2}	$\cdots$	b_n

同理,对于 ω' 而言,完成全部工作的时间为

$$T(\omega') = B + \max_{\substack{1 \leqslant r \leqslant i-1 \\ i+2 \leqslant r \leqslant n}} \left\{ \sum_{j=1}^{r-1} (a_j - b_j) + a_r, \sum_{j=1}^{i-1} (a_j - b_j) + a_{i+1}, \right.$$
$$\left. \sum_{j=1}^{i-1} (a_j - b_j) + a_i + a_{i+1} - b_{i+1} \right\}. \tag{2.2}$$

由此可知,若

$$\max_{1 \leqslant r \leqslant n} \left\{ \sum_{j=1}^{r-1} (a_j - b_j) + a_r \right\} \leqslant \max_{\substack{1 \leqslant r \leqslant i-1 \\ i+2 \leqslant r \leqslant n}} \left\{ \sum_{j=1}^{r-1} (a_j - b_j) + a_r, \right.$$
$$\left. \sum_{j=1}^{i-1} (a_j - b_j) + a_{i+1}, \sum_{j=1}^{i-1} (a_j - b_j) + a_i + a_{i+1} - b_{i+1} \right\}, \tag{2.3}$$

则排列 ω 比排列 ω' 来得好. 因 (2.1) 式可写成

$$B + \max_{\substack{1 \leqslant r \leqslant i-1 \\ i+2 \leqslant r \leqslant n}} \left\{ \sum_{j=1}^{r-1} (a_j - b_j) + a_r, \sum_{j=1}^{i-1} (a_j - b_j) + a_i, \right.$$
$$\left. \sum_{j=1}^{i-1} (a_j - b_j) + a_i + a_{i+1} - b_i \right\},$$

故若要 (2.3) 式成立,只须

$$\max\left\{\sum_{j=1}^{i-1}(a_j-b_j)+a_i,\ \sum_{j=1}^{i-1}(a_j-b_j)+a_i+a_{i+1}-b_i\right\}$$

$$\leqslant \max\left\{\sum_{j=1}^{i-1}(a_j-b_j)+a_{i+1},\ \sum_{j=1}^{i-1}(a_j-b_j)+a_i+a_{i+1}-b_{i+1}\right\} \tag{2.4}$$

成立即可．将上式两边各加以 $-a_i-a_{i+1}-\sum_{j=1}^{i-1}(a_j-b_j)$，则 (2.4) 可化为

$$\max\{-a_{i+1},\ -b_i\}\leqslant\max\{-a_i,\ -b_{i+1}\},$$

又因对任何实数列 $\{\alpha_1,\ \cdots,\ \alpha_n\}$，有

$$\max\{-\alpha_1,-\alpha_2,\cdots,-\alpha_n\}=-\min\{\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_n\}.$$

故 (2.4) 化为

$$\min\{a_i,\ b_{i+1}\}\leqslant\min\{a_{i+1},\ b_i\}. \tag{2.5}$$

(2.5) 式可解释如下：对于相邻的两个另件 J_i 和 J_{i+1} 来说，若 a_i，b_i，a_{i+1}，b_{i+1} 中最小的数是在上行（即为 a_i 或 a_{i+1}），则与该数相应的另件应排在前面；若最小数是在下行（即为 b_i 或 b_{i+1}），则相应的另件应排在后面．因此，若 $A(\omega)$ 中最小数是在上行，则相应的另件与其他另件相比应排在最前面；同理，若在下行，则相应的另件应排在最后面．排好一个另件后，对余下的另件可依同样规则进行排列．最后所得到的顺序必然是最优的．

§3. 三台机器的情况[1]

对于三台以上机器的排序问题，我们也能给出类似于 (2.5) 的公式来判断两个相邻的另件谁应在先，谁应在后．但这时条件不只一个，所以这判断法则并不经常有效．以后记另件在 M_3 上的加工时间为 c_i.

定理 2. 设 J_i 和 J_j 是某一排列 ω 中相邻的两个另件．若它们的加工时间满足关系

$$\begin{aligned}&\min\{a_i,\ b_j\}\leqslant\min\{b_i,\ a_j\},\\&\min\{b_i,\ c_j\}\leqslant\min\{c_i,\ b_j\},\\&\min\{a_i,\ c_j\}\leqslant\min\{c_i,\ a_j\},\end{aligned} \tag{3.1}$$

则 J_i 应排在 J_j 的前面．

在证明定理之前，我们先证明一引理．

引理． 设给定了一个排列 $\omega=(\omega_1,\ \omega_2,\ \cdots,\ \omega_n)$，

	J_{ω_1}	J_{ω_2}	$\cdots$	J_{ω_n}
M_1	a_{ω_1}	a_{ω_2}	$\cdots$	a_{ω_n}
M_2	b_{ω_1}	b_{ω_2}	$\cdots$	b_{ω_n}
M_3	c_{ω_1}	c_{ω_2}	$\cdots$	c_{ω_n}

令 $A_1=a_{\omega_1}$，$B_1=b_{\omega_1}$；当 $k\geqslant 2$ 时，令

$$\begin{aligned}A_k&=a_{\omega_1}+a_{\omega_2}+\cdots+a_{\omega_k}-b_{\omega_1}-b_{\omega_2}-\cdots-b_{\omega_{k-1}},\\B_k&=b_{\omega_1}+b_{\omega_2}+\cdots+b_{\omega_k}-c_{\omega_1}-c_{\omega_2}-\cdots-c_{\omega_{k-1}},\end{aligned}$$

则

$$T(\omega)=C+\max\{A_k+B_r,\ k\leqslant r,\ k,\ r=1,\ 2,\ \cdots,\ n\}.$$

证． 在定理 1 中，我们已证明 $T(\omega)$ 等于 $A(\omega)$ 的最大可行和．设 l 是连接 $a_{\omega_1},\cdots,a_{\omega_k},\ b_{\omega_k},\ \cdots,\ b_{\omega_r},\ c_{\omega_r},\ \cdots,\ c_{\omega_n}$ 的一可行线，与 l 相应的可行和为

$$
\begin{aligned}
& a_{\omega_1}+\cdots+a_{\omega_k}+b_{\omega_k}+\cdots+b_{\omega_r}+c_{\omega_r}+\cdots+c_{\omega_n} \\
&= c_{\omega_1}+\cdots+c_{\omega_n}+(a_{\omega_1}+\cdots+a_{\omega_k}-b_{\omega_1}-\cdots-b_{\omega_{k-1}}) \\
&\quad +(b_{\omega_1}+\cdots+b_{\omega_r}-c_{\omega_1}-\cdots-c_{\omega_{r-1}})=C+A_k+B_r,
\end{aligned}
$$

因 C 与排列无关,利用定理 1,即可证明此引理.

定理的证明. J_i 和 J_j 在排列 $\omega=(i',\cdots,i'',i,j,j',\cdots,j'')$ 中相邻, J_i 排在第 p 个位置. 令 $\omega'=(i',\cdots,i'',j,i,j',\cdots,j'')$. 我们要证: 若条件 (3.1) 成立, 则有 $T(\omega)\leqslant T(\omega')$. 根据上之引理,只须证明: 在 (3.1) 之下,有

$$
\begin{aligned}
&\max\{A_k+B_r,\ k\leqslant r,\ k,\ r=1,\cdots,n\} \\
&\leqslant \max\{A'_k+B'_r,\ k\leqslant r,\ k,\ r=1,\cdots,n\},
\end{aligned}
$$

其中 A'_k 和 B'_r 分别表示相应于 ω' 的 A_k 和 B_r. 容易看出,当 k 和 $r\neq p$ 和 $p+1$ 时, $A'_k=A_k$, $B'_r=B_r$;同时

$$
\begin{aligned}
A'_p &= A_p+a_j-a_i, \\
A'_{p+1} &= A_{p+1}+b_i-b_j=A_p+a_j-b_j, \\
B'_p &= B_p+b_j-b_i, \\
B'_{p+1} &= B_{p+1}+c_i-c_j=B_p+b_j-c_j.
\end{aligned}
$$

我们令

$$
\begin{aligned}
M &= \max\{A_k+B_r,\ k\leqslant r,\ k\neq p \text{ 和 } p+1,\ r\neq p \text{ 和 } p+1\}, \\
M' &= \max\{A_k,\ k=1,\cdots,p-1\}, \\
M'' &= \max\{B_r,\ r=p+2,\cdots,n\}.
\end{aligned}
$$

显然有

$$
\begin{aligned}
T(\omega) = C+\max\{&M,\ M'+B_p+\max(0,\ b_j-c_i), \\
&M''+A_p+\max(0,\ a_j-b_i), \\
&A_p+B_p+\max(0,\ b_j-c_i,\ a_j+b_j-b_i-c_i)\}, \\
T(\omega') = C+\max\{&M,\ M'+B_p+\max(b_j-b_i,\ b_j-c_j), \\
&A_p+B_p+\max(a_j-a_i+b_j-b_i,\ a_j-a_i+b_j-c_j,\ a_j-c_j), \\
&M''+A_p+\max(a_j-a_i,\ a_j-b_j)\}.
\end{aligned}
$$

比较上面两等式的右边,可知若不等式组

$$
\max\{0,\ b_j-c_i\}\leqslant\max\{b_j-b_i,\ b_j-c_j\}, \tag{3.2}
$$

$$
\max\{0,\ a_j-b_i\}\leqslant\max\{a_j-a_i,\ a_j-b_j\}, \tag{3.3}
$$

$$
\begin{aligned}
&\max\{0,\ b_j-c_i,\ a_j+b_j-b_i-c_i\} \\
&\leqslant\max\{a_j-a_i+b_j-b_i,\ a_j-a_i+b_j-c_j,\ a_j-c_j\}
\end{aligned} \tag{3.4}
$$

成立,则有 $T(\omega)\leqslant T(\omega')$. (3.2) 可改写为

$$
b_j-\min\{b_j,\ c_i\}\leqslant b_j-\min\{b_i,\ c_j\},
$$

此即 (3.1) 中之第二式. (3.3) 可改写为

$$
a_j-\min\{a_j,\ b_i\}\leqslant a_j-\min\{a_i,\ b_j\},
$$

此即 (3.1) 中之第一式. (3.4) 可改写为

$$
\min\{b_i+c_j,\ c_j+a_i,\ a_i+b_i\}\leqslant\min\{a_j+b_j,\ a_j+c_i,\ c_i+b_i\}. \tag{3.5}
$$

我们现分两种情况来证明: 若 (3.1) 成立,则有 (3.5).

(1) 设 $c_j\leqslant a_i$. 由于 (3.1) 之第三式,可得

$$
c_j\leqslant a_i,\ c_i,\ a_j,
$$

因而有 $a_j + b_j \geqslant b_j + c_j$. 另一方面,利用(3.1)之第一式,可知

$$\min\{a_j + c_i, b_i + c_i\} = c_i + \min\{a_j, b_i\} \geqslant c_i + \min\{a_i, b_i\}$$
$$\geqslant c_j + \min\{a_i, b_j\} = \min\{b_j + c_j, c_j + a_i\},$$

故(3.5)右边的每一项皆大于其左边.

(2) 设 $c_j > a_i$. 由于(3.1)之第三式,可得

$$a_i < c_i, c_j, a_j,$$

故得 $b_i + c_i > a_i + b_i$. 另一方面,利用(3.1)之第二式,有

$$\min\{a_i + b_j, a_j + c_i\} = a_i + \min\{b_j, c_i\}$$
$$\geqslant a_i + \min\{b_i, c_j\} = \min\{c_j + a_i, a_i + b_i\},$$

故(3.5)右边之每一项皆大于其左边. 因此,无论在哪种情形,(3.5)皆成立. 这就证明了定理 2.

我们应注意: 第一. 定理 2 的结论只是对相邻的两另件而言才是适用的; 第二. J_i 和 J_j 的加工时间同时满足(3.1)的三个条件的情况一般不是很多的. 所以对某些相邻的两另件,从(3.1)不能判断出谁应在前,谁应在后. 但定理 2 在我们求最优顺序时仍起一定作用;第三. 对于三台以上机器的排序问题,要想从判断相邻的另件的先后出发,来得到最优顺序,这是不可能的,必须探索别的途径来解决这问题(如分支定界法).

§4. 分支定界方法

这种方法的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数方法,也就是把所研究的问题的全部可行解分成若干部分,对每一部分估计出目标函数在其中的一个下界(或上界),从而大致上推测出最优解在哪一部分,然后对这一部分再进行细分和估界,直到找出一个比较好的解答来. 我们还需把这个解对应的函数值与各部分的下界(或上界)进行比较,以判断它是否为最优解. 一般说来,分支定界法的运算量都很大,它只是在没有更好的解法时被采用的一种方法. 下面提出的适用于同顺序的排序问题的分支定界法在估计下界方面改进了[4, 5]中已有的方法.

我们用例子来说明此方法的计算步骤.

例.

	J_1	J_2	J_3	J_4	J_5	J_6
a_i	1	12	5	2	9	11
b_i	8	10	9	6	3	3
c_i	2	4	6	12	7	3

第一步. 根据 $J_1, J_2, \cdots, J_6$ 的每一排列的左边第一个数字,我们将全部排列分成 6 组. 第一个数字是 i 的所有排列归为一组, 用符号 $K_i(i = 1, 2, \cdots, 6)$ 来表示. 然后估计 K_i 中排列的总加工时间 $T(\omega)$ 的下界 r_i. r_1 的算法如下: 把 J_1 排在最左边, 再根据 b_i 和 c_i 的值,将 $J_2, \cdots, J_6$ 按照约翰逊规则加以重排,得到

	J_1	J_6	J_5	J_4	J_3	J_2
	1	11	9	2	5	12
	8	3	3	6	9	10
	2	3	7	12	6	4

计算如下定义的 s_1, s_2 和 s_3:

$$s_1 = A + \min_{i \neq 1}(b_i + c_i) = 46,$$

$$s_2 = a_1 + b_1 + \begin{bmatrix} 3 & 3 & 6 & 9 & 10 \\ 3 & 7 & 12 & 6 & 4 \end{bmatrix} \text{的最大可行和} = 44,$$

$$s_3 = a_1 + b_1 + c_1 + \sum_{i \neq 1} c_i = 43.$$

由定理 1 可知，$\max(s_1, s_2, s_3) \leqslant \min_{\omega \in K_1} T(\omega)$，故取 $r_1 = \max(s_1, s_2, s_3) = 46$.

一般说来，如 $D = (\omega_1, \cdots, \omega_j)$ 是 j 个另件 ($j \leqslant n$) 排成的一顺序，K_D 是 $J_1, \cdots, J_n$ 排成的顺序中所有 $(\omega_1, \cdots, \omega_j, \cdots)$ 形的顺序所组成的集合，则我们定义总加工时间 $T(\omega)$ 在 K_D 中之下界 r_D 如下：根据 b 和 c 的值，将 D 以外的另件按约翰逊规则重排为 $(\omega_{j+1}, \cdots, \omega_n)$，令

$$s_1 = A + \min_{i \in (\omega_{j+1}, \cdots, \omega_n)} (b_i + c_i),$$

$$s_2 = \text{矩阵} \begin{bmatrix} a_{\omega_1}, & \cdots, & a_{\omega_j} \\ b_{\omega_1}, & \cdots, & b_{\omega_j} \end{bmatrix} \text{的最大可行和}$$

$$+ \text{矩阵} \begin{bmatrix} b_{\omega_{j+1}}, & \cdots, & b_{\omega_n} \\ c_{\omega_{j+1}}, & \cdots, & c_{\omega_n} \end{bmatrix} \text{的最大可行和},$$

$$s_3 = \text{矩阵} \begin{bmatrix} a_{\omega_1}, & \cdots, & a_{\omega_j} \\ b_{\omega_1}, & \cdots, & b_{\omega_j} \\ c_{\omega_1}, & \cdots, & c_{\omega_j} \end{bmatrix} \text{的最大可行和} + \sum_{i=j+1}^{n} c_{\omega_i},$$

则定义 $r_D = \max(s_1, s_2, s_3)$.

利用上述定义，我们计算出 $r_2 = 56$, $r_3 = 48$, $r_4 = 46$, $r_5 = 48$, $r_6 = 52$.

第二步. 选取 $r_1, \cdots, r_6$ 中最小者（若最小数不止一个，可任选其一），今取 r_4. 我们对 K_4 再进行分组. 根据 K_4 中每一排列的左边第二个数字，将 K_4 分为 5 组，分别表为 K_{41}, K_{42}, K_{43}, K_{45} 和 K_{46}. 然后计算出总加工时间在各组的下界 r_{4i}. 利用上述定义，我们有 $r_{41} = 46$, $r_{42} = 49$, $r_{43} = 46$, $r_{45} = 46$, $r_{46} = 50$.

第三步. 选取 $r_{41}, \cdots, r_{46}$ 中最小者，今取 r_{41}. 继续对 K_{41} 进行分组. 根据 K_{41} 中每一排列的左边第三个数字，将 K_{41} 分成 K_{412}, K_{413}, K_{415}, K_{416}. 再计算下界 $r_{412} = 47$, $r_{413} = 46$, $r_{415} = 46$, $r_{416} = 50$.

第四步. 选取 $r_{412}, \cdots, r_{416}$ 中最小数，今取 r_{415}. 把 K_{415} 分为 K_{4152}, K_{4153}, K_{4156}. 由于 K_{4152} 只含两个排列，即 (415236) 和 (415263)，我们可直接计算出它们的总加工时间 $T(415236) = 52$, $T(415263) = 55$. 对 K_{4153}, K_{4156} 亦同样处理. 这 6 个排列的总加工时间中最小者为 $T(415326) = 46$.

第五步. 将以上分组情况划成一"树形图"(图 3)，并注明各组的下界. 显然下界不小于 $T(415326)$ 的那些组内不可能含有比 (415326) 更好的排列，因此这些组即可被消去. 由于图中所有的下界均不小于 46，故 (415326) 就是最优顺序. 在一般情况下，如果不能消去所有的组，则需进入下一步.

第六步. 在未被消去的那些组中选出下界最小的一个组，重复上述的计算步骤. 由

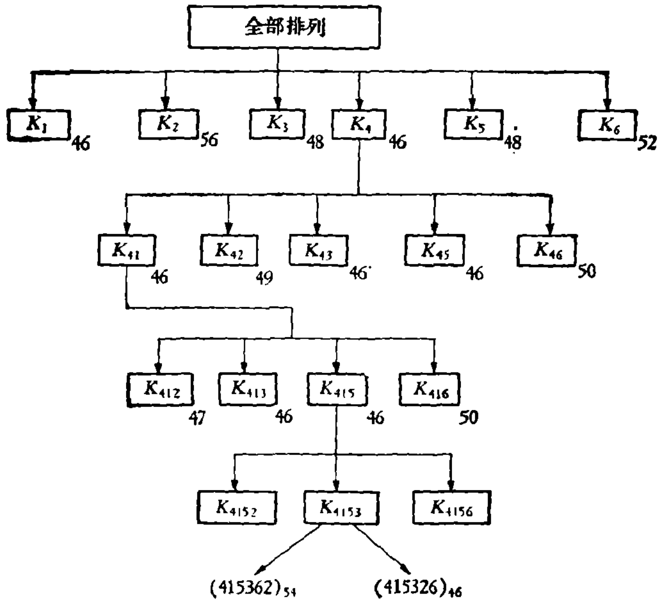

图 3

于全部排列是有限个,所以这种分支定界法一定能得到最优顺序.

我们把上述计算步骤划成框图形式(图 4),以便于使用电子计算机来解排序问题.框图中的符号所代表的意思如下:

S: 全体另件 $J_1, \cdots, J_n$;

z^*: 最优顺序的总加工时间的上界,即

$$z^* \geqslant \min_{\omega} T(\omega);$$

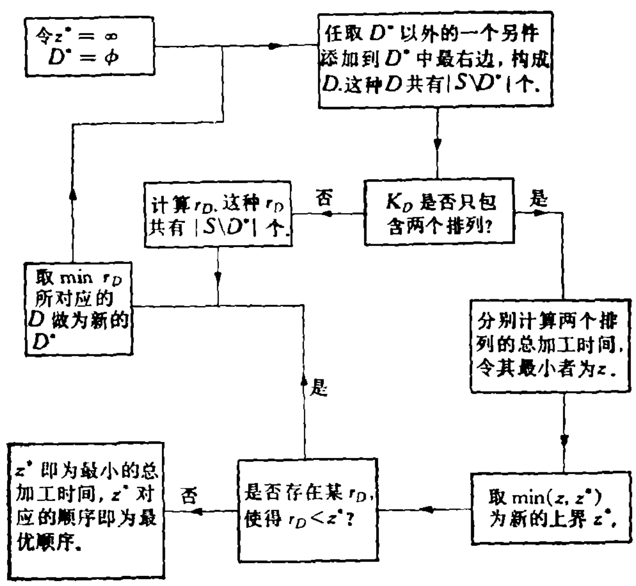

图 4

K_D：如 $D=(\omega_1, \cdots, \omega_j)$，则 K_D 是 $J_1, \cdots, J_n$ 排成的顺序中所有 $(\omega_1, \cdots, \omega_j, \cdots)$ 形的顺序所组成的集合；

r_D：总加工时间 $T(\omega)$ 在 K_D 中之下界，

$|S\backslash D^*|$：D^* 表示一部分另件的一个排列，$|S\backslash D^*|$ 表示 D^* 以外的另件的个数（ϕ 表示空集）.

§5. 几种特殊情形

对于三台机器的情况，一般说来寻求最优顺序是比较麻烦的．但对于几种特殊情况，我们却容易算出其最优顺序．现仅举两种情况：

(5.1) 对所有的 i，皆有 $b_i \geqslant \min(\max\limits_j a_j, \max\limits_j c_j)$.

由定理 1 可知：任一排序问题的最优顺序 $(\omega_1^*, \omega_2^*, \cdots, \omega_n^*)$ 的逆顺序 $(\omega_n^*, \omega_{n-1}^*, \cdots, \omega_1^*)$ 也是逆问题

$$\begin{bmatrix} c_n & c_{n-1} & \cdots & c_1 \\ b_n & b_{n-1} & \cdots & b_1 \\ a_n & a_{n-1} & \cdots & a_1 \end{bmatrix}$$

的最优顺序．故对 5.1 的情况，我们只须考虑

$$b_i \geqslant \max_j a_j, \quad i = 1, 2, \cdots, n \tag{5.1}$$

即可．利用定理 2 中的引理，对于任意排列 $(\omega_1, \omega_2, \cdots, \omega_n)$，在 (5.1) 的情况下，因

$$A_k = a_{\omega_k} + \sum_{j=1}^{k-1}(a_{\omega_j} - b_{\omega_j}) \leqslant a_{\omega_{k-1}} + \sum_{j=1}^{k-2}(a_{\omega_j} - b_{\omega_j}) = A_{k-1},$$

故 $T(\omega) = C + \max\limits_r (A_1 + B_r) = C + A_1 + \max\limits_r B_r$. 这样，在 ω_1 已经固定时，要想求出最小的 $\max B_r$，我们只须对

$$\begin{bmatrix} b_{\omega_2}, & \cdots, & b_{\omega_n} \\ c_{\omega_2}, & \cdots, & c_{\omega_n} \end{bmatrix}$$

按约翰逊规则加以排列即可．现在我们假设

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \\ c_1 & c_2 & \cdots & c_n \end{bmatrix}$$

中 b 行和 c 行已经按约翰逊规则重新排列过．欲得更好的顺序，我们只能将 J_1 换成别的 J_i，而且除把 J_i 排在最前面外，其余的另件仍保持原有的先后次序．也就是说，我们只须比较以下 n 个排列：$(1, 2, 3, \cdots, n)$，$(2, 1, 3, \cdots, n)$，$(3, 1, 2, \cdots, n)$，$\cdots$，$(n, 1, 2, \cdots, n-1)$，就可得出最优顺序．

(5.2) 对于所有的 i，皆有 $b_i \leqslant \max(\min\limits_k a_k, \min\limits_k c_k)$.

此问题的最优顺序可以加工时间为

$$\begin{bmatrix} a_1 + b_1, & \cdots, & a_n + b_n \\ b_1 + c_1, & \cdots, & b_n + c_n \end{bmatrix} \tag{5.2}$$

的两台机器的排序问题的最优顺序得到．

我们只考虑 $b_i \leqslant \min\limits_k a_k$，$i = 1, \cdots, n$ ($b_i \leqslant \min\limits_k c_k$ 的情况可同样证明)．此时有

$$A_k = a_{\omega_k} + \sum_{i=1}^{k-1}(a_{\omega_i} - b_{\omega_i}) \leqslant a_{\omega_{k+}} + \sum_{i=1}^{k}(a_{\omega_i} - b_{\omega_i}) = A_{k+1}.$$

故对于任意排列 $(\omega_1, \omega_2, \cdots, \omega_n)$ 来说，

$$T(\omega) = C + \max_r(A_r + B_r).$$

但因

$$A_r + B_r = \sum_{j=1}^{r}(a_{\omega_j} + b_{\omega_j}) - \sum_{j=1}^{r-1}(b_{\omega_j} - c_{\omega_j}),$$

利用公式(2.1)，即可证明(5.2)的两台机器的最优顺序也就是原问题的最优顺序.

四、关于不同顺序的多台机器的排序问题

许多实际的排序问题并不具备同顺序的性质，往往是每个另件各以其特定的次序通过若干台机器. 在这种不同顺序的排序问题中，有的另件可能只需被一部分机器加工，不必通过所有的机器；有的另件被某台机器加工后，再通过其他的机器，而后又到这台机器上再次加工；有的机器在加工完另件 i 后，需要进行一番调整才能再加工另件 j，这段调整时间的长短与 i, j 有关，等等. 应该说，不同顺序的排序问题更能反映实际出现的排序问题的多样性，因此需要我们加以研究. 这种排序问题不仅在工厂内发生，在交通运输、邮电通讯、电子仪器设计等许多方面也会发生. 例如熟知的“邮路问题”就是一种具有调整时间的单台机器的排序问题.

我们在本章内仍只考虑完成全部加工工作的总加工时间这个目标函数. 为了使总加工时间达到最短，我们必须对每一台机器都求出加工另件的最优顺序. 在不同的机器上这种最优顺序可能不同. 显然，不同顺序的排序问题所考虑的可行解比同顺序的排序问题要广泛得多. 由于不同顺序的排序问题所涉及的因素的复杂性，以及可行解数量的激剧增多，就使得它的最优解的寻求比同顺序的排序问题更为困难. 目前解决不同顺序的排序问题的最好方法是把它化成一个“图论”问题，然后用分支定界法来求解. 不过这种方法的计算过程仍是很繁杂的.

§1. 图和选出集

设有若干个点构成的集合 X. 若在 X 内点 i 和 j 之间存在着先后关系，这种关系用符号 (i, j) 表示，称为一(有向)“弧”. 我们也用带箭头的线段来表示弧，$i \to j$，i 是弧的起点，j 是弧的终点. 点集 X 和全体弧 Z 就构成一个(有向)“图”，用符号 (X, Z) 表示之. 在现实生活中，如以点表示城市，弧表示两城市间的交通线，则图就可表示若干城市之间的交通网络；如以点表示施工中一个工序，弧表示两工序的先后关系，则图就可表示施工的“统筹图”. 我们在每一弧 (i, j) 上对应一个数 d_{ij}. d_{ij} 可表示交通网络中点 i 到点 j 的距离；也可表示施工统筹图中工序 i 的开工时间到工序 j 的开工时间这段时间之长. 图 (X, Z) 中若干个首尾相接的弧 $\{(i_1, i_2), (i_2, i_3), (i_3, i_4), \cdots, (i_{k-1}, i_k)\}$ 就构成一条“路”；如果 $i_k = i_1$，这样的路称为一个“圈”. 把一条路上相应的 d_{ij} 加起来，我们得到这条路之“长”. 图 (X, Z) 中最长的路称为“关键路”. 例如在统筹图中关键路就是主要矛盾线. 我们可把不同顺序的排序问题化为图的问题. 为此，我们约定：

(1) 一个另件被一台机器加工一次，称为一个“工序”. 每个工序用图中一个点代表.

此外再增加图的起点0和终点 e. 全部点用 X 表示;

(2) 一另件在通过工序 i 后, 下一个工序若是 j, 即以弧 (i, j) 代表这种关系. 此外从起点. 0连一弧到每一另件的第一个工序, 再从每一另件的最后工序连一弧到终点 e. 这三种弧的全体用 Z 表示;

(3) 若干个另件要在同一台机器上加工, 这些工序的先后顺序就是排序问题所要解决的. 我们在这些工序的每两个工序 i, j 之间, 都连上一对弧 (i, j) 和 (j, i). 所谓寻求这些工序的最优顺序, 也就是要在每一对弧中选出一个弧. 这种成对的弧的全体用 D 表示. $(X, Z\cup D)$ 即构成一个图;

(4) 每一弧 $(i, j)\in Z\cup D$ 都对应一个非负数 d_{ij}, d_{ij} 是工序 i 的加工时间与工序 j 开工前的调整准备时间之和. 此外令 $d_{0i}=0$.

例. 现有两台机器 M_1 和 M_2, 两个另件 J_1 和 J_2. J_1 先在 M_1 上加工(工序①), 再到 M_2 上加工(工序②), 再到 M_1 上加工(工序③); J_2 先在 M_2 上加工(工序④), 再到 M_1 上加工(工序⑤)、再到 M_2 上加工(工序⑥). 各工序的加工时间如下表(不考虑调整时间):

工序	①	②	③	④	⑤	⑥	
工时	3	7	5	1	4	6	(小时)

这个例子对应的图 $(X, Z\cup D)$ 如图5所示.

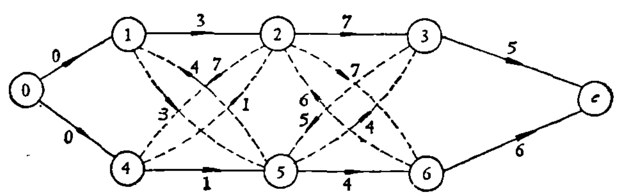

图 5

全体实线弧构成 Z, 全体虚线弧构成 D. 弧 (i, j) 上标的数是工序 i 的加工时间. 若我们把 M_1 加工的工序①, ③和⑤任意地排个顺序(当然这个顺序必须附合另件的加工流程的规定), 比如⑤在①, ③之后; 同时把 M_2 加工的工序②, ④和⑥排成②在④之后, 在⑥之前. 这相当于在每对虚线弧中选出一弧, 它们是 $S=\{(1, 5), (3, 5), (4, 2), (2, 6)\}$. 显然图 $(X, Z\cup S)$ 是无圈的. 利用递推公式

$$\begin{cases} t_0=0, \\ t_j=\max\{t_i+d_{ij}:\ 使\ (i, j)\in Z\cup S\ 的所有\ i\},\ j\in X, \end{cases} \tag{4.1}$$

我们容易算出每个工序 i 的开工时间 t_i: $t_1=0$, $t_2=3$, $t_3=10$, $t_4=0$, $t_5=15$, $t_6=19$, $t_e=25$. 可以证明: 完成全部加工工作的总加工时间也就是 $t_e=25$ 小时(即图 $(X, Z\cup S)$ 的关键路之长). 反之, 若在 D 的每对虚线弧中任意选出一弧, 这些弧构成 S, 只要图 $(X, Z\cup S)$ 内无圈, 则 M_1 和 M_2 上另件的加工顺序就由 S 给定. 以后我们称使 $(X, Z\cup S)$ 内无圈的 S 为 D 的一个"选出集". 选出集和另件加工顺序之间的这种对应关系对于一般的 $m\times n$ 排序问题也成立. 如果 D 内共有 p 对虚线弧, 那么选出集 S 的个数不超过 2^p 个. 最优加工顺序所对应的选出集 S^* 必然使图 $(X, Z\cup S^*)$ 的关键路为最短. 这样, 寻

求最优顺序问题就化为寻求最优选出集 S^* 的问题.

§2. 图 $(X, Z\cup D)$ 的基本性质

下面的几个定理是计算最优选出集 S^* 的方法的理论基础.

定理 3. 如果 i 是 X 中起点 0 和终点 e 以外的任意一点，则图 (X, Z) 中必有一条从 0 连到 i 的路,和一条从 i 连到 e 的条;且图 (X, Z) 是无圈的.

定理 4. 如果给定了一个选出集 S,则图 $(X, Z\cup S)$ 中各工序的开工时间 $\{t_i, i\in X\}$ 满足条件

$$\begin{cases} t_j - t_i \geqslant d_{ij}, \text{如} (i, j)\in Z\cup S, \\ t_i \geqslant 0, \qquad i\in X, \\ t_e = \min^{1)}. \end{cases} \tag{4.2}$$

证. 由于图 $(X, Z\cup S)$ 是无圈的,故利用 (4.1) 可算出开工时间 $(t_i, i\in X)$,其中 t_e 也就是 $(X, Z\cup S)$ 的关键路之长. 所以 (4.2) 中诸条件必然被满足.

设机器的台数为 m. 我们现将 X 内除起点 0 和终点 e 以外的所有点分为 m 组，分别表为 $X_1, X_2, \cdots, X_m$,其中 X_k 表示第 k 台机器加工的所有工序. 又设 D' 是由若干对虚线弧构成的集合. 在 D' 的每一对虚线弧中任意选出一弧，这些弧构成集合 S'. 如果图 $(X, Z\cup S')$ 是无圈的,我们可利用递推公式:

$$\begin{cases} u_0 = 0, \\ u_j = \max\{u_i + d_{ij}: \text{使} (i, j)\in Z\cup S' \text{的所有} i\}, \quad j\in X, \end{cases} \tag{4.3}$$

算出一组 u_i, $i\in X$. 这组 u_i 满足下面的

定理 5. (i) 若 $\{u_i, i\in X\}$ 满足方程

$$\begin{cases} u_j - u_i \geqslant d_{ij}, \text{当} (i, j)\in Z, \\ u_j - u_i \geqslant d_{ij}, \text{或} u_i - u_j \geqslant d_{ji}, \text{当} i, j\in X_k, k = 1, \cdots, m, \\ u_i \geqslant 0, \qquad i\in X, \end{cases} \tag{4.4}$$

则在 D 中存在一选出集 S,使得 $S\supseteq S'$,且 u_e 等于图 $(X, Z\cup S)$ 的关键路之长;

(ii) 若 $\{u_i, i\in X\}$ 不满足 (4.4),也就是存在 i_0 和 j_0, 它们同属于某个 X_k, 使得不等式

$$\begin{cases} u_{j_0} - u_{i_0} < d_{i_0 j_0}, \\ u_{i_0} - u_{j_0} < d_{j_0 i_0}, \end{cases} \tag{4.5}$$

成立,则总可在虚线弧对 $\{(i_0, j_0), (j_0, i_0)\}$ 中选出一弧，将此弧并入 $Z\cup S'$ 后,所得的图仍是无圈的.

证. (i) 当 $\{(i, j), (j, i)\}$ 不属于 D' 时,因不等式 $u_j - u_i \geqslant d_{ij}$ 和 $u_i - u_j \geqslant d_{ji}$ 中必有一式成立,如前式成立,则我们将 (i, j) 并入 S' 内;反之,将 (j, i) 并入 S' 内. 对 D' 以外的每一对虚线弧都如此处理，我们即得一选出集 S. 由于 u_i 满足 (4.4)，所以利用 (4.1) 算出的 $(X, Z\cup S)$ 中各工序的开工时间 t_i 必等于 u_i.

---

1) $t_e = \min$ 的意思是：如一组别的数 $(u_i, i\in X)$ 也满足不等式

$$\begin{cases} u_j - u_i \geqslant d_{ij}, & (i, j)\in Z\cup S, \\ u_i \geqslant 0, & i\in X, \end{cases}$$

则必有 $t_e \leqslant u_e$.

(ii) 假设 $G_1=(X, Z\cup S'\cup\{(i_0, j_0)\})$ 和 $G_2=(X, Z\cup S'\cup\{(j_0, i_0)\})$ 都是有圈的，因 $(X, Z\cup S')$ 是无圈的，故 G_1 和 G_2 的圈必分别通过 (i_0, j_0) 和 (j_0, i_0). 即 G_1 的圈可表为 $\{(i_0, j_0), (j_0, h), \cdots, (k, i_0)\}$, G_2 的圈可表为 $\{(j_0, i_0), (i_0, f), \cdots, (g, j_0)\}$,所以 $\{(j_0, h), \cdots, (k, i_0), (i_0, f), \cdots, (g, j_0)\}$ 就构成 $(X, Z\cup S')$ 中的一个圈. 这与 $(X, Z\cup S')$ 无圈相矛盾,故 G_1 和 G_2 总有一个是无圈的.

§3. 分支定界法[6,7]

我们先叙述此方法的计算步骤:

首先令 $z^*=\infty$, $S'=\phi$,以及令 $q=0$.

z^* 是最优顺序对应的总加工时间 t_e^* 的上界, q 是 S' 中弧的个数.

第一步. 计算图 $G=(X, Z\cup S')$ 的关键路之长, 也就是利用 (4.3) 算出 u_i, $i\in X$. 如 $u_e\geqslant z^*$,则转入第四步;如 $u_e<z^*$,则取 $z_q=u_e$,并转入第二步.

z_q 是包含 S' 的所有选出集对应的总加工时间的下界. 当 $S'=\phi$ 时, 算出的 u_e 显然不大于最优顺序对应的总加工时间 t_e^*.

第二步. 如 $\{u_i, i\in X\}$ 满足 (4.4), 则令 $z^*=z_q$, $\bar{S}=S'$, 再转入第四步; 如 $\{u_i, i\in X\}$ 不满足 (4.4), 则根据定理 5(ii), 可选出虚线弧 (i_q, j_q), i_q, j_q 同属于某 X_k, 再转入第三步.

第三步. 将弧 (i_q, j_q) 并入 S', 取所得之集为新的 S', 再取 $q+1$ 为新的 q. 令 $p_q=1$, 然后转回第一步.

符号 "$p_q=1$" 表示将 (i_q, j_q) 并入 S', "$p_q=-1$" 表示将 (j_q, i_q) 并入 S'.

第四步. 如 $p_q=1$,则转入第五步;如 $p_q=-1$,则转入第六步.

第五步. 从 S' 中除去弧 (i_{q-1}, j_{q-1}),取所得之集为新的 S'. 如 $z_{q-1}=z^*$,则取 $q-1$ 为新的 q,再转回第四步;如 $z_{q-1}<z^*$,则转入第八步.

第六步. 取 $q-1$ 为新的 q. 如 $q=0$,则计算过程到此终止. 根据定理 5(i),最优选出集 S^* 可由 $\bar{S}$ 扩大而得. 如 $q>0$, 则从 S' 中除去弧 (j_q, i_q), 取所得之集为新的 S', 再转入第七步.

第七步. 如 $p_q=-1$, 则转回第六步;如 $p_q=1$, 则从 S' 中除去弧 (i_{q-1}, j_{q-1}), 取所得之集为新的 S',再转入第八步.

第八步. 将弧 (j_{q-1}, i_{q-1}) 并入 S', 取所得之集为新的 S'. 令 $p_q=-1$, 再转回第一步.

我们具体计算本章中的例子, 以说明以上步骤的用法. 显然本例的 $Z=\{(0, 1), (0, 4), (1,2), (2,3), (4,5), (5,6), (3,e), (6, e)\}$, $X_1=\{1, 3, 5\}$, $X_2=\{2, 4, 6\}$. 先令 $z^*=\infty$, $S'=\phi$, $q=0$.

第一步. $u_1=0$, $u_2=3$, $u_3=10$, $u_4=0$, $u_5=1$, $u_6=5$, $u_e=15$. 取 $z_0=15$, 转入第二步.

第二步. 因 $u_5-u_1=1<3=d_{15}$, $u_1-u_5=-1<4=d_{51}$,故可选出弧 $(1, 5)$,转入第三步.

第三步. 取 $S'=\{(1, 5)\}$,令 $q=1$, $p_1=1$. 转回第一步. ……

为节省篇幅,我们将每一步的计算结果列成下表: (见下页表)

步数	q	并入弧	消去弧	u_q	z_q	i_q	j_q	p_q	z^*
1	0			15	15				
2						1	5		
3	1	(1,5)						1	
1				15	15				
2						2	6		
3	2	(2,6)						1	
1				16	16				
2									16
5			(2,6)						
8		(6,2)						−1	
1				25					
6	1		(6,2)						
7			(1,5)						
8		(5,1)						−1	
1				18					
6	0								

最后得到 $\bar{S}=\{(1,5),(2,6)\}$，所以最优选出集 $S^*=\{(1,5),(2,6),(4,2),(5,3)\}$，最短的总加工时间是 16. 为了便于理解计算步骤，我们再划出分支的情况：

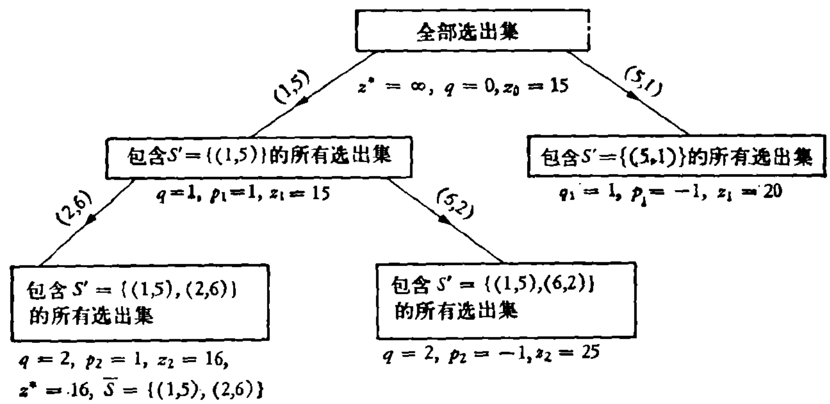

图 6

全部计算步骤的框图见后页：

参 考 文 献

[1] 越民义 韩继业，n 个零件在 m 台机床上的加工顺序问题 (I)，中国科学，5(1975)，462—470.

[2] Johnson S. M., Optimal Two-and-Three Stage Production Schedules with Setup Times Included, *Naval Res. Logis. Quart.* **1**:1 (1954), 61—68.

[3] Szwarc W., Mathematimal Aspects of the 3×n Job-Shop Sequencing Problem, *Naval Res. Logis Quart.*, **21**:1 (1974), 145—153.

[4] Lomnicki Z. A., A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem, *Opera. Res. Quart.*, **16**:1 (1965), 89—100.

[5] Ignall E., Schrage L., Application of the Branch-and-Bound Techniques to Some Flow Shop Scheduling Problems, *Opns. Res.*, **13**:3 (1965), 400—412.

[6] Balas E., Machine Sequencing Via Disjunctive Graphs:An Implicit Enumeration Algorithm, *Opns. Res.*, **17**:6 (1969), 941—957,

[7] Charlton J. M., Death C. C., A Generalized Machine Scheduling Algorithm, *Opera. Res. Quart.*, **21**:1 (1970), 127—134.

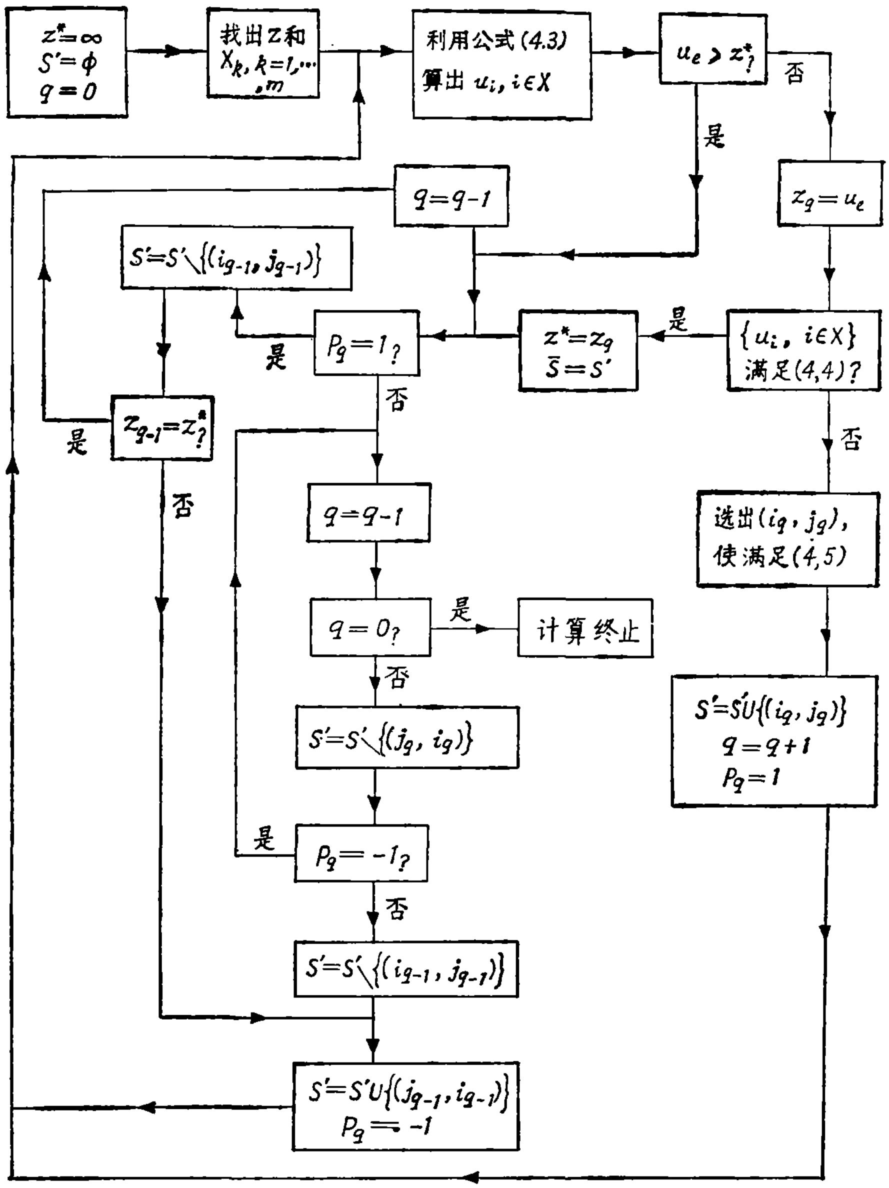
$z^*=\infty$
$S'=\phi$
$q=0$
找出Z和
$X_k, k=1,\cdots,m$
利用公式(4.3)
算出$u_i, i\in X$
$u_e>z^*$?
否
是
$z_q=u_e$
$q=q-1$
$S'=S'\setminus\{(i_{q-1}, j_{q-1})\}$
$P_q=1$?
是
$z^*=z_q$
$\bar{S}=S'$
是
$\{u_i, i\in X\}$
满足(4,4)?
否
$z_{q-1}=z^*$?
是
否
否
选出(i_q, j_q),
使满足(4,5)
$q=q-1$
$q=0$?
是
计算终止
否
$S'=S'\setminus\{(j_q, i_q)\}$
$S'=S'\cup\{(i_q, j_q)\}$
$q=q+1$
$P_q=1$
是
$P_q=-1$?
否
$S'=S'\setminus\{(i_{q-1}, j_{q-1})\}$
$S'=S'\cup\{(j_{q-1}, i_{q-1})\}$
$P_q=-1$

图 7