文章编号：0255 - 8297(2004)02 - 0247 - 05

两台机器流水作业中带成组加工的最大迟后问题

陈 跃， 孙世杰， 宋政芳， 何龙敏
（上海大学 数学系，上海 200436）

摘 要：考虑分批加工中的流水作业问题：且工件在两台机器间作成批转移，目标函数为 L_{max}. 文中指出该问题为 NP-hard 后给出了其多项式可解的特例并构造了相应的动态规划算法.

关键词：排序；批处理机；最大迟后；强 NP-hard；多项式可解
中图分类号：O223 **文献标识码**：A

Minimizing the Maximum Lateness on a Two Machine Flowshop with Batch Processors

CHEN Yue， SUN Shi-jie， SONG Zheng-fang， HE Long-min
(*Department of Mathematics, Shanghai University, Shanghai* 200436, *China*)

Abstract: This paper considers the following problem: n jobs need to be processed on two machines (M_1, M_2) successively. The deadline for job j is d_j, and the processing times of job j on M_1, M_2 are a_j, b_j, respectively. Both machines are batch processors. This means n jobs are grouped into several batches on M_i, $i=1,2$, respectively, and the machines process the jobs in same batch simultaneously. The processing time of a batch is equal to the longest processing time of all the jobs in this batch. Thus all the jobs in same batch are processed in the same length of time on the given machine, and the jobs will be also shifted in batches. We take the maximum lateness as our objec function for minimization. After pointing out that this scheduling problem is NP-hard, we give some special cases that can be solved in polynomial time and construct the corresponding dynamic programming.

Key words: scheduling; batch processor; maximum lateness; strong NP-hard; polynomial time algorithm

本文考虑下述存在分批加工的流水作业问题：n 个工件需要依次在机器 M_1, M_2 上加工. 工件 j 的应交工时间为 d_j，其在机器 M_1, M_2 上的加工时间分别为 a_j, b_j. 机器 $M_i, i=1,2$ 依成批方式加工，即 n 个工件在机器 M_i 上分成若干批，机器 M_i 同时加工同批工件，其所需的加工时间为该批工件中所需加工时间的最大值. 因此同批工件在机器 M_i 上的加工时间相同，而工件也就作成批转移. 不同机器的分批方式可以不同. 用三参数表示法[1]将此类问题记为：$F2(B_1,B_2)|\mathrm{BI}|f$，其中 B_1, B_2 分别是 M_1, M_2 上的每批工件数的上界，若无上界则取 ∞，BI 指 Burn In，因为此种加工方式常出现在半导体元件老化测试中，故用 BI 表示[2]，f 为目标函数.

对上述问题，Ahmadi[3]对 $F2(B_1,1)|a_j=a,\mathrm{BI}|C_{max}$，$F2(1,B_2)|b_j=b,\mathrm{BI}|C_{max}$，$F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|C_{max}$，分别给出了 $O(n\log n)$，$O(n\log n)$，

收稿日期：2003-03-19； **修订日期**：2003-09-01
作者简介：陈跃(1978-)，男，浙江杭州人，硕士生；孙世杰(1945-)，男，上海人，教授，博导.

$O(n)$算法；对 $F2(1,B_2)|b_j=b,\mathrm{BI}|\sum C_j$，$F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|\sum C_j$，分别给出了 $O(n^3)$算法，并且证得 $F2(B_1,1)|a_j=a,\mathrm{BI}|\sum C_j$ 为强 NP-hard 问题. Potts[4]证得：(1) $F2(\infty,\infty)|\mathrm{BI}|C_{\max}$，$O(n\log n)$可解，所获最优解中 n 个工件最多分成两批；(2) 如果 $\max\{B_1,B_2\}\geqslant 2$，则 $F2(B_1,B_2)|\mathrm{BI}|C_{\max}$为普通意义下的 NP-hard 问题，即存在伪多项式时间算法；(3) $F2(1,\infty)|\mathrm{BI}|C_{\max}$可以转化为 $1|\mathrm{BI}|L_{\max}$，因此利用 Brucker[5]的算法可以 $O(n^2)$求解. Hoogereen 和 Van de Velde[6]进一步证得即使 $B_1\geqslant 1$，$F2(B_1,1)|a_j=a,\mathrm{BI}|\sum C_j$ 已经为强 NP-hard 的，并给出了两个性能比为 3/2 的 $O(n\log n)$的近似算法. 对 Common due date 下的 JIT 问题，Sung 和 Min[7]对 $F2(1,B_2)|b_j=b,\mathrm{BI}|\sum|C_{2j}-d|$，$F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|\sum|C_{2j}-d|$分别给出了多项式时间算法，并在证得 $F2(B_1,1)|a_j=a,\mathrm{BI}|\sum|C_{2j}-d|$的 NP-hard 性以后给出了伪多项式时间算法.

可以看出，上述对批处理机的研究大部分局限于加工全程 $C_{\max}$和完工时间之和 $\sum C_j$ 这两个目标函数，对最大迟后问题 $L_{\max}$的研究也只是研究了一台机器的情况. 本文研究两台批处理机的最大迟后问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$，在指出该问题为 NP-hard 后给出了其多项式可解的特例并构造了相应的动态规划算法.

1 问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$为 NP-hard 性

定理 1 一般情况下的问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$为 NP-hard 的

证明 Potts et al[4]证得：如果 $\max\{B_1,B_2\}\geqslant 2$，则 $F2(B_1,B_2)|\mathrm{BI}|C_{\max}$为 NP-hard 的，而当取所有的 $d_j=0$ 时，问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$退化为 $F2(B_1,B_2)|\mathrm{BI}|C_{\max}$，因此由后者的 NP-hard 性即得一般情况下的问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$为 NP-hard 的.

下面给出问题 $F2(B_1,B_2)|\mathrm{BI}|L_{\max}$多项式可解的特例，并构造相应的算法.

2 一致性条件下问题 $F2(1,B_2)|b_j=b,\mathrm{BI}|L_{\max}$的求解

此时第一台机器为单处理机而第二台机器为批处理机. 已经知道对于问题 $F2(1,B_2)|b_j=b,\mathrm{BI}|f$，当 $f\in\{C_{\max},\sum_{j\in J}C_j\}$时，SPT 序是最优的，但是对于 $F2(1,B_2)|b_j=b,\mathrm{BI}|L_{\max}$这个问题相应的 EDD 序却不一定是最优的.

例 对 $n=4$；$B_2=2$；$a_1=1,a_2=9,a_3=a_4=1$；$d_1=d_2=12,d_3=13,d_4=14$；$b_1=b_2=b_3=b_4=10$.

对应的 EDD 序(1,2,3,4)，其 $L_{\max}=17$，而对应非 EDD 序(1,3,2,4)，其 $L_{\max}=10$，此例即说明 EDD 序不一定比非 EDD 序好. 但在一致性条件下问题 $F2(1,B_2)|b_j=b,\mathrm{BI}|L_{\max}$可在多项式时间内获解.

称工件的应交工时间和加工时间之间具有一致性，即若工件的加工时间有 $a_1\leqslant a_2\leqslant\cdots\leqslant a_n$，则有 $d_1\leqslant d_2\leqslant\cdots\leqslant d_n$.

引理 1 如果 $a_i\leqslant a_j$ 隐含着 $d_i\leqslant d_j$，$i\neq j$，则存在这样的最优解，在机器 1 上工件 i 先于工件 j 加工.

证明 假设有一最优解 π^*，在其中存在两工件 i、j，$d_i<d_j$(结合一致性条件知 $a_i\leqslant a_j$)，但在机器 1 上工件 j 先于工件 i 加工；且不妨假设在解 π^* 中，机器 2 上包含工件 i,j 的批分别为 P^*,Q^*(可相同).

现在机器 1 上将工件 i 和工件 j 的位置交换，其余不动；在机器 2 上令 $P=(Q^*-\{j\})\cup\{i\}$，$Q=(P^*-\{i\})\cup\{j\}$，其余批不动，依 $a_i\leqslant a_j$ 知这样的分批是可行的. 将所得解记为 π.

容易证明解 π 不比解 π^* 差，故以下省略，引理得证.

定义 1 批-EDD 序：指序中的两批工件 P 和 Q，如 P 先加工，Q 后加工，则不存在这样的工件对 i 和 j，使得 $i\in P$，$j\in Q$，并且 $d_i>d_j$.

引理 2 在一致性条件下，存在这样的最优解，在机器 2 上各工件依批-EDD 序加工.

证明 不妨设存在一最优解，在机器 2 上各工件不依批-EDD 序加工，即存在某两批 P，Q 及某两个工件，$i\in P$，$j\in Q$，机器 2 先加工 P，后加工 Q，但是 $d_i>d_j$. 即加工状况呈下述图中的 a,b 形状之一.

相应 a 由 $d_i>d_j$ 及一致性条件知 $a_i\geqslant a_j$，在机

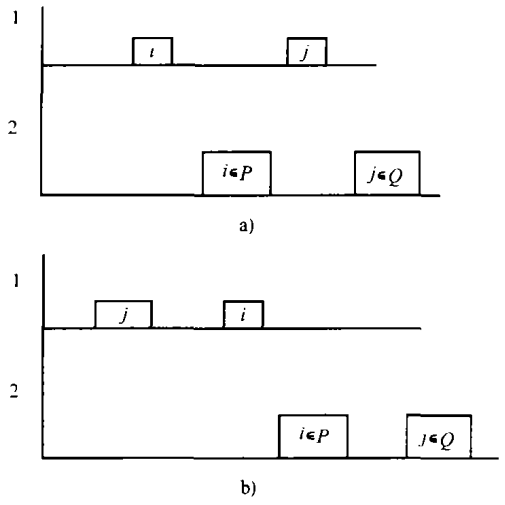

图 1　加工状况

Fig. 1　Status of process

器 1 上互换工件 i 和 j 的位置,在机器 2 上将 i 移入 Q,j 移入 P,易知所得解不比原来差;相应 b 只要在机器 2 上将 i 移入 Q,j 移入 P,也易知所得解不比原来差.将所有这类 i,j 作上述移动即得引理 2.

从引理 2 知,存在这样的最优解,在机器 2 上,若批 P 先于批 Q 加工,则 $\max\{d_j: j\in P\}\leqslant\min\{d_j: j\in Q\}$.因此这些批实际上是将一 EDD 序分段而成,每一段组成同一批,我们称这种特征的分批序为不考虑分批的 EDD 序.

结合引理 1 和引理 2 可推出下面定理 2.

定理 2　在一致性条件,即对任意的 $i\neq j, i,j\in\{1,2,3,\cdots,n\}$,$a_i\leqslant a_j\Rightarrow d_i\leqslant d_j$ 下,存在这样的最优解,其中:

1)在机器 1 上工件依照 EDD 序加工的序为最优序;

2)在机器 2 上只要考虑这样的序:将 EDD 序依次切割成若干段(批),每段(批)中工件数不超过 B_2,依次加工所得段,且每段中的工件同时加工;

3)在机器 2 上,各连续批所加工的工件,其开始加工的时间必相应于机器 1 上某工件的完工时间.

其中连续批是指批处理机从某一时刻开始连续不间断加工的某些批.连续批之内机器没有空转,两连续批之间机器有空转.

证明　由引理 1,引理 2 的证明,即得出 1),2).

因为在机器 2 上每一批工件开始加工的时间必须在这一批工件已全部到达,也就是必须等到此批全部工件在机器 1 上完工之后才能开始在机器 2 上的加工,因此在机器 2 上,各连续批所加工的工件,其开始加工的时间必相应于机器 1 上某工件的完工时间,3)获证.定理证毕.

利用定理 2 知在机器 1 上工件按照 EDD 序工件从时刻 0 开始连续加工,因此只需考虑机器 2 上工件的批处理.容易看出,每当一个工件到达机器 2 上的时侯,都要考虑批处理问题.记 $r_k=\sum_{j=1}^{k}a_j, k\in J=\{1,2,\cdots,n\}$.现对符合一致性条件的问题,构造如下的动态规划算法以确定一最优分批序.

令 $\omega(v)$ 表示在时刻 v 在单处理机上已经加工完毕,等待在批处理机上加工的工件数目,很显然有 $n\geqslant\omega(v)\geqslant 0$.

假设在批处理机上从时刻 r_k 起连续加工 h 批工件,那么其中第一批工件的数目为 $\min\{\omega(r_k), B_2\}$,令 N_v 表示第 v 批工件的数目,$v\leqslant h$,则第 v 批工件在机器 2 上开始处理时间为 $t_v=r_k+(v-1)b$,因此有 $N_v=\min\{\omega(r_k+(v-1)b), B_2\}, v\in\{1,2,3,\cdots,h\}$.其中 $\omega(r_k+(v-1)b)=|\{j: r_k+(v-2)b<r_j\leqslant r_k+(v-1)b\}|+[\omega(r_k+(v-2)b)-B_2]^+$,式中第一部分表示当机器 2 在时间段$[r_k+(v-2)b, r_k+(v-1)b]$正在加工第 $v-1$ 批工件时,机器 1 上新完工的工件及工件数;第二部分表示在时刻 $r_k+(v-2)b$ 已经到达 β 并经第 $v-1$ 批加工后还在等待加工的工件.$|X|$表示集合 X 中元素的数目.

以$(r_k,\omega(r_k))$表示一种状态,因为可有多达 n 个批处理时刻,而 $\omega(r_k)\leqslant n$,所以总的状态有 n^2 个.相应状态$(r_k,\omega(r_k))$,需要决定的是从 r_k 起,在机器 2 上连续加工的工件批数 h 及每批中的工件数 $N_v, v=1,2,\cdots,h$.

定义 $T(k,h)$为状态$(r_k,\omega(r_k))$开始连续加工的 h 批工件对应的最大迟后,那么利用机器 2 上的分批序由 EDD 序分段而得知:

$$T(k,h)=\max\{r_k+b-d_{U(k)+1}, r_k+2b-d_{U(k)+N_1+1},\cdots, r_k+hb-d_{U(k)+N_1+N_2+\cdots N_{h-1}+1}\} \tag{1}$$

其中 $1\leqslant N_1, N_2,\cdots, N_{h-1}, N_h\leqslant B_2, U_k=k-\omega(r_k)$表示时刻 r_k 在机器 2 上已排定的工件数.

令 $H(r_k,\omega(r_k))$表示相应状态$(r_k,\omega(r_k))$的问题最大迟后值,可得递推公式

$$H(r_k,\omega(r_k))=\min\{H(r_{k+1},\omega(r_k)+1), \min_{1\leqslant h\leqslant n}\{\max\{T(k,h), H(r_{k(h)}, \omega(r_k+hb)+1)\}\}\} \tag{2}$$

其中 $r_{k(h)}$表示在连续加工完 h 批工件后接下去一个

连续批可能开始加工的时间，$r_{k(h)}=\min\{r_j: r_j>r_k+hb\}$.

递推公式的边界条件是 $H(r_{n+1},x)=-\infty$，对任何的 x 都成立，最优值是 $H(r_1,1)$.

因为相应每一种状态，式(2)计算量为 $O(n)$，故整个算法计算量为 $O(n^3)$，即：

定理 3 在一致性条件下，对 $F2(1,B_2)|b_j=b$，$\mathrm{BI}|L_{\max}$ 问题，上述动态规划算法可以在 $O(n^3)$ 时间内获得最优解.

一般情况下的问题 $F2(1,B_2)|b_j=b,\mathrm{BI}|L_{\max}$，如果按照 EDD 序排，只能给出近似解. 如此时先将工件在机器 1 上按照 EDD 序排，然后在机器 2 上将上述 EDD 序应用前面的动态规划算法作分批，记这样一个近似算法为 HEDP. 对于目标函数 $L_{\max}$ 来说，由于最优值有可能是负数，因此我们估算近似解与最优解之间的最坏性能比时应用 Masuda[8] 的比率 $\frac{L-L^*}{L^*+d_{\max}}$，其中 L 为从 HEDP 算法中得到的最大迟后，L^* 是原问题的最优值，$d_{\max}$ 为 n 个工件的最迟的应交工时间. 在引入 n 个工件最早的应交工时间 $d_{\min}$ 和 n 个工件的加工时间之和 $P=\sum_{j=1}^{n}a_j$ 后有：

定理 4 $\frac{L-L^*}{L^*+d_{\max}}\leqslant\frac{1}{P+b}\left[\left\lceil\frac{n-B_2}{B_2}\right\rceil b+d_{\max}-d_{\min}\right]$

证明 令 S_t 表示对 n 个工件用 HEDP 算法所得分批序中前 t 个批中工件的集合，其中批序按批-EDD 序排. 令 $L^*(S_t)$ 为对集合 S_t 中工件作最优排列可得的目标函数值. 那么结合 L^* 的定义，我们有 $L^*\geqslant L^*(S_t)$.

令 $L(S_t)$ 为用 HEDP 算法所得分批序中相应 S_t 的最大迟后值，并假设对原问题应用 HEDP 算法时在第二阶段的第 k 批达到最大迟后值，即 $L=L(S_k)$，那么 $L-L^*\leqslant L-L^*(S_k)=L(S_k)-L^*(S_k)$.

记 d^k 为 k 批中工件最早的应交工时间，则有 $L(S_k)=C_{\max}(S_k)-d^k$，其中 $C_{\max}(S_k)$ 为在 HEDP 算法中集合 S_k 中工件的完工时间.

而 $L^*(S_k)\geqslant C_{\max}{}^*(S_k)-d_{\max}$，这里 $C_{\max}{}^*(S_k)$ 为对集合 S_k 中工件作最优排列时产生的最优加工全长，故有

$$L-L^*\leqslant L(S_k)-L^*(S_k)\leqslant[C_{\max}(S_k)-d^k]-[C_{\max}{}^*(S_k)-d_{\max}]$$

两边同除 $L^*+d_{\max}$，结合 $L^*\geqslant L^*(S_k)$ 和 $C_{\max}{}^*(S_k)\leqslant L^*(S_k)+d_{\max}$ 得

$$\frac{L-L^*}{L^*+d_{\max}}\leqslant\frac{[C_{\max}(S_k)-C_{\max}{}^*(S_k)]}{L^*+d_{\max}}+\frac{d_{\max}-d^k}{L^*+d_{\max}}\leqslant\frac{[C_{\max}(S_k)-C_{\max}{}^*(S_k)]}{C_{\max}{}^*(S_k)}+\frac{d_{\max}-d^{\min}}{L^*+d_{\max}}$$

此式对任 $k=1,2,\cdots$，成立，又由

$$C_{\max}{}^*\geqslant\sum_{j=1}^{n}a_j+b\text{ 和 }C_{\max}\leqslant\sum_{j=1}^{n}a_j+\lceil n/B_2\rceil b, L^*+d_{\max}\geqslant P+b$$

便得到定理的结果.

3 问题 $F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|L_{\max}$ 的求解

对于 $F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|L_{\max}$ 这个问题，因为两个阶段都是批处理机，工件在同一台机器上的加工时间相同，因此此问题是问题 $F2(1,B_2)|b_j,\mathrm{BI}|L_{\max}$ 中的工件符合一致性条件的一种特殊情况，即第一阶段的加工时间 $a_j=a$.

定义 2 LOE(n,B)(last-only-empty)规则：批处理机上前$\left(\left\lceil\frac{n}{B}\right\rceil-1\right)$批达到最大容量，即满批，最后一批含有$[n-\left(\left\lceil\frac{n}{B}\right\rceil-1\right)B]$个工件.

定理 5 对于 $F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|L_{\max}$ 问题，存在一个最优解，从时刻 0 开始机器 1 对 n 个工件的加工依照 LOE(n,B_1) 的规则来处理.

证明 因为 $L_{\max}$ 是一个正则目标函数，由 Sung[9] 给出的正则目标函数的性质即知 LOE(n,B_1) 规则对于第一台批处理机是最优的，因此定理成立.

从上面的定理可以看出，如果给定了工件的加工顺序，那么在第一台批处理机上，可从第一个工件开始依次按照批的最大容量来截取工件，直至结束. 而因为置换序全体构成了优势序集合，因此在第二台批处理机上只需按照第一阶段工件的加工顺序加工即可，即有：

定理 6 对问题 $F2(B_1,B_2)|a_j=a,b_j=b,\mathrm{BI}|L_{\max}$，两个阶段中工件均依 EDD 序截断作分批

加工可达最优.

证明　设有工件 i 和 j,$d_i<d_j$,但是工件 i 排在工件 j 的后面,这里"后面"的含义指机器 1 上工件 i 和 j 不在同一批,j 先于 i 加工,或指机器 2 上 i 和 j 不在同一批,j 先于 i 加工,此序记为 π^*,且为最优.相应 π^*,如果工件 i 和 j 在第一阶段处于同一批内,则同时到达第二阶段,此时若 i 和 j 在第二阶段处于不同批,j 先于 i 加工,则只要在阶段 2 交换 i 和 j 的位置即知目标函数值不增大;如果工件 i 和 j 在第一阶段处于不同的批,j 先于 i 加工,但是在第二阶段仍处于同一批,则只需在第一阶段交换 i 和 j 的位置,而两序的最大迟后值仍然相同;而如果在第一和第二阶段 i 和 j 均分在不同批,j 先于 i 加工,则只需在第一和第二阶段工件 i 和 j 的位置均作交换,易知交换后不差于原来的序,定理获证.

在批处理问题已如上解决后,剩下的问题是给出相应的动态规划算法.本节的动态规划算法完全可以仿照上一节的动态规划算法给出.只是批处理时刻由原来的 $r_k=\sum_{j=1}^{k}a_j$ 变成了 $r_k=ka,k=1,2,\cdots,\left\lceil\frac{n}{B_1}\right\rceil$,其余完全相同,这里不再详细描述了.

由于本节问题是前面问题的一个特例,因此所需时间为 $O(n^3)$.

参考文献:

[1] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling[J]. Annual Discrete Math, 1979, 5:287－326.

[2] Lee Chung-yee, Louios R U, Martin-vega A. Efficient algorithms for scheduling semiconductor burn-in operations[J]. Operations Research, 1992,40:764－775.

[3] Ahmadi J H, Ahmadi R H, Dasu S, et al. Batching and scheduling jobs on batch and discrete processors [J]. Operations Research, 1992, 39:750－763.

[4] Potts C N, Struserich V A, Tautenhahn T. Scheduling batches with simultaneous job processing for two machine shop problems [R]. Report, Faculty of Mathematical Studies University of Southampton, UK,1998.

[5] Brucker P, Gladky A, Hoogeveen J A, et al. Scheduling a batch machine[J]. Journal of Scheduling, 1998, 1:31－54.

[6] Hoogeveen H, Van de Velde S. Scheduling by positional completion times: Analysis of a two-stage flow shop problem with a batching machine[J]. Mathematical Programming, 1998, 82: 273－289.

[7] Sung C S, Min J I. Scheduling in a two-machine flowshop with batch processing machine(s) for earliness/tardiness measure under a common due date[J]. European Journal of Operational Research, 2001,131:95－106.

[8] Masuda T, Ishii H, Nishzda I. Some bounds on approximation algorithms for $n|m|I|L_{max}$ and $n|2|F|L_{max}$ scheduling problems[J]. Opns Res Soc Japan, 1983, 26:212－224.

[9] Sung C S, Kim Y H, Yoon S H. A problem reduction and decomposition approach for scheduling for a flowshop of batch processing machines [J]. European Journal of Operational Research, 2000, 121: 179－192.