运筹学与控制论

DOI :10.3969/J. ISSN. 1672-6693.2010.01.001

加工时间为线性增函数有上界的排序问题*

闻振卫, 莫 泽
(苏州大学 数学科学学院, 江苏 苏州 215006)

摘要 :本文讨论工件的加工时间是其开工时间的一类线性增加函数有上界的单机排序问题 $1\mid p_j(t)(t_0,T_1,T_2)\mid C_{max}$ 设工件集 $J=\{J_1,J_2,\cdots,J_n\}$ 中的每个工件需要在一台机器上得到加工 ;工件集 J 被划分成两组 $J=\Omega_1+\Omega_2$ 机器上第一个被加工的工件在时刻 $t_0>0$ 开始加工 ;Ω_1 中工件的加工时间为 $p_j(t)=a_jt$(当 $t<T_1$) 或 $p_j(t)=a_jT_1$(当 $t\geqslant T_1$),Ω_2 中工件的加工时间为 $p_j(t)=a_jt$(当 $t<T_2$) 或 $p_j(t)=a_jT_2$(当 $t\geqslant T_2$),其中 $T_2>T_1>t_0$ 均是给定的常数,t 表示对应工件的开工时刻 排序的目的是极小化时间表长(最大完工时间) $C_{\max}$。在所得的引理 2 和引理 3 的基础上,本文给出一个复杂度为 $n\log n$ 的多项式时间算法,从而也证明了所讨论的问题是多项式时间可解得的。

关键词 排序 加工时间增加 ;最大完工时间
中图分类号 :O223 文献标识码 :A 文章编号 :1672-6693(2010)01-0001-06

在经典排序问题中,一个工件的加工时间是一个固定不变的常数 ;而在某些实际排序问题中,一个工件的加工时间可以是随该工件的开工时间不同而变化的函数。例如在钢铁企业中,某些工件的加工时间与工件的温度有关。当工件的温度落在一定的范围内时,它的加工时间为固定的常数 如果工件在加工之前等待的时间较长,引起工件温度的下降,那么该工件的加工时间会随之增加。这种现象通常被看做工件的加工时间与其开工时间有关的排序环境。对于加工时间与开工时间有关的排序问题人们已经研究了许多不同的模型以及解决了在各种准则下的不同问题[1-4]。

已有的研究加工时间与开工时间有关的排序问题主要处理单机模型[5-8]。Browne S & Yechiali U[9] 研究了如下类型的排序问题 :假设工件 J_j 的加工时间是 $p_j=p_j^0+a_jt$,其中 t 是工件 J_j 的开始加工时间 排序的目的是极小化时间表长(即工件的最大完工时间) $C_{\max}$;并指出工件按 (p_j^0/a_j) 非降的顺序排列得到的是最优排序 ;Mosheiov 在文献[10] 中讨论了工件加工时间为简单线性恶化 $p_j=a_jt$ 的排序问题,并证明了相应的极小化最大完工时间问题是多项式时间可解的 ;张峰在文献[11] 中还研究了加工时间为 $p_j(t)=\begin{cases}a_jt, t<T\\ a_jT, t\geqslant T\end{cases}$,目标函数为极小化最大完工时间的单机排序问题,并指出工件按 a_j 递减的顺序排列得到的是最优排序。

本文在文献[11] 的基础上讨论更一般的排序问题,并给出了多项式时间算法。第一节提出问题 ;第二节分析讨论问题的性质 ;第三节在所得定理的基础上给出问题的多项式时间算法 ;第四、五节给出一个算例及对本文的小结。

1 问题的描述

设有 n 个工件 ;$J=\{J_1,J_2,\cdots,J_n\}$ 需要在一台机器上得到加工 ;工件被划分成两组 $J=\Omega_1+\Omega_2$;集合 Ω_1 中有工件 n_1 个,集合 Ω_2 中有工件 n_2 个($n_1+n_2=n$) 机器上第一个被加工的工件在时刻 $t_0>0$ 开始加工 ;每个工件的加工时间与它的开工时刻有关 ;Ω_1 中工件的加工时间为 $p_j(t)=\begin{cases}a_jt, t<T_1\\ a_jT_1, t\geqslant T_1\end{cases}$,$J_j\in\Omega_1$,$\Omega_2$ 中工件的加工时间为 $p_j(t)=\begin{cases}a_jt, t<T_2\\ a_jT_2, t\geqslant T_2\end{cases}$,$J_j\in\Omega_2$,其中 $T_2>T_1>t_0$ 均是给定的常数,t 表示对应工件的开工时

* 收稿日期 :2009-11-19
基金项目 :国家自然科学基金(No. 10871143)
作者简介 :闻振卫,男,副教授,研究方向为运筹学组合最优化。

刻 排序的目的是极小化时间表长(最大完工时间)$C_{\max}$。

显然,如果$T_1 = T_2$,所述问题就是文献[11]中已讨论的问题 以下总假定$T_1 < T_2$ 如果$t_0 \geqslant T_1$,则Ω_1中所有工件的加工时间均为常数,只要将Ω_1中的全部工件排在Ω_2中的全部工件之后,此情况也等价于文献[11]中的问题。于是,以下总假定$T_2 > T_1 > t_0$。

对于一个给定的工件排列$\delta = (\delta(1),\delta(2),\cdots,\delta(n))$,在不引起混淆下也叫排序,分别用$S_j(\delta)$和$C_j(\delta)$(或在不引起混淆下简单地用$S_j$和$C_j$)表示工件$J_j$(在排序$\delta$下)的开工时间和完工时间,并且总是让$S_{\delta(k)} = C_{\delta(k-1)}(k = 1,2,\cdots,n)$,$C_{\delta(0)} = S_{\delta(1)} = t_0$,即机器上不产生空闲。记这里所讨论的问题为:$1 \mid p_j(t)(t_0,T_1,T_2) \mid C_{\max}$,并且以下假定,在无专门指出时,所论及的问题就是问题$1 \mid p_j(t)(t_0,T_1,T_2) \mid C_{\max}$。

定义 对于一个给定问题$1 \mid p_j(t)(t_0,T_1,T_2) \mid C_{\max}$以及它的一个给定排序$\delta$,如果工件$J_{\delta(k)}$满足$S_{\delta(k)} < T_1 \leqslant C_{\delta(k)}$,则称工件$J_{\delta(k)}$为$T_1$临界工件 如果工件$J_{\delta(h)}$满足$S_{\delta(h)} < T_2 \leqslant C_{\delta(h)}$,则称工件$J_{\delta(h)}$为$T_2$临界工件。

若不存在T_1临界工件,即对于任意一个排序δ,均使得$C_{\max}(\delta) \equiv t_0 \prod_{i=1}^{n}(1 + a_i) < T_1$,那么所讨论问题已解决而无意义。因此以下总假定$t_0 \prod_{i=1}^{n}(1 + a_i) > T_1$,即总假设一定存在$T_1$临界工件。

2 问题的分析与性质

引理1[11] 对于一个给定问题$1 \mid p_j(t)(t_0,T_1,T_2) \mid C_{\max}$,总存在满足如下性质的某最优排序$\delta$ 对于同属于集合$\Omega_i(i = 1$或$2)$中的相邻的两个工件J_j与J_k,如果$a_j > a_k$,则工件J_j排在工件J_k之前加工。

对于一个给定的排序δ,设$J_{\delta(k)}$为T_1临界工件,由文献[10]知,δ上的前$k-1$个工件可按任意顺序排列而不影响第$k-1$个被加工工件的完工时间,事实上,$C_{\delta(k-1)} \equiv t_0 \prod_{i=1}^{k-1}(1 + a_{\delta(i)})$,从而前$k-1$个工件可任意调换位置也并不改变第$n$个被加工工件的完工时间$C_{\max}$。

引理2 对于一个给定的排序δ,如果$S_{\delta(k)} < T_1$(尤其当$J_{\delta(k)}$为T_1临界工件时),$k > 1$,$J_{\delta(k)} \in \Omega_2$,且$J_{\delta(k-1)} \in \Omega_1$,交换工件$J_{\delta(k)}$与工件$J_{\delta(k-1)}$的顺序,其余工件的相对位置保持不变,并记新排序为$\sigma$,即$\sigma(k) = \delta(k-1)$,$\sigma(k-1) = \delta(k)$,对于其余的工件下标$j$,$\sigma(j) = \delta(j)$,那么$C_{\sigma(n)}(\sigma) \leqslant C_{\delta(n)}(\delta)$。

证明 记
$$T = S_{\delta(k-1)}(\delta) = S_{\sigma(k-1)}(\sigma) < C_{\delta(k-1)}(\delta) = T(1 + a_{\delta(k-1)}) = S_{\delta(k)} < T_1 < T_2$$
$$C_{\delta(k)}(\delta) = T(1 + a_{\delta(k-1)})(1 + a_{\delta(k)}),C_{\sigma(k-1)}(\sigma) = T(1 + a_{\sigma(k-1)}) = T(1 + a_{\delta(k)})$$
当$T(1 + a_{\delta(k)}) \leqslant T_1$时
$$C_{\sigma(k-1)}(\sigma) = T(1 + a_{\sigma(k-1)}) = T(1 + a_{\delta(k)}) \leqslant T_1$$
$$C_{\sigma(k)}(\sigma) = T(1 + a_{\sigma(k-1)})(1 + a_{\sigma(k)}) = T(1 + a_{\delta(k-1)})(1 + a_{\delta(k)}) = C_{\delta(k)}(\delta)$$
得$C_{\sigma(n)} = C_{\delta(n)}$;

当$T(1 + a_{\delta(k)}) > T_1$时(若$\delta$最优,则此情形不发生)
$$C_{\sigma(k-1)}(\sigma) = T(1 + a_{\sigma(k-1)}) = T(1 + a_{\delta(k)}) > T_1$$
$$C_{\sigma(k)}(\sigma) = T(1 + a_{\sigma(k-1)}) + a_{\sigma(k)} T_1 = T(1 + a_{\delta(k)}) + a_{\delta(k-1)} T_1$$
$$C_{\delta(k)} - C_{\sigma(k)} = a_{\delta(k-1)} [T(1 + a_{\delta(k)}) - T_1] > 0$$
得$C_{\sigma(n)} < C_{\delta(n)}$。 证毕

引理2说明,对于任意的最优排序δ,对于开工时间$< T_1$的所有工件,Ω_1中的工件一定可以排在Ω_2中的工件之后而不破坏排序的最优性。也就是具体地设$J_{\delta(k)}$为T_1临界工件,则在排序δ上的前k个工件中,一定可以将属于Ω_2的工件排在属于Ω_1的工件的前面而仍保持排序的最优性 除非前k个工件中没有Ω_1中的工件,否则T_1临界工件$J_{\delta(k)}$一定属于Ω_1。

引理3 对于一个给定的排序δ,若$S_{\delta(k)} \geqslant T_1$,$J_{\delta(k)} \in \Omega_1$,且$J_{\delta(k+1)} \in \Omega_2$,交换工件$J_{\delta(k)}$与工件$J_{\delta(k+1)}$的顺序,其余工件的相对位置保持不变,并记新排序为$\sigma$,即$\sigma(k) = \delta(k+1)$,$\sigma(k+1) = \delta(k)$,对于其余的工件下标$j$,$\sigma(j) = \delta(j)$,那么$C_{\sigma(n)}(\sigma) \leqslant C_{\delta(n)}(\delta)$。

证明 记
$$T = S_{\delta(k)}(\delta) = S_{\sigma(k)}(\sigma) \geqslant T_1,C_{\delta(k)}(\delta) = T + a_{\delta(k)} T_1$$

如果 $T_2 \leqslant T$，那么 $C_{\delta(k)}(\delta) = T + a_{\delta(k)} T_1$；$C_{\sigma(k)}(\sigma) = T + a_{\sigma(k)} T_2$，$C_{\delta(k+1)}(\delta) = T + a_{\delta(k)} T_1 + a_{\delta(k+1)} T_2$

$$C_{\sigma(k+1)}(\sigma) = T + a_{\sigma(k)} T_2 + a_{\sigma(k+1)} T_1 = C_{\delta(k+1)}(\delta)$$

于是 $C_{\sigma(n)}(\sigma) = C_{\delta(n)}(\delta)$。

如果 $T < T_2$（若 δ 最优，则此情形不发生），那么

$$C_{\delta(k)}(\delta) = T + a_{\delta(k)} T_1, C_{\sigma(k)}(\sigma) = T(1 + a_{\sigma(k)}) = T(1 + a_{\delta(k+1)})$$

如果 $C_{\delta(k)}(\delta) \geqslant T_2$，那么　$C_{\delta(k+1)}(\delta) = t + a_{\delta(k)} T_1 + a_{\delta(k+1)} T_2$

$$C_{\sigma(k+1)}(\sigma) = T(1 + a_{\sigma(k)}) + a_{\sigma(k+1)} T_1 = T(1 + a_{\delta(k+1)}) + a_{\delta(k)} T_1$$

$$C_{\delta(k+1)}(\delta) - C_{\sigma(k+1)}(\sigma) = a_{\delta(k+1)}(T_2 - T) > 0$$

如果 $C_{\delta(k)}(\delta) < T_2$，那么　$C_{\delta(k+1)}(\delta) = (T + a_{\delta(k)} T_1)(1 + a_{\delta(k+1)})$

$$C_{\sigma(k+1)}(\sigma) = T(1 + a_{\sigma(k)}) + a_{\sigma(k+1)} T_1 = T(1 + a_{\delta(k+1)}) + a_{\delta(k)} T_1$$

$$C_{\delta(k+1)}(\delta) - C_{\sigma(k+1)}(\sigma) = a_{\delta(k)} a_{\delta(k+1)} T_1 > 0$$

于是 $C_{\sigma(n)}(\sigma) < C_{\delta(n)}(\delta)$。　　证毕

引理 3 说明，对于任意的最优排序 δ，对于开工时间 $\geqslant T_1$ 的所有工件，Ω_1 中的工件一定可以排在 Ω_2 中的工件之后而不破坏排序的最优性。也就是具体地设 $J_{\delta(k)}$ 为 T_1 临界工件，则在排序 δ 上的后 $n-k$ 个工件中，一定可以将属于 Ω_2 的工件排在属于 Ω_1 的工件的前面而仍保持排序的最优性。

为了叙述的方便，以下将 Ω_2 中的工件 J_j 的恶化（或腐化）系数 a_j 改记为 b_j，$j = 1, 2, \cdots, n_2$，将 Ω_1 中的工件 J_j 的恶化系数 a_j 仍记为 a_j，$j = n_2 + 1, n_2 + 2, \cdots, n_2 + n_1 = n$，并设工件已排列成 $b_1 \geqslant b_2 \cdots \geqslant b_{n_2}$，$a_{n_2+1} \geqslant a_{n_2+2} \geqslant \cdots \geqslant a_n$。

推论 4　记 $\Omega_{11} \triangleq \{j \in \Omega_1 \mid a_j T_1 \leqslant T_2 - T_1\}$，$\Omega_{12} \triangleq \{j \in \Omega_1 \mid a_j T_1 > T_2 - T_1\}$，则存在最优排序 δ，使 Ω_{11} 中的所有工件一定排在 Ω_2 中的所有工件之后；

推论 4 说明，只有 Ω_{12} 中的工件，即恶化系数 a_j 比较大的工件才有可能排在某些 Ω_2 的工件的前面。

推论 5　若 $\Omega_{11} = \Omega_1$，即 $\Omega_{12} = \varnothing$，则最优排序为 $\delta = (1, 2, \cdots, n)$。

定理 6　存在某最优排序 $\delta = (\delta(1), \delta(2), \cdots, \delta(n))$ 满足：

1）若 T_1 临界工件属于 Ω_2，则 $\delta = (1, 2, \cdots, n)$，设 T_1 临界工件是 $J_s \in \Omega_2 (1 \leqslant s \leqslant n_2)$，那么 $s = \min\{m \mid t_0 \prod_{i=1}^{m} (1 + b_i) \geqslant T_1\}$ 并且，若 T_2 临界工件是 $J_u \in \Omega_2$，且 $u < n_2$，则

$$C_n(\delta) = t_0 \prod_{i=1}^{u} (1 + b_i) + T_2(b_{u+1} + b_{u+2} + \cdots + b_{n_2}) + T_1(a_{n_2+1} + a_{n_2+2} + \cdots + a_n)$$

否则（或 T_2 临界工件是 J_{n_2}，或存在 T_2 临界工件 $\in \Omega_1$，或不存在 T_2 临界工件）

$$C_n(\delta) = t_0 \prod_{i=1}^{n_2} (1 + b_i) + T_1(a_{n_2+1} + a_{n_2+2} + \cdots + a_n)$$

2）若 T_1 临界工件 $J_{n_2+k} \in \Omega_1$，则 $\delta = (1, 2, \cdots, j, n_2 + 1, n_2 + 2, \cdots, n_2 + k, j + 1, j + 2, \cdots, n_2, n_2 + k + 1, n_2 + k + 2, \cdots, n_2 + n_1)$，其中 $0 < k \leqslant n_1$，$0 \leqslant j \leqslant n_2$。并且，若 T_2 临界工件是 $J_{n_2+k} \in \Omega_1$，则

$$C_n(\delta) = t_0 \prod_{i=1}^{j} (1 + b_i) \prod_{t=1}^{k} (1 + a_{n_2+t}) + T_2(a_{j+1} + a_{j+2} + \cdots + a_{n_2}) + T_1(a_{n_2+k+1} + a_{n_2+k+2} + \cdots + a_n)$$

若 T_2 临界工件 $J_u \in \Omega_2$，且 $j < u < n_2$，则

$$C_n(\delta) = t_0 \prod_{i=1}^{j} (1 + b_i) \prod_{t=1}^{k} (1 + a_{n_2+t}) \prod_{i=j+1}^{u} (1 + b_i) + T_2(a_{u+1} + a_{u+2} + \cdots + a_{n_2}) + T_1(a_{n_2+k+1} + a_{n_2+k+2} + \cdots + a_n)$$

若 T_2 临界工件是 $J_{n_2} \in \Omega_2$，或存在 T_2 临界工件 $\in \Omega_1$，或不存在 T_2 临界工件，则

$$C_n(\delta) = t_0 \prod_{i=1}^{j} (1 + b_i) \prod_{t=1}^{k} (1 + a_{n_2+t}) \prod_{i=j+1}^{n_2} (1 + b_i) + T_1(a_{n_2+k+1} + a_{n_2+k+2} + \cdots + a_n)$$

证明　设存在某最优排序 δ'，由前面的引理 1、2、3 可知，对 δ' 进行调整，使调整后的排序可表示为 $\delta = (\delta(1), \delta(2), \cdots, \delta(n))$ 满足定理 6 的条件。

可见满足定理6条件的排序至多只有 $s+1$ 个(如果存在定理6情形(1)中所定义的 s)或至多只有 n_2+1 个(如果不存在这样的 s)。 证毕

推论7 若2个排序 δ_j 和 δ_{j-1} 具有

$$\delta_j=(1,2,\cdots,j,n_2+1,\cdots,n_2+k,j+1,\cdots,n_2,n_2+k+1,\cdots,n_2+n_1)$$

其中,$J_{n_2+k}(\in\Omega_1)$ 是 T_1 临界工件,$2\leqslant j\leqslant n_2$,$0<k\leqslant n_1$;

$$\delta_{j-1}=(1,\cdots,j-1,n_2+1,\cdots,n_2+k',j,\cdots,n_2,n_2+k'+1,\cdots,n_2+n_1)$$

其中,$J_{n_2+k'}(\in\Omega_1)$ 是 T_1 临界工件,并且 $k'=k$(通常情况下 $k'\geqslant k$),则 $C_{\max}(\delta_{j-1})\leqslant C_{\max}(\delta_j)$。

推论7说明,当属于 Ω_1 的 T_1 临界工件 J_{n_2+k} 确定之后,最优排序 δ_j 一定可以是让 j 达到最小的那个排序,即 $j=\min\{j\mid t_0\prod_{i=1}^{j}(1+b_i)\prod_{i=1}^{k}(1+a_i)\geqslant T_1\}\geqslant 0$。

3 求解问题的算法

算法

步骤0 设工件已排列成 $J_1,J_2,\cdots,J_{n_2}\in\Omega_2,J_{n_2+1},J_{n_2+2},\cdots,J_n\in\Omega_1$,并具有 $b_1\geqslant\cdots\geqslant b_{n_2}$,$a_{n_2+1}\geqslant\cdots\geqslant a_n$ 若 $t_0\prod_{i=1}^{n_2}(1+b_i)\geqslant T_1$,则令 $j=s=\min_m\{m\mid t_0\prod_{i=1}^{m}(1+b_i)\geqslant T_1\}\geqslant 1$,否则,令 $j=s=n_2$;

步骤1 对于当前的 j,若 $t_0\prod_{i=1}^{j}(1+b_i)\prod_{l=1}^{n_1}(1+a_{n_2+l})\geqslant T_1$,则令

$$k=\min_m\{m\mid t_0\prod_{i=1}^{j}(1+b_i)\prod_{l=1}^{m}(1+a_{n_2+l})\geqslant T_1\}\geqslant 0$$

以下记排序 $\delta_j=(1,2,\cdots,j,n_2+1,\cdots,n_2+k,j+1\cdots,n_2,n_2+k+1,\cdots,n)$(其中 k 是由 j 所确定的),再令 $j=\min_j\{j\mid t_0\prod_{i=1}^{j}(1+b_i)\prod_{l=1}^{k}(1+a_{n_2+l})\geqslant T_1\}\geqslant 0$,以下记排序 $\delta^{n_2+k}=(1,2,\cdots,j,n_2+1,\cdots,n_2+k,j+1,\cdots,n_2,n_2+k+1,\cdots,n)$(其中 j 是由 k 所确定的),并计算它的目标函数值;

若 $j=0$,则转步骤2;

否则,令 $j:=j-1$,返回步骤1。

否则,直接转步骤2。

步骤2 从已得到的(至多只有 n_2+1 个)排序中选出最好者,即为最优排序。

容易判断本算法的复杂性是 $O(n\log n)$。

4 算例

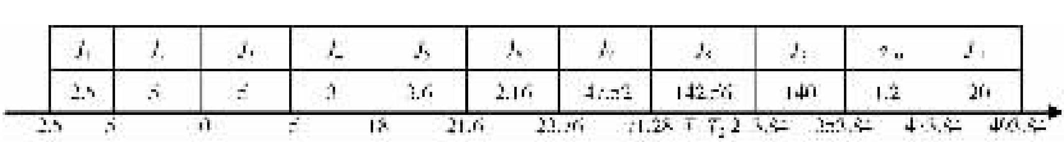
本算法的执行中,依次地在计算排序 $\delta_s,\delta_{s-1},\delta_{s-2},\cdots$。在工件 J_j 向后移动得到下一个排序 δ_{j-1} 的过程中,由于 a_{n_2+k+1} 比 b_j 大得较多,除非原属于 Ω_1 的 T_1 临界工件未发生变动,否则至多只有一个工件 $J_{n_2+k+1}(\in\Omega_1)$ 向前移动而成为新的 T_1 临界工件。

例 $n=11$,$n_1=5$,$n_2=6$;$t_0=2.5$,$T_1=100$,$T_2=150$

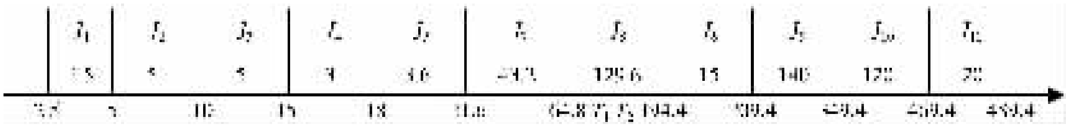
b_1	b_2	b_3	b_4	b_5	b_6	a_7	a_8	a_9	a_{10}	a_{11}
1.0	1.0	0.5	0.2	0.2	0.1	2.0	2.0	1.4	1.2	0.2

解 $s=n_2=6$;

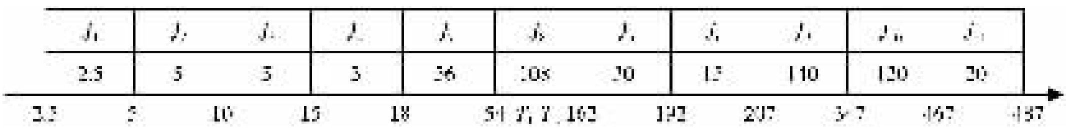
若排序为 $\delta_6=(1,2,\cdots,11)$,则甘特图是:



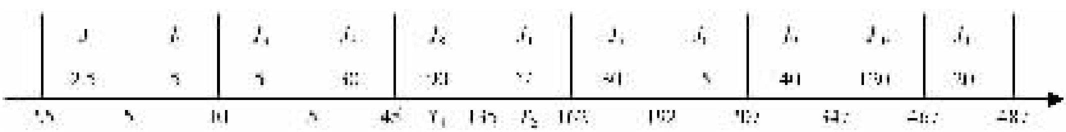
若排序为 $\delta_5 = (1, 2, 3, 4, 5, 7, 8, 6, 9, 10, 11)$，则甘特图是：



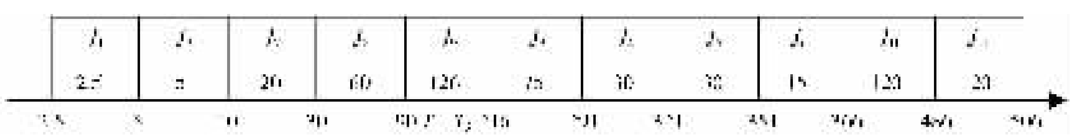
若排序为 $\delta_4 = (1, 2, 3, 4, 7, 8, 5, 6, 9, 10, 11)$，则甘特图是：(最优)



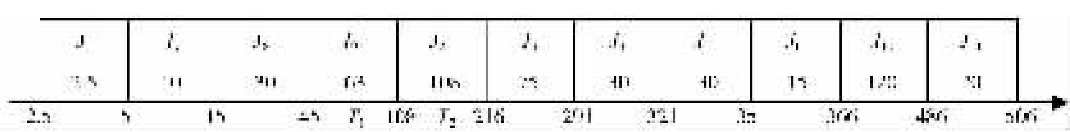
若排序为 $\delta_3 = \delta^8 = (1, 2, 3, 7, 8, 4, 5, 6, 9, 10, 11)$，则甘特图是：(最优)



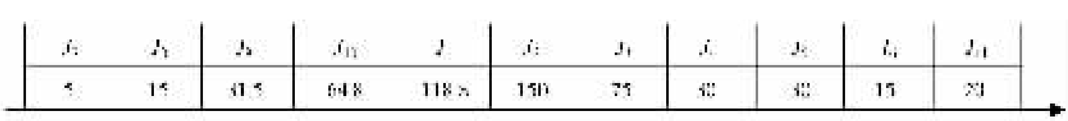
若排序为 $\delta_2 = (1, 2, 7, 8, 9, 3, 4, 5, 6, 10, 11)$，则甘特图如下：
尽管 $a_9 (=1.4) > b_3 (=0.5)$，但排序 δ_2 不如排序 δ_3 优。



若排序为 $\delta_1 = \delta^9 = (1, 7, 8, 9, 2, 3, 4, 5, 6, 10, 11)$，则甘特图是：



若排序 $\delta_0 = \delta^{10} = (7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 11)$，则甘特图如下：
尽管 $a_{10} (= 1.2) > b_1 (= 1.0)$，但排序 δ_1 比排序 δ_2 更差。



于是，最优排序是 $\delta_3 = (1, 2, 3, 4, 7, 8, 5, 6, 9, 10, 11)$，或 $\delta_4 = \delta^8 = (1, 2, 3, 7, 8, 4, 5, 6, 9, 10, 11)$，最优值为 $C_{\max} = 487$。

5 小结

本文讨论了问题 $1 \mid p_j(t)(t_0, T_1, T_2) \mid C_{\max}$，并给出一个有效算法。从引理 2 和引理 3 可知，Ω_1 中的工件更倾向于排在 Ω_2 中的工件之后。而由文献[11]知，当 $T_1 = T_2$ 时，工件按 a_j(包括 b_j)从大到小排列得到最优排序，因此，当 T_1 与 T_2 相差很小而 a_i 比 b_j 大得较多时，Ω_1 中的工件 J_i 仍应该排在 Ω_2 中的工件 J_j 之前；也就是说，当 a_i 比 b_j 大得很多时，工件 J_i 应排在工件 J_j 之前，这正是本文算法的步骤 1 中将工件 J_j 向后移动(到工件 J_i 之后)的理由。

参考文献：

[1] Alidaee B, Womer N K. Scheduling with time dependent processing times: review and extensions[J]. Journal of the Operational

Research Society, 1999, 50: 711-720.

[2] Cheng T C E, Ding Q, Lin B M T. A concise survey of scheduling with time-dependent processing times[J]. European Journal of Operational Research, 2004, 152: 1-13.

[3] 唐国春. 误工排序问题的研究[J]. 重庆师范大学学报(自然科学版), 2009, 26(2): 1-6.

[4] 刘朝晖. 批处理机上有就绪和截止时间的等长度工件排序[J]. 重庆师范大学学报(自然科学版), 2009, 26(3): 1-4.

[5] Biskup D. Single-machine scheduling with learning considerations[J]. European Journal of Operational Research, 1999, 115(1): 173-178.

[6] Zhao C L, Tang H Y. Single machine scheduling problems with deteriorating jobs[J]. Applied Mathematics and Computation, 2005, 161: 865-874.

[7] Gupta J N D, Gupta S K. Single facility scheduling with nonlinear processing times[J]. Computers and Industrial Engineering, 1988, 14: 387-393.

[8] Sundararaghavan P S, Kunnathur A S. Single machine scheduling with start time-dependent processing times: Some solvable cases [J]. European Journal of Operational Research, 1994, 78(3): 394-403.

[9] Browne S, Yechiali U. Scheduling deteriorating jobs on a single processor [J]. Oper. Res, 1990, 38: 495-498.

[10] Mosheiov G. Scheduling jobs under simple linear deterioration [J]. Comput. Oper. Res, 1994, 21(6): 653-659.

[11] 张峰. 工件加工时间增加的排序问题($1 \| C_{max}$)[J]. 高校应用数学学报 A 辑, 2000, 16(2): 228-234.

Single Machine Scheduling to Minimize Makespan with Bounded Linear Increasing Processing Time

WEN Zhen-wei, MO Ze

(School of Mathematical Sciences, Soochow University, Suzhou 215006, China)

Abstract: Traditional scheduling problems usually involve jobs with constant processing times. In this paper, a single machine scheduling problem $1 \mid p_j(t)(t_0, T_1, T_2) \mid C_{max}$ is discussed, in which the processing time of a job is a bounded linear increasing function of its starting time. Each job in the job set $J = \{J_1, J_2, \cdots, J_n\}$ is suppose to be processed on a single machine. The job set is partitioned into two subsets $J = \Omega_1 + \Omega_2$. The first job processed on the machine stars at time $t_0 > 0$. The processing time of a job in Ω_1 is $p_j(t) = a_j t$ (if $t < T_1$) or $p_j(t) = a_j T_1$ (if $t \geqslant T_1$), and the processing time of a job in Ω_2 is $p_j(t) = a_j t$ (if $t < T_2$) or $p_j(t) = a_j T_2$ (if $t \geqslant T_2$), where $T_2 > T_1 > t_0$ are all given constants, and t is the starting time of the job. The objective is to minimize the makespan C_{max}. Based on Lemma 2 and Lemma 3 obtained, we show that this scheduling problem is polynomial time solvable and a polynomial time algorithm with complexity $n \log n$ is proposed.

Key words: scheduling; increasing processing times; makespan; polynomial time algorithm

(责任编辑 黄 颖)