运筹学与控制论

DOI:10.11721/cqnuj20140604

带退化效应和不可用区间的并行批处理机问题*

沈晓飞，赵玉芳，王晓丹
(沈阳师范大学 数学与系统科学学院，沈阳 110034)

摘要:本文研究的是一类带有不可用区间和线性退化效应的单机无界并行批处理机排序问题。工件开始加工时间的线性递增函数看成其实际的加工时间。批工件中加工时间的最大者为这批的加工时间,同批工件同时开始加工,且批一旦开始加工就不可中断,同批中工件的完工时间都相同并为这批的完工时间。本文通过对最优解性质的分析,分别给出了求解极小化最大费用和极小化总费用的拟多项式时间算法。特别当 k 固定、目标函数为误工工件数时,该问题为多项式时间可解的,并用数值例子验证了算法的有效性。

关键词:并行批;不可用区间;退化效应;拟多项式算法
中图分类号:O223　　**文献标志码**:A　　**文章编号**:1672-6693(2014)06-0016-06

最早研究并行批处理机问题的是 Lee 等[1]人,他们对目标函数为最大延误和误工数的排序问题都给出了有效的动态规划算法。Brucker 等人[2]将并行批处理机问题分为两种模型:1)无界模型—批处理机的容量 B 无限,即 $B\geqslant n$;2)有界模型—批处理机的容量 B 有限,即 $B<n$。对于并行批处理机为单机无界模型,且目标函数为最大完工时间时,Lu 和 Yuan[3]研究的是工件大小不同且带有交货期的排序问题,分别给出了相应的最优算法;Lee 和 Uzsoy[4]对工件动态到达时进行了研究,目标函数为最大完工时间的排序问题,给出了多项式动态规划算法。

对于并行批处理机工作过程中出现维修、保养等现象,并使并行批处理机带有不可用区间,从而影响工件的加工过程的问题,当目标函数为最大完工时间时,Lee[5]给出了最优算法。对于单机情况,当中断可恢复时,是多项式时间可解的;中断不可恢复时,是 NP-难的。对于单机并行批排序问题,当处理机带有 k 个不可用区间的,Yuan 等人[6]证明了批处理机容量无限且工件带有释放时间时,目标函数为最大完工时间问题在多项式时间内可解。对于处理机带有不可用区间、工件可拒绝的单机排序问题,张敏娇等人[7]研究了极小化被加工工件的总完工时间与被拒绝工件的总惩罚和,利用削减状态空间方法得到了一个全多项式时间近似方案(FPTAS)。

在经典的排序理论中,工件的加工时间往往是一个常数,然而在很多实际问题中,工件带有退化效应。对于工件带有退化效应的模型且同时并行批处理机的容量无限时,Li 等人[8]研究了工件带有释放时间、目标函数为最大完工时间的排序问题;Qi 等人[9]研究了目标函数分别为最大费用、误工工件数与总加权完工时间的排序问题;这一类问题都是多项式可解的。Ji 等人[10]研究的处理机带有不可用区间(不可恢复),目标函数为最大完工时间和总完工时间的问题都是 NP-难的;对于总完工时间问题,他们给出的是启发式算法。Jafari 和 Moslehi[11]研究的是工件带有退化效应的单机排序问题,证明了误工工件数问题是 NP-难的。

在工业的生产过程中,常常有带有退化效应的工件,以及带有不可用区间的并行批处理机同时出现的情况。对于带有 k 个不可用区间的排序问题,Yuan 等人[6]证明了批处理机容量无限且工件带有释放时间时,目标函数为最大完工时间问题是多项式时间内可解的。沈晓飞等人[12]研究了带有线性退化效应和不可用区间的单机无界并行批处理机排序问题,给出了求解最大完工时间的的多项式时间算法。在沈晓飞等人[12]研究的基础上,本文讨论了目标函数分别为最大费用及总费用的单机排序问题。本文第 1 节给出了问题描述;第 2 节给出了基本定义;第 3 节给出了最优解性质以及拟多项式时间的算法,并用数值例子对算法进行了验证。

1 问题描述

本文考虑工件集 $J=\{J_1,J_2,\cdots,J_n\}$,n 个相互独立的且无优先约束的工件在一台并行批处理机上加工。工

* **收稿日期**:2013-09-16　**修回日期**:2014-01-03　**网络出版时间**:2014-11-19 21:49
资助项目:辽宁省教育厅科学技术研究项目(No. L2014433)
作者简介:沈晓飞,女,研究方向为组合最优化,E-mail:719625617@qq.com;**通讯作者**:赵玉芳,E-mail:yfzhao2004@163.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20141119.2149.004.html

件 J_j 的释放时间都相同，工期 $d_j\geqslant 0$，权 $w_j\geqslant 0$。工件 J_j 的基本加工时间为 $a_j\geqslant 0$，那么实际加工时间为 $p_j=a_j+bt$，其中，t 为其开始加工时间，$b>0$ 表示工件的退化率。

处理机有 k 个不可用区间：$[h_{1,1},h_{1,2})$，$[h_{2,1},h_{2,2})$，…，$[h_{k,1},h_{k,2})$，当 $1\leqslant q\leqslant k$ 时，$h_{q,1}<h_{q,2}$，$h_{q,2}<h_{q+1,1}$，$(h_{0,2}=0,h_{k+1,1}=\infty)$。因此，工件只能在形成的 $k+1$ 个可用的区间段$[h_{0,2},h_{1,1})$，$[h_{1,2},h_{2,1})$，…，$[h_{k-1,2},h_{k,1})$，$[h_{k,2},h_{k+1,1})$内加工。设所有工件 $J_1,J_2,\cdots,J_n$ 被分成 m 个批的形式进行加工，即 $B_1,\cdots,B_m$。在并行批处理机中，定义批中工件加工时间的最大者等于这一批的加工时间，同一批中的工件同时开始加工且同时完工，批的完工时间等于批中最后一个工件的完工时间。记批 B_l 的开始加工时间等于 s_l，批 B_l 的基本加工时间等于 $a_{(l)=\max\{a_j\mid J_j\in B_l\}}$，那么批 B_l 的加工时间和完工时间记为 $p_{(l)}$ 和 $C_{(l)}$，即 $p_{(l)}=\max\{a_j+bs_l\mid J_j\in B_l\}=\max\{a_j\mid J_j\in B_l\}+bs_l=a_{(l)}+bs_l$，$C_{(l)}=s_l+p_{(l)}=s_l+a_{(l)}+bs_l=a_{(l)}+(1+b)s_l$。也就是批 B_l 中工件 J_j 的完工时间为

$$C_j=s_l+a_{(l)}+bs_l=a_{(l)}+(1+b)s_l。\tag{1}$$

本文研究的排序问题中批处理机可以同时加工无穷多个工件，即批处理机容量为无限。为表示方便，将并行批处理机记为 *p-batch*，不可用区间记为 FB。本文研究的目标函数为极小化正则目标函数，即极小化最大费用 $f_{\max}$和极小化总费用 $\sum f_j$，用三参数表示法记为

$$1\mid p\text{-}batch,FB,p_j=a_j+bt\mid f_{\max},1\mid p\text{-}batch,FB,p_j=a_j+bt\mid\sum f_j,$$

其中，*p-batch* 为并行批处理机，FB 为不可用区间。

2 基本定义

以下为了叙述方便，给出下面的定义和引理。

定义 1　对于每个正则目标函数，称满足下面两个条件的排序 $\pi=(BS;ST)$为可行排序，其中 $BS=(B_1,B_2,\cdots,B_m)$为批序列，对应的开始加工时间序列为 $ST=(s_1,s_2,\cdots,s_m)$。

1) 批的开始加工时间不小于前一批的完工时间，即对于 $1\leqslant l\leqslant m-1$，$s_{l+1}\geqslant a_{(l)}+(1+b)s_l$；

2) 任一批一定在某一可用时间段内加工完，即对于批 $B_l(1\leqslant l\leqslant m-1)$，存在 $i(1\leqslant i\leqslant k+1)$，使得 $s_l\in[h_{i-1,2},h_{i,1})$，且 $s_l+p_{(l)}\in[h_{i-1,2},h_{i,1})$，其中 $h_{0,2}=0,h_{k+1,1}=\infty$。

定义 2　在最优排序 $\pi=(BS;ST)$中，称批序列 BS 为最优批序列。

为了说明批 B_l 的开始加工时间，也就是对于前一批 B_{l-1}的完工时间 t，其中 t 为非负正整数，设 $i(1\leqslant i\leqslant k)$是满足 $t\in[h_{i-1,2},h_{i,1})$的最小整数，若 $a_{(l)}+(1+b)t\leqslant h_{i,1}$，那么批 B_l 的开始加工时间等于 t。否则，找最小 $h_{q,2}$，使其满足 $a_{(l)}+(b+1)h_{q,2}\leqslant h_{q+1,1}$，此时批 B_l 的开始加工时间等于 $h_{q,2}$。综上，笔者给出如下定义。

定义 3　称满足

$$S(t,a_{(l)})=\begin{cases}t,\ \text{如果 } a_{(l)}+(1+b)t\leqslant h_{i,1}\\ \min\{h_{q,2}:a_{(l)}+(1+b)h_{q,2}\leqslant h_{q+1,1}\},\ \text{否则}\end{cases}\tag{2}$$

的 $S(t,a_{(l)})$为不可能在 t 之前开始加工的批 B_l 的可能开始加工时间。

因为不可用区间有 k 段，故 $S(t,a_{(l)})$可以在 $O(k)$时间内得到。由于批 B_l 不能在 $C_{(l-1)}$ 之前开始加工，令 $s_0=0,a_{(0)}=0$，则 B_l 的开始加工时间 s_l 可以由以下公式计算得出

$$s_l=S(a_{(l-1)}+(1+b)s_{l-1},a_{(l)})。\tag{3}$$

引理 1　对于每个正则目标函数，一定存在一最优批序列 $BS=(B_1,B_2,\cdots,B_m)$，对于任意两批 B_l 和 B_k，当 $l<k$ 时，有下式

$$\max\{a_i:J_i\in B_l\}<\min\{a_j:J_j\in B_k\}。\tag{4}$$

证明　假设 $BS=(B_1,B_2,\cdots,B_m)$为最优批序列，对应的批的开始加工时间序列为 $ST=(s_1,s_2,\cdots,s_m)$，$\pi=(BS;ST)$是一个最优排序。设在 BS 中存在两个批 B_l和 $B_k(l<k)$，有 $\max\{a_i:J_i\in B_l\}\geqslant\min\{a_j:J_j\in B_k\}$，即存在 $J_h\in B_k$，且 $a_h\leqslant\max\{a_i:J_i\in B_l\}$。将 J_h 移入批 B_l 中，得到新的批序列$\overline{BS}$，其中 $\overline{B}_k=B_k\backslash\{J_h\}$，$\overline{B}_l=B_l\cup\{J_h\}$（若 $B_k=\{J_h\}$，则 $\overline{B}_k=\varnothing$）。记$\overline{BS}$对应批开始加工时间序列为$\overline{ST}$（$\overline{s}_l=s_l,\overline{s}_k=s_k$），得到新排序 $\overline{\pi}=(\overline{BS};\overline{ST})$。因为 $a_h\leqslant\max\{a_i:J_i\in B_l\}$，所以 $\overline{p}_{(l)}=p_{(l)}$，$\overline{p}_{(k)}\leqslant p_{(k)}$，其余批不变，因此 $\overline{\pi}$ 也是一个可行排序。对于新的可行排序 $\overline{\pi}$，由于各批的完工时间没有增加，所以排序 $\overline{\pi}$ 也是原问题的最优排序。重复上述过程，只要经过有限次调换，可得结论成立。　　证毕

从引理1的证明可知,经过调换之后,仍然是最优排序,即任意一个最优排序都可以经有限次调换,使得对于任意两批 B_l 和 B_k,当 $l<k$ 时,有 $\max\{a_i:J_i\in B_l\}<\min\{a_j:J_j\in B_k\}$。设批内工件都是按基本加工时间非减的顺序排列的。因此称满足引理1的批序列为批-SBPT(Shortest basic processing time first,最短基本加工时间优先)序的。

引理 2 对于每个正则目标函数,一定存在一个最优排序是批-SBPT序的。

由引理1和引理2,假设工件按基本加工时间非减顺序重新排列,即 $a_1\leqslant a_2\leqslant\cdots\leqslant a_n$,那么批-SBPT序可写成$(B_1,B_2,\cdots,B_m)$,其中每一批 $B_l(1\leqslant l\leqslant m)$为以下形式

$$B_l=\{J_{i_l},J_{i_l+1},\cdots,J_{i_{(l+1)}-1}\},(1=i_1<i_2<\cdots<i_{(m+1)}=n+1)。$$

也就是在最优批序列中,每批中的工件,一定是下标连续的。因为在并行批处理机中,批中工件加工时间的最大者等于这一批的加工时间,所以每一批的加工时间由这一批中有最大基本加工时间的工件决定。

引理 3 对于批-SBPT序列$(B_1,B_2,\cdots B_m)$,记 $D=\sum_{1\leqslant j\leqslant n}(1+b)^{n-j}a_j+(1+b)^n h_{k,2}$,则每个工件的完工时间都不超过 D。

证明 最坏的情况是:每批工件不能在任意可加工区间段内加工完,那么第一批工件只能在 $h_{k,2}$ 时刻开始加工。而并行批处理机的加工特点是每一批中工件加工时间中最大者等于这一批的加工时间,所以对于工件集 J 中的工件,不分批加工时间最长。此时 J_j 的完工时间计算如下

$$C_{(1)}=a_{(1)}+(1+b)h_{k,2},$$

$$C_{(2)}=a_{(2)}+(1+b)C_{(1)}=a_{(2)}+(1+b)a_{(1)}+(1+b)^2h_{k,2},$$

$$C_{(3)}=a_{(3)}+(1+b)C_{(2)}=a_{(3)}+(1+b)a_{(2)}+(1+b)^2a_{(1)}+(1+b)^3h_{k,2},$$

$$\cdots\cdots$$

$$C_{(n)}=a_{(n)}+(1+b)C_{(n-1)}=$$

$$a_{(n)}+(1+b)a_{(n-1)}+(1+b)^2a_{(n-2)}+\cdots+(1+b)^{n-1}a_{(1)}+(1+b)^nh_{k,2}=$$

$$a_n+(1+b)a_{n-1}+(1+b)^2a_{n-2}+\cdots+(1+b)^{n-1}a_1+(1+b)^nh_{k,2}=$$

$$\sum_{1\leqslant j\leqslant n}(1+b)^{n-j}a_j+(1+b)^nh_{k,2}。$$

记 $D=\sum_{1\leqslant j\leqslant n}(1+b)^{n-j}a_j+(1+b)^nh_{k,2}$,则每个工件的完工时间都不超过 D。 证毕

3 问题的算法

3.1 1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}$

在这一部分,首先讨论问题1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}$的判定问题1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}\leqslant Y$,即对于给定的 Y 值,是否存在可行排序 π,使得 $f_{\max}(\pi)\leqslant Y$ 成立。

根据引理3,对于任意排序 π,每个工件的完工时间都不超过 D,因此,若令 $\Delta=\max_{1\leqslant j\leqslant n}f_j(D)$,则判定问题中的最小整数 Y 可通过二维搜索方法在$[0,\Delta]$内找到。

对于任意整数$Y\in[0,\Delta]$,如果判定问题1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}\leqslant Y$ 在 $O(F(n,k))$时间内可解,那么原问题1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}$就可在 $O(F(n,k)\log\Delta)$时间内解出。

定义 $d_j=\max\{C_j;f_j(C_j)\leqslant Y,1\leqslant j\leqslant n\}$,则 $L_j=C_j-d_j\leqslant 0$,即 $L_{\max}\leqslant 0$。当 $C_j\in[0,D]$时,d_j 可通过二维搜索方法在 $O(\log D)$时间内找到。即,对于给定的 Y 值,用 $O(n\log D)$时间就可以确定所有的 $d_j(1\leqslant j\leqslant n)$值。

由以上讨论得以下定理。

定理 1 判定问题1$|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}\leqslant Y$ 可以在 $O(n\log D)$时间内归约到问题1$|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$。

以下讨论问题1$|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$。

对于给定的 $j,1\leqslant j\leqslant n,\xi_j=\{J_1,J_2,\cdots,J_j\}$中的工件在处理机上加工。对于 ξ_j 中工件,如果排序 π 满足 $L_{\max}\leqslant 0$,那么称 π 为 ξ_j-可行排序。设

$$F(j)=\min\{C_j(\pi):\text{其中}\ \pi\ \text{是}\ \xi_j\text{-可行排序}\}。$$

对于 ξ_j 中工件,若排序 π 不满足 $L_{\max}\leqslant 0$,则没有 ξ_j-可行排序,此时令 $F(j)=\infty$。如果 ξ_j-可行排序中,最后一批工件为$\{J_{i+1},\cdots,J_j\}$,开始加工时间为 $S(F(i),a_j)$,那么

$$F(j)=S(F(i),a_j)+a_j+bS(F(i),a_j)=a_j+(1+b)S(F(i),a_j)。$$

由以上讨论，有以下动态规划算法。

算法 DP1

边界条件：$F(0)=0$。

递归函数：$F(j)=a_j+(1+b)\min\{S(F(i),a_j)+\delta(i,j):0\leqslant i\leqslant j-1\}$，其中

$$\delta(i,j)=\begin{cases}0, & a_j+(1+b)S(F(i),a_j)\leqslant d_l;i+1\leqslant l\leqslant j\\ \infty, & \text{否则}\end{cases}。$$

由(2)式，当 $1\leqslant q\leqslant k$ 时，开始加工时间函数

$$S(F(i),a_j)=\begin{cases}F(i), & a_j+(1+b)F(i)\leqslant h_{l,1}\\ \min\{h_{q,2}:a_j+(1+b)h_{q,2}\leqslant h_{q+1,1}\}, & \text{否则}\end{cases}。$$

这里 $l(1\leqslant l\leqslant k)$ 是满足 $F(i)\in[h_{l-1,2},h_{l,1})$ 的最小整数。

目标函数的最优值：$\max\{F(j):1\leqslant j\leqslant n\}$

问题 $1|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$ 有解当且仅当 $F(n)<\infty$。因此，对于上述动态规划算法 DP1，有以下定理。

定理 2　用上述动态规划算法 DP1 求解问题 $1|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$ 的时间复杂性为 $O(n^3k)$。

由定理 1 和定理 2，有以下定理。

定理 3　问题 $1|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}$ 在 $O(n(nk+\log D)\log\Delta)$ 时间内可解。

证明　对于 $Y\in[0,\Delta]$，由定理 1，判定问题 $1|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}\leqslant Y$ 可在时间 $O(n\log D)$ 内归约到问题 $1|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$；由定理 2 可知，$1|p\text{-}batch,FB,p_j=a_j+bt|L_{\max}\leqslant 0$ 可以在 $O(n^3k)$ 时间内解出，所以 $1|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}\leqslant Y$ 可以在 $O(n(n^2k+\log D))$ 时间内解出。综上，原问题 $1|p\text{-}batch,FB,p_j=a_j+bt|f_{\max}$ 在 $O(n(n^2k+\log D)\log\Delta)$ 时间内可解。　证毕

3.2 $1|p\text{-}batch,FB,p_j=a_j+bt\Big|\sum f_j$

在本节中，讨论目标函数为一般求和形式的问题 $1|p\text{-}batch,FB,p_j=a_j+bt\Big|\sum f_j$。

设 $N=\max\{f_j\left(\sum_{1\leqslant j\leqslant n}(1+b)^{n-j}a_j+(1+b)^n h_{k,2}\right):1\leqslant j\leqslant n\}$，一定存在批-SBPT 序的排序 π，满足 $f_j(C_j(\pi))\leqslant N$。在最优批序列中，每批中的工件一定是下标连续的，因此在最优批-SBPT 序中，若批 B_l 包含工件 J_i 和 $J_j(i<j)$，则 B_l 一定包含工件 $J_i,J_{i+1},J_{i+2},\cdots,J_j$，下面推导求解这个问题的动态规划算法。

设 $H(z,j)$ 表示批-SBPT 序中工件 J_j 的最小完工时间，此时问题的目标函数值恰好为 z。若不存在目标函数值恰好为 z 的批-SBPT 排序，则定义 $H(z,j)=\infty$。最后一批工件集为：$\Omega=\{J_i,J_{i+1},\cdots,J_j\},1\leqslant i\leqslant j$，由于 $a_1\leqslant\cdots\leqslant a_j$，所以 Ω 的基本加工时间为 a_j。因此，$\sum f_j=\sum_{i\leqslant l\leqslant j}f_l(H(z,j))$，前 $i-1$ 个工件排好之后的目标函数值为 $z^*=z-\sum_{i\leqslant l\leqslant j}f_l(H(z,j))$，那么，工件 J_j 的完工时间计算如下

$$H(z,j)=a_j+(1+b)S(H(z^*,i-1),a_j),$$

其中 $z^*+\sum_{i\leqslant l\leqslant j}f_l(a_j+(1+b)S(H(z^*,i-1),a_j))=z$。

由以上讨论，有以下动态规划算法。

算法 DP2

边界条件：$H(0,0)=0,H(z,0)=\infty,z>0$。

递归函数：$H(z,j)=a_j+(1+b)\min\limits_{1\leqslant i\leqslant j}\min\limits_{z^*\in\sigma}S(H(z^*,i-1),a_j)$，其中

$$\sigma=\{z^*:z^*+\sum_{i\leqslant l\leqslant j}f_l(a_j+(1+b)S(H(z^*,i-1),a_j))=z\},$$

$$S(H(z^*,i-1),a_j)=\begin{cases}H(z^*,i-1), & \text{如果 } a_j+(1+b)H(z^*,i-1)\leqslant h_{l,1}\\ \min\{h_{q,2}:a_j+(1+b)h_{q,2}\leqslant h_{q+1,1}\}, & \text{否则}\end{cases}。$$

这里 $l(1\leqslant l\leqslant k)$ 是满足 $H(z^*,i-1)\in[h_{l-1,2},h_{l,1})$ 的最小整数。

目标函数最优值：$\min\{z\in[0,nN]:H(z,n)<\infty\}$。

针对上述动态规划算法 DP2，有以下定理。

定理 4 用上述动态规划算法 DP2 求解问题 $1|p\text{-}batch, FB, p_j=a_j+bt \mid \sum f_j$ 的时间复杂性为 $O(n^4kN^2)$。

证明 首先 $z\in[0,nN]$,动态规划算法 DP2 有 n^2N 个状态。处理机带有 k 个不可用区间,批的开始加工时间 $S(t,a_{(l)})$可以在 $O(k)$时间内计算出,每一次递推中的 z^* 至多 nN 种选择,对于每一个 z^*,需要运行 $O(nk)$次判断是否有 $z^*\in\sigma$。因此,每一次递推需要运行 $O(n^2kN)$时间,所有的 $H(z,j)$可以在 $O(n^4kN^2)$时间内解出。

证毕

当 k 固定,目标函数为 $\sum U_j$,$N=1$ 时,有以下定理。

定理 5 当处理机的不可用区间个数固定时,排序问题 $1|p\text{-}batch, FB, p_j=a_j+bt \mid \sum U_j$ 在多项式时间内可解。

当目标函数为 $\sum w_jU_j$ 时,$N=\max w_j$。因此有以下定理。

定理 6 排序问题 $1|p\text{-}batch, FB, p_j=a_j+bt \mid \sum w_jU_j$ 在拟多项式时间内可解。

以下给出问题 $1|p\text{-}batch, FB, p_j=a_j+bt \mid \sum U_j$ 的一个例子。

例 $a_1=1,a_2=2,a_3=3,a_4=8,b=1,d_1=1,d_2=7,d_3=10,d_4=20$。$[h_{1,1},h_{1,2})=[2,3);[h_{2,1},h_{2,2})=[20,22);b=1$。计算过程如下。

初始条件:$H(0,0)=0,H(t,0)=\infty$。

$t=0$ 时,$j=1,H(0,1)=1$;

$i=1,t^*=0,S(H(0,0),a_1)=S(0,1)=0,f_1(1)=0$。

$j=2,H(0,2)=\infty$;

$i=1,t^*=0,S(H(0,0),a_2)=S(0,2)=0,f_1(2)=1,f_2(2)=0$;

$i=2,t^*=0,S(H(0,1),a_2)=S(1,2)=3,f_2(8)=1$。

$j=3,H(0,3)=\infty$;

$j=4,H(0,4)=\infty$。

$t=1$ 时,$j=1,H(1,1)=\infty$;

$j=2,H(1,2)=\min\{2,8\}=2,B_1=\{1,2\}$;

$i=1,t^*=0,S(H(0,0),a_2)=S(0,2)=0,f_1(2)=1,f_2(2)=0$;

$i=2,t^*=0,S(H(0,1),a_2)=S(1,2)=3,f_2(8)=1$。

$j=3,H(1,3)=9,B_1=\{1\},B_2=\{2,3\}$或者 $B_1=\{1,2\},B_2=\{3\}$;

$j=4,H(1,4)=\infty$。

$t=2$ 时,$j=1,H(2,1)=\infty$;

$j=2,H(2,2)=\infty$;

$j=3,H(2,3)=9,B_1=\{1,2,3\}$;

$j=4,H(2,4)=14,B_1=\{1\},B_2=\{2,3,4\}$或者 $B_1=\{1,2\},B_2=\{3,4\}$。

$t=3$ 时,$j=1,H(3,1)=\infty$;

$j=2,H(3,2)=\infty$;

$j=3,H(3,3)=\infty$;

$j=4,H(3,4)=\min\{14,52\},B_1=\{1,2,3,4\}$。

$t=4$ 时,$j=1,H(4,1)=\infty$;

$j=2,H(4,2)=\infty$;

$j=3,H(4,3)=\infty$;

$j=4,H(4,4)=\infty$。

综上:$\min\{t:H(t,n)<\infty\}=2$。

最优解有两种:$B_1=\{1\},B_2=\{2,3,4\}$,此时 J_2,J_3 误工;$B_1=\{1,2\},B_2=\{3,4\}$,此时 J_1,J_3 误工。

4 结论

在生活生产过程中,会遇到很多并行批排序问题,研究这类问题具有广泛的应用价值和实际意义。本文将处理机带有不可用区间与工件带有释放时间和退化效应模型 $p_j=a_j+bt$ 结合起来,研究目标函数分别为极小化最大费用和极小化总费用问题,这类并行批排序问题,是否有 FPTAS,有待进一步研究讨论。

参考文献:

[1] Lee C Y, Uzsoy R, Martin-vega L A. Efficient algorithms for scheduling semiconductor burn-in operations[J]. Operational Research, 1992, 40(4): 764-775.

[2] Brucker P, Gladky A, Hoogeveen H, et al. Scheduling batching machine[J]. Journal of Scheduling. 1998, 1(1): 31-54.

[3] Lee C Y, Uzsoy R. Minimizing makespan on a single batch processing machine with dynamic job arrivals[J]. International Journal of Production Research, 1999, 37(1): 219-236.

[4] Lu L F, Yuan J J. Unbounded parallel batch scheduling with job delivery to minimize makespan[J]. Operations Research Letters, 2008, 36(4): 477-480.

[5] Lee C Y. Machine scheduling with an availability constraint [J]. Journal of Global Optimization, 1996, 9(3/4): 395-416.

[6] Yuan J J, Qi X L, Lu L F, et al. Single machine unbounded parallel-batch scheduling with forbidden intervals[J]. European Journal of Operational Research, 2008, 186(3): 1212-1217.

[7] 张敏娇,罗成新. 带有拒绝工件和机器维修区间的单机排序问题[J]. 重庆师范大学学报:自然科学版, 2012, 29(6): 15-19.
Zhang M J, Luo C X. Single machine scheduling problem with rejection jobs and a fixed machine non-availability interval[J]. Journal of Chongqing Normal University: Natural Science, 2012, 29(6): 15-19.

[8] Li S S, Ng C T, Cheng T C E, et al. Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan[J]. European Journal of Operational Research, 2011, 210(3): 482-488.

[9] Qi X L, Zhou S G, Yuan J J. Single machine parallel-batch scheduling with deteriorating jobs[J]. Theoretical Computer Science, 2009, 410(8/9/10): 830-836.

[10] Ji M, He Y, Cheng T C E. Scheduling linear deteriorating jobs with an availability constraint on a single machine [J]. Theoretical Computer Science, 2006, 362(1/2/3): 115-126.

[11] Jafari A, Moslehi G. Scheduling linear deteriorating jobs to minimize the number of tardy jobs[J]. Journal of Global Optimization, 2012, 54(2): 389-404.

[12] 沈晓飞,赵玉芳,王晓丹. 带有退化效应和不可用区间的并行批排序问题[J]. 沈阳师范大学学报:自然科学版, 2014, 32(1): 39-43.
Shen X F, Zhao Y F, Wang X D. Parallel-batch scheduling problem with forbidden intervals and deteriorating jobs [J]. Journal of Shenyang Normal University: Natural Science, 2014, 32(1): 39-43.

Operations Research and Cybernetics

Parallel-batch Scheduling Problem with Forbidden Intervals and Deteriorating

SHEN Xiaofei, ZHAO Yufang, WANG Xiaodan

(School of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034, China)

Abstract: In this paper, we consider the single machine unbounded parallel-batch scheduling problem with linear deteriorating and forbidden intervals. In this model, the actual processing time of a job is an increasing linear function of its starting time. The processing time of a batch is equal to the largest processing time among all jobs in the batch. Jobs in the same batch start processing and finish simultaneously, and once processing of a batch is initiated, it cannot be interrupted. The completion time of all jobs in a batch is equal to the completion time of the batch. In this paper, we analyzed properties of the optimal solution; two pseudo-polynomial time optimal algorithms for minimizing the maximum cost and minimizing the total cost are given, respectively. Especially when k is fixed, we prove that the problem for minimizing the number of tardy jobs can be solved in polynomial time, and show the efficiency of the algorithm by a numeral example.

Key words: parallel-batch; forbidden intervals; deteriorating effect; pseudo-polynomial algorithm

(责任编辑 方 兴)