文章编号:1000 - 2367(2017)01 - 0008 - 07　　DOI:10.16366/j.cnki.1000 - 2367.2017.01.002

最大化接收工件总权值的批处理机在线排序

李文杰[1]，熊建栋[2]，翟红村[1]

(1. 洛阳师范学院 数学科学学院,河南 洛阳 471934; 2. 河南师范大学 数学与信息科学学院,河南 新乡 453007)

摘　要:研究 m 台无界批处理机上的在线排序问题. 每个工件 J_j 具有一个相同的加工时间 $p>0$,一个到达时间 $r_j \ge 0$,一个权值 $w_j>0$,一个必须交货期 $d_j>0$. 无界批处理机是指一台机器可以同时加工任意多个工件,目标是确定一个工件允许被中断重启的在线排序使得接收工件的总权值最大化. 主要设计了一个在线算法并证明其竞争比为 $3-\frac{1}{m}-(4m-2)(2m^2-m)^{1/2}/(2m^2-m)$.

关键词:在线排序; 竞争比; 批处理机; 接收工件总权值

中图分类号:O224　　**文献标志码:**A

批处理环境下的最大化接收工件总利益(具体指最大化按时完工工件的总个数、总权值或总加工时间)在线排序是近几年学者们持续研究的一类重要排序模型,该模型可以应用于网络通信中的客户服务系统[1-5].

m 台无界批处理机上的最大化接收工件(工件长度相同)总权值的在线排序是本文所要研究的问题. 设 I 是一个工件集合并且包含 n 个加工时间均为 $p>0$ 工件. 用 $M_1, M_2, \cdots, M_m$ 表示 m 台批处理机. 对每一个 I 中的工件 J_j, 它具有一个到达时间 $r_j \ge 0$,一个权值 w_j,一个必须交货期 $d_j>0$ 如果工件 J_j 在时刻点 t 满足 $t+p \le d_j$,就称工件 J_j 在时刻 t 是有效工件;反之,若 $t+p>d_j$ 就称工件 J_j 在时刻 t 已经过期. 用 b 表示批的容量,每台批处理机可以同时加工 b 个工件,本文主要研究批容量无界情形,即 $b=\infty$. 给定一个排序 σ,如果一个工件被 σ 按时完工,即该工件的完工时间小于等于其必须交货期,就称该工件被 σ 接收. 本文的目标就是对实例集 I 确定一个可中断重启的在线排序,最大化接收工件的总权值. 这里的"中断重启"是指一个正在被加工的工件可以在某个时刻被中断,一旦该工件被中断那么该工件加工过的部分将被忽略掉,并且在之后的时间里该工件还需被重新加工.

竞争比是衡量一个在线算法性能的主要依据. 称一个在线算法是 ρ- 竞争的($\rho \ge 0$), 定义 ρ 为 $\{O_{PT}(I)/A(I)\}$ 的上确界,其中 I 是满足 $O_{PT}(I)$ 大于0的任一实例集,$A(I)$ 表示算法 A 作用于 I 而产生的目标函数值,$O_{PT}(I)$ 表示最优排序关于 I 产生的目标值. 由 ρ 的定义可知,如果一个排序问题的在线算法的竞争比 ρ 值越接近于1,那么该算法的性能就越优良. 然而,大多数在线排序问题往往不存在竞争比等于1的在线算法.

有关最大化接收工件总权值的在线排序问题的研究成果很多,这里主要介绍一些与本文研究内容密切相关的一些结果. 文献[6]首先研究了单台无界批处理机模型,并给出了一个 $3+2\sqrt{2}$-竞争的在线算法. 紧接着,文献[7]对文献[6]中的算法进行了紧界分析,证明该算法的竞争比是5. 后来,文献[8]再次对上述排序问题进行研究并得到了一个4.56-竞争的改进算法. 文献[9]研究了更一般的情形:任意多台批处理机,多组不兼容工件,批容量无界和有界模型,给出了一个 $3+2\sqrt{2}$-竞争的在线算法. 此算法是解决该问题的第一个

收稿日期:2016-07-19;**修回日期:**2016-12-26.

基金项目:国家自然科学基金(11501279);河南省科技攻关计划项目(162102210265);河南省基础与前沿技术研究计划项目(162300410085).

作者简介(通信作者):李文杰(1981 -),男,河南正阳人,洛阳师范学院讲师,博士,主要从事在线排序研究, E-mail:liwenjie0521@163.com.

常数界竞争比的在线算法. 对 m 台无界批处理机模型,当 $m=3$ 时,文献[10]给出了一个在线算法 A 并证明其竞争比为 $(8+2\sqrt{15})/3$. 本文将该算法推广到 m 台机情形,并用更简明的方法证明其竞争比为 $3-\frac{1}{m}+\frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$. 易知 $(8+2\sqrt{15})/3$ 就是 $m=3$ 时的值,表明本文的结果覆盖了前者的结果.

1 在线算法 A_m^α 及其竞争比分析

本节,首先设计出问题的一个在线算法 A_m^α,该算法是文献[10]中算法 A 的推广. 其次对算法 A_m^α 的竞争比进行分析. 设 I 是一工件集合,用 $W(I)$ 表示 I 中所有工件的权和,并用 $|I|$ 表示 I 中的元素个数. 设 t 是当前时刻点. 令 $B^i(t)$ 表示在 t 时刻 M_i 上正在被加工的批,其中 $i=1,2,\cdots,m$. 如果 M_i 在 t 时刻是空闲的,就令 $B^i(t)=\phi$. 设 $Q^i(t)=\{J\in B^i(t):d_j\geq t+p\}$, $i=1,2,\cdots,m$. 则 $Q^i(t)$ 表示 $B^i(t)$ 中在时刻 t 仍然有效的工件所组成的集合. 用 $U(t)$ 表示在时刻 t 还没有被安排加工的所有有效工件组成的集合. 设 $U(t,i)=U(t)\cup Q^i(t)$. 令 $\alpha_m=1+\frac{\sqrt{2m^2-m}}{m}$. 称时刻点 t 是机器 M_i 的一个中断重启点,如果满足 $W(U(t,i))>\alpha_m W(B^i(t))$. 算法 A_m^α 执行步骤如下.

算法 A_m^α:

0^0 置 $t=0$.

1^0 如果 $U(t)=\phi$,则等到有新的工件到达,重置 t 为新工件的到达时间.

2^0 如果 $U(t)\neq\phi$,并且有机器出现空闲,则把 $U(t)$ 中的所有工件作为一批在 t 时刻安排在空闲机器上开工. 转到第 1^0 步.

3^0 如果 $U(t)\neq\phi$,并且所有机器都在忙碌,则执行以下策略.

(3.1) 如果 t 是机器 M_i 的一个中断重启点,其中 $1\leq i\leq m$ 则中断 $B^i(t)$ 并把 $U(t,i)$ 作为一批在 t 时刻安排在 M_i 上开工. 转到第 1^0 步.

(3.2) 如果 t 不是任何机器的中断重启点,则重置 $t:=\min\{r,d\}$ 其中 r 是在 t 之后新工件的到达时刻, d 是在 t 之后机器出现空闲的最早时刻. 转到第 1^0 步.

算法 A_m^α 的基本思想是:在每一个满足 $U(t)\neq\phi$ 的决策时刻点 t(可能为工件的到达时刻或机器的空闲时刻),如果有机器出现空闲,则将 $U(t)$ 中的工件构成一批并安排在空闲机器上加工;如果所有机器都在忙碌且有某台机器(比如第 i 台)出现中断重启点,那么此时就将该台机器上正在被加工的批中断,然后将 $U(t,i)$ 中的工件构成一批并安排在该台机器上加工;除上述两种情形之外的任何情形,算法不做任何决策. 下面以3台机器为例举一个简单的实例验证一下算法 A_m^α 执行情况. 共有5个长度均为1的工件,它们的详细信息如下:

$$r_1=0,\quad r_2=\frac{1}{5},\quad r_3=\frac{1}{5},\quad r_4=\frac{1}{3},\quad r_5=\frac{1}{2};$$
$$W_1=3,\quad W_2=1.1,\quad W_3=1.2,\quad W_4=2,\quad W_5=4.7;$$
$$D_1=1,\quad D_2=\frac{6}{5},\quad d_3=\frac{3}{2},\quad d_4=\frac{4}{3},\quad D_5=\frac{3}{2}.$$

注意到,当 $m=3$ 时, $\alpha_3=1+\sqrt{15}/3$. 易知 A_m^α 首先将工件 $\{J_1\}$, $\{J_2,J_3\}$ 和 $\{J_4\}$ 分别作为一批并安排在机器 M_1,M_2,M_3 上依次在时刻点 $0,\frac{1}{5},\frac{1}{3}$ 开始加工. 在时刻点 $\frac{1}{2}$,有新工件 J_5 到达并且满足 $U(\frac{1}{2},2)=\{J_3,J_5\}$. 又知,在机器 M_2 上正在被加工的工件的总权值为 $W(B^2(\frac{1}{2}))=2.3$,有效工件集合 $U(\frac{1}{2},2)$ 中的工件总权值为 $W(U(\frac{1}{2},2))=5.9$. 不难发现 $W(U(\frac{1}{2},2))>\alpha_3 W(B^2(\frac{1}{2}))$. 因此时刻点 $\frac{1}{2}$ 为机器 M_2 的一个中断重启点. 于是 A_m^α 首先中断在 M_2 上正在被加工的批 $\{J_2,J_3\}$,然后将 $\{J_3,J_5\}$ 作为新的一批工件在 $\frac{1}{2}$ 时刻安排

在 M_2 上开工.

一个正在机器上被加工的批,如果在没有完工之前被 A_m^α 中断,称之为中断批;如果该批被 A_m^α 完全加工,称之为完成批. 为了便于分析算法 A_m^α 的竞争比,下面将沿用文献[10]中的记号和证明思路. 设$(B_1^i,B_2^i,\cdots,B_k^i)$为由算法 A_m^α 关于工件实例 I 在第 i 台机器上产生的一个极大的连续批序列,其中前 $k-1$ 个连续批都是中断批,最后一个批 B_k^i 是完成批,$k\geq 1$. 称这样的批序列为由 A_m^α 生成的一个基本中断链. 因此,可知算法 A_m^α 关于 I 产生的排序一定是由一系列的基本中断链组成. 对每台机器 M_i,$1\leq i\leq m$,令 $P_1^i,P_2^i,\cdots,P_{s_i}^i$ 表示算法 A_m^α 在 M_i 上产生的所有基本中断链,其中 $P_h^i=(B_{h,1}^i,B_{h,2}^i,\cdots,B_{h,k_h^i}^i)$ 是 M_j 上的第 h 个基本中断链,$1\leq h\leq s_i$. 设 $\tilde{B}$ 表示由算法 A_m^α 产生的所有中断批和完成批构成的批集合,并记 $B=\{B_{h,k_h^i}^i:1\leq i\leq m,1\leq h\leq s_i\}$. 可得所有完成批构成的集合 B 包含在 $\tilde{B}$ 中,即 $B\subseteq\tilde{B}$ 且 $A_m^\alpha(I)=W(B)$. 对任一批 $B\in\tilde{B}$,定义其开工时间和完工时间分别为 $S(B)$ 和 $C(B)$. 令 C_0 和 C^* 分别表示 I 中第一个工件的到达时间和 I 中最后一批的完工时间. 则可知所有没有被 A_m^α 接收的工件将会在时刻点 C^*+p 处全部过期并且在时间区间$[C_0,C^*)$中的任何一个机器空闲时刻点都不会有还未被加工的有效工件存在.

设 π 是关于工件集合 I 的一个离线最优排序. 定义一个典型最优排序 π^* 其满足条件:① 在 π^* 中每个工件的开工时刻都是最晚的,即表明如果提前开工将会破坏 π^* 的最优性;② 在 π^* 中所产生的加工批(即被 π^* 接收的批)的个数最小. 于是有下面的两个引理,它们分别是文献[10]中的引理3.1和引理3.2的内容,其证明略.

引理 1 对一个基本中断链 $P=(B_1^i,B_2^i,\cdots,B_k^i)$,$k\geq 2$,$1\leq i\leq m$,可得 $\sum_{i\leq j\leq k}W(B_j^i)\leq \frac{\alpha_m}{\alpha_m-1}W(B_k^i)$.

引理 2 若 π^* 是一个关于实例 I 的典型最优排序,则以下两个结论成立:

(a) 没有工件在时间段$[C^*,\infty)$内被 π^* 接收;

(b) 对任一时刻点 $t\in[C_0,C^*)$,至多有一批工件在时刻点 t 被 π^* 开工.

考虑任一关于实例 I 的典型最优排序 π^*. 令 $O_1,O_2,\cdots,O_q$ 分别表示被 π^* 接收的批,它们的开工时间分别设为 $S^*(O_1),S^*(O_2),\cdots,S^*(O_q)$. 令 $O_0\neq\phi$ 且 $S^*(O_0)=C_0$. 由引理2,可得 $O_j\neq\phi$ 对所有的 $j=1,2,\cdots,q$ 且 $C_0=S^*(O_0)<S^*(O_2)<\cdots<S^*(O_q)<C^*$. 对每一个 $1\leq j\leq q$,定义集合 $B(O_j)=\{B\in\tilde{B}:S(B)\leq S^*(O_j)<C(B)\}$ 和 $B(O_{j-1},O_j)=\{B\in\tilde{B}:S^*(O_{j-1})<S(B)\leq S^*(O_j)\}$. $B(O_j)$ 表示在时刻点 $S^*(O_j)$ 正在被算法 A_m^α 加工的批组成的集合;$B(O_{j-1},O_j)$ 表示在时间段$(S^*(O_{j-1}),S^*(O_j)]$内由 A_m^α 产生的批组成的集合. 为方便起见,$S^*(O_j)$,$B(O_j)$ 和 $B(O_{j-1},O_j)$ 可以分别简写为 S_j^*,$B(j)$ 和 $B(j-1,j)$. 对 π^* 中的一批工件 O_j,$1\leq j\leq q$,如果满足 $B(j-1,j)=\phi$,就称 O_j 是一个畸形批;否则称 O_j 是一个正常批. 由正常批的定义可知 O_1 是一个正常批. 下面将文献[10]中的引理3.3和引理3.4的内容合在一起并证明当机器台数大于等于3时结论依然成立.

引理 3 对任一批 O_j,$1\leq j\leq q$,则以下两个结论成立:(i) 如果 O_j 是一个正常批,则有

$$W(O_j)\leq\sum_{B\in B(j-1,j)}W(B)+(\alpha_m-1)\min\{W(B):B\in B(j-1,j)\cap B(j)\};$$

(ii)如果 O_j 是一个畸形批,则有 $W(O_j)\leq\alpha_m\min\{W(B):B\in B(j-1,j)\}$.

证明 首先证明结论(ii). 由于 O_j 是一个畸形批,可得$j\geq 2$ 且 $B(j-1,j)=\phi$. 根据 π^* 是一个关于工件集合 I 的典型最优排序,对每一个工件 $J\in O_j$,可得其到达时间满足 $r(J)>S_{j-1}^*$. 于是可得 $r(J)>S_{j-1}^*\geq\max\{S(B):B\in B(j-1,j)\}$,由 $B(j)$ 的定义得 $r(J)\leq S_j^*<\min\{C(B):B\in B(j)\}$. 因此,得到 $O_j=U(S_j^*)$. 如果时刻点 S_j^* 是在算法 A_m^α 下某台机器的一个空闲时刻点,则有 $U(S_j^*)=\phi$. 于是得 $O_j=U(S_j^*)=\phi$,这显然与 $O_j\neq\phi$ 相矛盾. 因此,在算法 A_m^α 下的 S_j^* 时刻每台机器都在加工工件. 进而可得 $|B(j)|=m$. 设 $B(j)=\{B^1(S_j^*),B^2(S_j^*),\cdots,B^m(S_j^*)\}$,即表明 $B(j)$ 中刚好有 m 个加工批. 由算法 A_m^α 的执行策略可得,

$$W(O_j)=W(U(S_j^*))\leq\min\{W(U(S_j^*,1)),W(U(S_j^*,2)),\cdots,W(U(S_j^*,m))\}\leq$$

$$\alpha_m \min\{W(B^1(S_j^*)),W(B^2(S_j^*)),\cdots,W(B^m(S_j^*))\} = \alpha_m \min\{W(B):B\in B(j)\},$$

其中 $U(S_j^*,i) = U(S_j^*)\cup Q^i(S_j^*), i = 1,2,\cdots,m, Q^i(S_j^*)$ 是算法 A_m^α 下在 S_j^* 时刻加工批 $B^i(S_j^*)$ 中的所有有效工件组成的集合. 结论(ii)得证. 分以下两种情形证明结论(i).

情形 1　时刻点 S_j^* 是在算法 A_m^α 下某台机器的一个空闲时刻点. 则可得 $U(S_j^*) = \phi$. 因此可知, O_j 中的所有工件只能是批集合 $B(j-1,j)$ 中的工件. 于是可得

$$W(O_j)\le \sum_{B\in B(j-1,j)}W(B)\le \sum_{B\in B(j-1,j)}W(B) + (\alpha_m - 1)\min\{W(B):B\in B(j-1,j)\cap B(j)\}.$$

情形 2　在算法 A_m^α 下的 S_j^* 时刻每台机器都是忙碌的. 则可得 $|B(j)| = m$ 且 $B(j-1,j)\cap B(j) = \phi$. 设 $B^*(j) = B(j-1,j)\cap B(j) = \{B^{\sigma(1)}(S_j^*),B^{\sigma(2)}(S_j^*),\cdots,B^{\sigma(k)}(S_j^*)\}$, 其中 $B^{\sigma(l)}(S_j^*)$ 表示在时刻点 S_j^* 机器 $M_{\sigma(l)}$ 上正在被 A_m^α 加工的批, $1\le l\le k, 1\le \sigma(l)\le m$. 则有 $1\le k\le m$. 注意到, O_j 中的所有工件只能来自工件集合 $B(j-1,j)\cup U(S_j^*)$. 这里将 O_j 划分为 k 个不交的子集的并, 即令 $O_j = O^*\cup O^{\sigma(1)}\cup\cdots\cup O^{\sigma(k)}\cup O'$ 满足 $O^*\subseteq B(j-1,j)\backslash B^*(j), O^{\sigma(l)}\subseteq Q^{\sigma(l)}(S_j^*), l = 1,2,\cdots,k, O'\subseteq U(S_j^*)$. 则可得 $O^{\sigma(l)}\subseteq Q^{\sigma(l)}(S_j^*)\subseteq B^{\sigma(l)}(S_j^*)$ 且对所有的 $l = 1,2,\cdots,k, (O^{\sigma(l)}\cup O')\subseteq U(S_j^*,l)$. 于是断言下面的不等式成立:

$$\sum_{1\le l\le k}W(O^{\sigma(l)}) + W(O')\le \sum_{1\le l\le k}W(B^{\sigma(l)}(S_j^*)) + (\alpha_m - 1)\min\{W(B):B\in B^*(j)\}. \tag{1}$$

为方便起见, 不妨假设 $W(B^{\sigma(1)}(S_j^*))\ge W(B^{\sigma(2)}(S_j^*))\ge\cdots\ge W(B^{\sigma(k)}(S_j^*))$. 由算法 A_m^α 可知, 对所有的 $l = 1,2,\cdots,k$,

$$W(O^{\sigma(l)}) + W(O')\le W(U(S_j^*,l))\le \alpha_m W(B^{\sigma(l)}(S_j^*)). \tag{2}$$

因此, 要证明(1)式成立只需证明不等式

$$\sum_{1\le l\le k}W(O^{\sigma(l)}) + W(O')\le \sum_{1\le l\le k}W(B^{\sigma(l)}(S_j^*)) + \alpha_m W(B^{\sigma(l)}(S_j^*)) \tag{3}$$

成立. 由(2)式可得

$$\sum_{1\le l\le k}W(O^{\sigma(l)}) + W(O') = \sum_{1\le l\le k-1}W(O^{\sigma(l)}) + W(O^{\sigma(k)}) + W(O')\le \sum_{1\le l\le k-1}W(O^{\sigma(l)}) + \alpha_m W(B^{\sigma(k)}(S_j^*))\le \sum_{1\le l\le k-1}W(B^{\sigma(l)}(S_j^*)) + \alpha_m W(B^{\sigma(k)}(S_j^*)),$$

其中第一个不等式成立是因为(2)式, 第二个不等式成立是因为, 对所有的 $l = 1,2,\cdots,k-1$, 都有 $O^{\sigma(l)}\subseteq B^{\sigma(l)}(S_j^*)$. 因此(3)式得证, 进而表明(1)式得证.

由(1)式以及事实 $O^*\subseteq B(j-1,j)\backslash B^*(j)$, 可以得到

$$W(O_j) = W(O^*) + \sum_{1\le l\le k}W(O^{\sigma(l)}) + W(O')\le W(O^*) + \sum_{1\le l\le k}W(B^{\sigma(l)}(S_j^*)) + (\alpha_m - 1)\min\{W(B):B\in B^*(j)\}\le \sum_{B\in B(j-1,j)\backslash B^*(j)}W(B) + \sum_{1\le l\le k}W(B^{\sigma(l)}(S_j^*)) + (\alpha_m - 1)\min\{W(B):B\in B^*(j)\}\le \sum_{B\in B(j-1,j)}W(B) + (\alpha_m - 1)\min\{W(B):B\in B(j-1,j)\cap B(j)\}.$$

结论(i)得证. 综上引理 3 得证.

下面将分两种情形证明算法 A_m^α 的竞争比为 $\rho = 3 - \dfrac{1}{m} + \dfrac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$, 即证明 $O_{PT}(I)\le \rho A_m^\alpha(I)$.

情形 a　最优排序 π^* 中的所有批都是正常批. 对每一个 $1\le j\le q$, 由引理 3(i) 可得, $W(O_j)\le \sum_{B\in B(j-1,j)}W(B) + (\alpha_m - 1)\min\{W(B):B\in B(j-1,j)\cap B(j)\}$. 注意到, $(B(j-1,j)\cap B(j))\subseteq B(j-1,j)$, 则有 $W(O_j)\le \alpha_m\sum_{B\in B(j-1,j)}W(B)$. 因此可得 $O_{PT}(I) = \sum_{1\le j\le q}W(O_j)\le \alpha_m\sum_{1\le j\le q}\sum_{B\in B(j-1,j)}W(B)$. 由于 $B(j-1,j)\cap B(k-1,k) = \phi$, 当 $j\ne k$ 时, 又 $B(j-1,j)\subseteq \tilde{B}$, 可得 $\cup_{1\le j\le q}B(j-1,j)\subseteq \tilde{B}$. 于是可得

$$O_{PT}(I) = \sum_{1\le j\le q}W(O_j)\le \alpha_m\sum_{1\le j\le q}\sum_{B\in B(j-1,j)}W(B) =$$

$$\alpha_m \sum_{B \in \cup_{1\le j\le q} B(j-1,j)} W(B) \le \alpha_m \sum_{B\in \tilde{B}} W(B).$$

注意到，$\alpha_m = 1 + \frac{\sqrt{2m^2 - m}}{m}$，$A_m^\alpha(I) = W(B)$. 根据引理 1 以及 $\tilde{B}$ 的定义可以得到，

$$O_{PT}(I) \le \alpha_m \sum_{B\in\tilde{B}} W(B) = \alpha_m \sum_{1\le i\le m} \sum_{1\le h\le s_i} \sum_{1\le j\le k_h^i} W(B_{h,j}^i) \le$$
$$\frac{\alpha_m^2}{\alpha_m - 1} \sum_{1\le i\le m} \sum_{1\le h\le s_i} W(B_{h,k_h^i}^i) = \frac{\alpha_m^2}{\alpha_m - 1} A_m^\alpha(I) =$$
$$\left(2 + \frac{(3m-1)\sqrt{2m^2-m}}{2m^2-m}\right) A_m^\alpha(I) < \rho A_m^\alpha(I).$$

情形 b 最优排序 π^* 中存在畸形批. 令 $\pi^*(I,R)$ 和 $\pi^*(I,A)$ 分别表示 π^* 中的所有正常批和畸形批组成的集合. 不妨设 $\pi^*(I,R) = \{O_1^r, O_2^r, \cdots, O_e^r\}$ 和 $\pi^*(I,A) = \{O_1^a, O_2^a, \cdots, O_{e'}^a\}$，则有 $q = e + e'$. 根据 π^* 中正常批和畸形批的定义可得：$B(O_{j-1}^r, O_j^r) \cap B(O_{k-1}^r, O_k^r) = \phi$，当 $1 \le j \ne k \le e$ 时，其中 O_0^r 是一个虚拟批在 C_0 时刻完工；$|B(O_l^a)| = m$ 对每一个畸形批 O_l^a，$1 \le l \le e'$. 注意到，对所有的 $l = 1,2,\cdots,e'$，有 $B(O_l^a) \subseteq (\cup_{1\le j\le e} B(O_{j-1}^r, O_j^r))$. 因此，如果 π^* 中一畸形批 O_h^a，$1 \le h \le e'$，在机器 $M_{\sigma(h)}(1 \le \sigma_{(h)} \le m)$ 上被加工，那么在 A_m^α 中一定存在一批 $B^{\sigma(h)} \in B(O_l^a) \subseteq (\cup_{1\le j\le e} B(O_{j-1}^r, O_j^r))$ 在时刻点 $S^*(O_h^a)$ 正在被 A_m^α 安排在机器 $M_{\sigma(h)}$ 上加工，并且 $S(B^{\sigma(h)}) < S^*(O_h^a) < C(B^{\sigma(h)})$. 于是可知，对 $\pi^*(I,A)$ 中的任一畸形批而言，在 $\cup_{1\le j\le e} B(O_{j-1}^r, O_j^r)$ 中都存在一批与之对应. 称批 O^a 与批 B 对应，记作 $O^a \mapsto B$，如果满足条件：$O^a \in \pi^*(I,A)$，$B \in \cup_{1\le j\le e} B(O_{j-1}^r, O_j^r)$，$S(B) < S^*(O^a) < C(B)$，并且 O^a 与 B 在 π^* 和 A_m^α 中被安排在同一台机器上加工. 对每个 $j(1 \le j \le e)$，定义集合 $X_j = \{O^a \in \pi^*(I,A) : O^a \mapsto B, B \in B(O_{j-1}^r, O_j^r)\}$，则有 $0 \le |X_j| \le m-1$. 可得 π^* 中批的划分

$$\pi^*(I,R) \cup \pi^*(I,A) = (O_1^r \cup X_1) \cup (O_2^r \cup X_2) \cup \cdots \cup (O_e^r \cup X_3). \tag{4}$$

引理 4 对所有的 $j = 1,2,\cdots,e$，$W(O_j^r \cup X_j) \le \frac{m\alpha_m + m - 1}{m} \sum_{B\in B(O_{j-1}^r, O_j^r)} W(B)$.

证明 设 $X_j = \{O_{j1}^a, O_{j2}^a, \cdots, O_{jx}^a\}$，则 $0 \le x \le m-1$ 且 $O_{j1}^a, O_{j2}^a, \cdots, O_{jx}^a$ 分别被 π^* 安排在不同的机器上加工. 如果 $x = 0$，即 $X_j = \phi$，由引理 3(i) 以及 O_j^r 是一个正常批，可得，

$$W(O_j^r \cup X_j) = W(O_j^r) \le \sum_{B\in B(O_{j-1}^r, O_j^r)} W(B) + (\alpha_m - 1)\min\{W(B) : B \in B(O_{j-1}^r, O_j^r) \cap B(O_j^r)\}.$$

因此，$W(O_j^r \cup X_j) \le \alpha_m \sum_{B\in B(O_{j-1}^r, O_j^r)} W(B) < \frac{m\alpha_m + m - 1}{m} \sum_{B\in B(O_{j-1}^r, O_j^r)} W(B)$.

下面假设 $1 \le x \le m-1$，对每个 $1 \le u \le x$，$O_{ju}^a \mapsto B^u$. 则有 $B^u \in B(O_{j-1}^r, O_j^r)$，$u = 1,2,\cdots,x$. 根据 X_j 的定义可得，$S^*(O_j^r) < S^* O_{ju}^a$. 由于 O_j^r 是一个正常批，$O_{j1}^a, O_{j2}^a, \cdots, O_{jx}^a$ 都是畸形批，则存在一批 $B^0 \in B(O_{j-1}^r, O_j^r)$ 满足 $B^0 \ne B^u$ 且 $S(B^0) \le S^*(O_j^r) \le C(B^0)$. 首先证明

$$\{B^0, B^1, \cdots, B^x\} \subseteq (B(O_{j-1}^r, O_j^r) \cap B(O_j^r) \cap (\cap_{1\le u\le x} B(O_{ju}^a))). \tag{5}$$

注意到，对每一个 $B^u(1 \le u \le x)$，$S(B^u) \le S^*(O_j^r) < C(B^u)$. 由 $B(O_{j-1}^r, O_j^r)$ 和 $B(O_j^r)$ 的定义可知，$\{B^0, B^1, \cdots, B^x\} \subseteq (B(O_{j-1}^r, O_j^r) \cap B(O_j^r))$. 不妨假设存在一批工件 B^v，$v \ne u$，$1 \le u \le x$，使得 $B^v \notin B(O_{ju}^a)$. 如果 O_{jv}^a 在 O_{ju}^a 之前被 π^* 加工，由于 $O_{jv}^a \mapsto B^v$，$O_{ju}^a \mapsto B^u$，则可得 $S^*(O_{jv}^a) \le S(B^u)$ 与事实 $S(B^u) \le S^*(O_j^r) \le S^*(O_{jv}^a)$ 相矛盾. 若 O_{jv}^a 在 O_{ju}^a 之后被 π^* 加工，用上述分析同样得出矛盾. 因此，对每一个 B^u，$1 \le u \le x$，$B^u \in \cap_{1\le u\le x} B(O_{ju}^a)$. 综上，(5) 式得证.

由引理 3(i) 和 (ii) 可得，

$$W(O_j^r \cup X_j) = W(O_j^r) + W(X_j) \le \sum_{B\in B(O_{j-1}^r, O_j^r)} W(B) + (\alpha_m - 1)\min\{W(B) : B \in$$
$$B(O_{j-1}^r, O_j^r) \cap B(O_j^r)\} + \alpha_m \sum_{1\le u\le x} \min\{W(B) : B \in B(O_{ju}^a)\}.$$

记 $B(*) = \{B^0, B^1, \cdots, B^x\}$. 由 (5) 式可得，

$$\min\{W(B):B\in B(O_{j-1}^r,O_j^r)\cap B(O_j^r)\}\le\min\{W(B^0),W(B^1),\cdots,W(B^x)\}$$

和 $\min\{W(B):B\in B(O_{ju}^{\alpha})\}\le\min\{W(B^0),W(B^1),\cdots,W(B^x)\}$ 对所有的 $u=1,2,\cdots,x$. 于是可得

$$W(O_j^r\cup X_j)\le\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)}W(B)+(\alpha_m-1)\min\{W(B):B\in B(O_{j-1}^r,O_j^r)\cap B(O_j^r)\}+$$
$$\alpha_m\sum\nolimits_{1\le u\le x}\min\{W(B):B\in B(O_{ju}^{\alpha})\}\le\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)\setminus B(*)}W(B)+\sum\nolimits_{0\le v\le x}W(B^v)+$$
$$((x+1)\alpha_m-1)\min\{W(B^0),W(B^1),\cdots,W(B^x)\}\le\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)\setminus B(*)}W(B)+$$
$$\frac{(x+1)\alpha_m+x}{x+1}\sum\nolimits_{0\le v\le x}W(B^v)\le\frac{m\alpha_m+m-1}{m}\Big(\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)\setminus B(*)}W(B)+$$
$$\sum\nolimits_{0\le v\le x}W(B^v)\Big)=\frac{m\alpha_m+m-1}{m}\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)}W(B),$$

最后一个不等式成立是因为 $x+1\le m$. 引理 4 得证.

注意到, $\cup_{1\le j\le e}B(O_{j-1}^r,O_j^r)\subseteq\tilde{B}$. 由(4)式和引理 4 可得:

$$O_{PT}(I)=\sum\nolimits_{1\le j\le e}W(O_j^r\cup X_j)\le\frac{m\alpha_m+m-1}{m}\sum\nolimits_{1\le j\le e}\sum\nolimits_{B\in B(O_{j-1}^r,O_j^r)}W(B)=$$
$$\frac{m\alpha_m+m-1}{m}\sum\nolimits_{B\in\cup_{1\le j\le e}B(O_{j-1}^r,O_j^r)}W(B)\le\frac{m\alpha_m+m-1}{m}\sum\nolimits_{B\in\tilde{B}}W(B).$$

再由引理 1 以及事实 $\alpha_m=1+\frac{\sqrt{2m^2-m}}{m}$, $A_m^{\alpha}(I)=W(B)$, 可以得到:

$$O_{PT}(I)\le\frac{m\alpha_m+m-1}{m}\sum\nolimits_{B\in\tilde{B}}W(B)=\frac{m\alpha_m+m-1}{m}\sum\nolimits_{1\le i\le m}\sum\nolimits_{1\le h\le s_i}\sum\nolimits_{1\le j\le k_h^i}W(B_{h,j}^i)\le$$
$$\frac{m\alpha_m^2+(m-1)\alpha_m}{m(\alpha_m-1)}\sum\nolimits_{1\le i\le m}\sum\nolimits_{1\le h\le s_i}W(B_{h,k_h^i}^i)=\Big(3-\frac{1}{m}+$$
$$\frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}\Big)A_m^{\alpha}(I)=\rho A_m^{\alpha}(I).$$

综合上述分析,得到如下本文的主要结论.

定理 1　算法 A_m^{α} 的竞争比为 $\rho=3-\frac{1}{m}+\frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$,其中 m 表示批处理机器的台数.

下面构造一个证明算法 A_m^{α} 的 $3-\frac{1}{m}+\frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$ 紧界的实例:实例 I 共包含 $2m(n+1)$ 个等长工件. 置 $p=n^n$. 令 $\Gamma=\{J_j^{(i)}\in I:0\le j\le n,1\le i\le m\}$ 表示 I 中 $m(n+1)$ 个工件,满足 $w(J_j^{(i)})=(1+\frac{\sqrt{2m^2-m}}{m})^j$, $r(J_j^{(i)})=jp-mj+i-1$, $d(J_j^{(i)})=(j+1)p+m$. I 中另外 $m(n+1)$ 个工件用集合 $\Gamma'=\{J_j^{(i)}\in I:0\le j\le n,m+1\le i\le 2m\}$ 表示,且满足 $r(J_j^{(m+1)})=jp+m$, $w(J_j^{(m+1)})=\frac{\sqrt{2m^2-m}}{m}(1+\frac{\sqrt{2m^2-m}}{m})^j-1$, $d(J_j^{(m+1)})=d(J_j^{(1)})=\cdots=d(J_j^{(m)})=(j+1)p+m=r(J_j^{(m+1)})+p$, $w(J_j^{(k)})=(1+\frac{\sqrt{2m^2-m}}{m})^{j+1}-1$, $r(J_j^{(k)})=jp+k-1$, $d(J_j^{(k)})=r(J^{(k)j})+p$ 对所有的 $k=m+2,\cdots,2m$. 关于实例 I,算法最终只会将工件 $J_n^{(1)},J_n^{(2)},\cdots,J_n^{(m)}$ 分别作为一批安排在 m 机器上完全加工. 然而最优排序 O_{PT} 可以接受实例 I 中的所有工件. 具体排序如下:首先将工件 $J_0^{(i)},J_1^{(i)},\cdots,J_n^{(i)}$ 分别作为一批安排在机器 M_{i-m-1} 上,并依次在时刻点 $i-1,p+i-1,2p+i-1,\cdots,np+i-1$ 时开始加工;然后将 $\{J_0^{(1)},J_0^{(2)},\cdots,J_0^{(m+1)}\}$, $\{J_1^{(1)},J_1^{(2)},\cdots,J_1^{(m+1)}\}$, $\cdots$, $\{J_n^{(1)},J_n^{(2)},\cdots,J_n^{(m+1)}\}$ 分别作为一批安排在机器 M_m 上,并依次在时刻点 $m,p+m,2p+m,\cdots,np+m$ 时开工. 因此可以得到 $A_m^{\alpha}(I)=w(J_n^{(1)})+w(J_n^{(2)})+\cdots+w(J_n^{(m)})=m(1+\frac{\sqrt{2m^2-m}}{m})^n$, 和

$$O_{PT}(I) \leq \sum_{1\leq i\leq 2m} \sum_{1\leq j\leq n} w(J_j^{(i)}) = (m(1+\frac{\sqrt{2m^2-m}}{m})^{n+2} + (m-1)(1+\frac{\sqrt{2m^2-m}}{m})^{n+1} - 2m+1-(n+2)\sqrt{2m^2-m})/(\frac{\sqrt{2m^2-m}}{m}).$$

于是可得,当 $n \to \infty$ 时,$O_{PT}(I)/A_m^{\alpha}(I) \to 3 - \frac{1}{m} + \frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$.

2 结论与展望

本文讨论了 m 台批处理机上的等长工件的在线排序问题,目标函数是最大化接收工件的总权值. 得到了一个参数界竞争比的在线算法覆盖了前人的结果. 当 m 逐渐增大时, 不难发现 $3 - \frac{1}{m} + \frac{(4m-2)\sqrt{2m^2-m}}{2m^2-m}$ 的值会逐渐接近 $3+2\sqrt{2}$. 表明对机器台数非常大时算法的性能不太理想,并且本文主要研究的是批容量无界情形,对批容量有界模型以及多组不兼容工件情形有待进一步研究.

参 考 文 献

[1] CONINCK E D, VERBELEN T, VANKEIRSBILCK B. Dynamic auto-scaling and scheduling of deadline constrained workloads on IaaS clouds [J]. Journal of Systems and Software,2016,108:101-114.

[2] BARSHAN M, MOENS H. Deadline-aware advance reservation scheduling algorithms for media production networks[J]. Computer Communications,2016,77:26-40.

[3] POTTS C N, KOVALYOV M Y. Scheduling parallel Kalman filters with quantized deadlines[J]. Systems & Control Letters,2015,86:9-15.

[4] CHWA H S, BACK H, LEE J. Capturing urgency and parallelism using quasi-deadlines for real-time multiprocessor scheduling[J]. Journal of Systems and Software,2015,101:15-29.

[5] LI W J, YUAN J J. An Improved Online Algorithm for the Online Preemptive Scheduling of Equal-Length Intervals on a Single Machine with Lookahead [J]. Asia-Pacific Journal of Operational Research,2015,32:1-9.

[6] KIM J, CHWA K. Scheduling broadcasts with deadlines[J]. Theoretical Computer Science,2004,325:479-488.

[7] CHAN W, LAM T, TING H, et al. New results on on-demand broadcasting with deadline via job scheduling with cancellation[C]. Lecture Notes in Computer Science:Computing and Combinatorics,2004:210-218.

[8] FUNG S, ZHENG F, CHAN W,et al. Improved online broadcast scheduling with deadlines[J]. Journal of Scheduling,2008,11:299-308.

[9] LI W J, ZHANG Z K, LIU H L, et al. Online scheduling of equal-length jobs with incompatible families on multiple batch machines to maximize the weighted number of early jobs[J]. Information Processing Letters,2012,112:503-508.

[10] LI W J, YUAN J J. Improved online algorithms for the batch scheduling of equal-length jobs with incompatible families to maximize the weighted number of early jobs[J]. Optimization Letters,2014,8:1691-1706.

Online Batch-machine Scheduling to Maximize Total Weight of the Accepted Jobs

Li Wenjie[1], Xiong Jiandong[2], Zhai Hongcun[1]

(1. School of Mathematical Sciences, Luoyang Normal University, Luoyang 471934, China;
2. College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, China)

Abstract: This paper studies the online scheduling on m unbounded batch machines. Each job J_j has a common processing time $p>0$, an arriving time $r_j \geq 0$, a weight $w_j > 0$, a deadline $d_j > 0$. "Unbounded batch machines" means that one machine can process up to countless jobs simultaneously as a batch. The goal is to determine a preemption-restart schedule which maximizes total weight of the accepted jobs. In this paper, we first design an online algorithm, and then prove that it has a competitive ratio of $3 - \frac{1}{m} - (4m-2)(2m^2-m)^{1/2}/(2m^2-m)$.

Keywords: online scheduling; competitive ratio; batch machines; total weight of the accepted jobs