

# 极小化最大完工时间的 单机连续型批调度问题<sup>1)</sup>

赵玉芳<sup>1,2</sup> 唐立新<sup>1</sup>

<sup>1</sup>(东北大学物流优化与控制研究所 沈阳 110004)

<sup>2</sup>(沈阳师范大学数学与系统科学学院 沈阳 110034)

(E-mail: yufangzhao66@163.com, qhjytlx@mail.neu.edu.cn)

**摘 要** 从钢铁工业中加热炉对管坯的加热过程, 提出一种新的连续型批处理机调度问题, 与传统批处理机调度问题的批进批出方式不同, 其主要特征为批中工件的进入、处理和离开都连续进行, 批  $B_i$  的处理时间与该批的大小  $|B_i|$ 、批中工件  $T_j$  的处理时间  $p_j$  及机器的容量  $C$  都有关, 表示为  $p^{(i)} = \max_{T_j \in B_i} \{p_j\} (1 + \frac{|B_i| - 1}{C})$ . 对于极小化最大完工时间问题, 给出了一个复杂性为  $O(n^2)$  的动态规划算法, 并证明了这个算法的最优性.

**关键词** 钢铁, 加热炉调度, 连续批, 动态规划算法  
**中图分类号** O223

## Scheduling a Single Continuous Batch Processing Machine to Minimize Makespan

ZHAO Yu-Fang<sup>1,2</sup> TANG Li-Xin<sup>1</sup>

<sup>1</sup>(The Logistics Institute, Northeastern University, Shenyang 110004)

<sup>2</sup>(School of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034)

(E-mail: yufangzhao66@163.com, qhjytlx@mail.neu.edu.cn)

**Abstract** A new problem of continuous batch scheduling arisen from the heating-process of blooms in the iron and steel industry is addressed, which differs from the traditional batching machine scheduling where jobs in the same batch have a starting time and a finishing time. Each heating furnace is modeled as a continuous batch processing machine, which can handle up to  $C$  jobs simultaneously, where jobs in the same batch enter and leave the machine continuously. The processing time of a batch  $B_i$  in this paper is related to  $|B_i|$ ,  $p_j$  and  $C$ , where  $|B_i|$  is the size of  $B_i$  and  $p_j$  is the processing time of job  $T_j$  in the batch. We give the expression of the processing time of batch  $B_i$  as  $p^{(i)} = \max_{T_j \in B_i} \{p_j\} (1 + \frac{|B_i| - 1}{C})$ . We also present a dynamic programming algorithm with complexity of  $O(n^2)$  and the properties of the optimal solution to minimize makespan.

**Key words** Iron and steel industry, heating-furnace scheduling, continuous batch, dynamic programming algorithm

1) 国家自然科学基金项目 (60274049, 60521003), 国家杰出青年科学基金 (70425003), 高等学校优秀青年教师教学科研奖励计划 (教育司 [2002]383) 资助

Supported by National Natural Science Foundation of P. R. China (60274049, 60521003), the Excellent Young Faculty Program of the Ministry of P. R. China (70425003), and the Science Foundation for Excellent Youth Teachers of Universities by the Ministry of P. R. China ([2002]383)

收稿日期 2004-7-8 收修改稿日期 2006-5-29

Received July 8, 2004; in revised form May 29, 2006

## 1 问题的描述

本文提出一种新的批调度问题, 其主要特征为工件以批方式进行处理, 且同一批中的工件依次匀速连续地进入机器, 同时被处理完的工件也依次连续地离开机器, 批处理机在同一时刻可同时处理多个工件, 其最大数称为批处理机的容量. 同一批工件中每个工件的处理时间均等于这批工件中处理时间的最大者, 批的大小为这批工件的个数, 而批的处理时间是从该批中的第一个工件进入机器 (开始处理), 到最后一个工件 (处理完) 离开机器所用的时间, 因此批的处理时间与批的大小、工件的处理时间及机器的容量都有关. 本文把这种批调度问题称为连续型批处理机调度问题 (Continuous batch scheduling problems), 简称为连续型批调度问题. 具体描述如下:

设有  $n$  个工件  $T_j (j = 1, 2, \dots, n)$ , 在一台批处理机上处理, 处理时间为  $p_j (j = 1, 2, \dots, n)$ . 这些工件可以分成若干个批, 记为  $B_1, B_2, \dots, B_m$ . 批处理机的容量为  $C$ , 批  $B_i$  的大小为  $|B_i|$ , 在处理过程中, 同一批中工件的处理时间记为  $p^{(i)}$ , 即  $p^{(i)} = \max_{T_j \in B_i} \{p_j\}$ ; 批的处理时间记为  $p^{(i)}$ .

本文研究的批调度问题与以往研究的批调度问题不同之处在于: 以往研究的批调度问题又称为同时处理调度问题, 一般都是批进批出<sup>[1]</sup>. 主要分为两类: 第一类是并型工件同时处理的调度问题 (Parallel batching problems), 最早出现在半导体生产过程中, 每批的处理时间是这批工件中所有工件处理时间的最大者. Ikura 和 Gimple<sup>[2]</sup> 最早发表这类批调度问题的成果. 文献 [3~9] 也研究过这类调度问题, 并取得了丰富的结果. 另一类是串型工件同时处理调度问题 (Serial batching problems), 每批的处理时间是这批工件中所有工件处理时间之和. 文献 [10~13] 等都研究过这类调度问题, Brucker 等<sup>[14]</sup> 写过关于同时处理调度问题的综述. 本文研究的批调度问题也是同时处理调度问题, 但工件是按连续进、连续出的形式进行处理的, 批处理时间与上面两种批处理时间都不相同. 这在目前还没有人研究过.

本文研究的问题是极小化最大完工时间问题. 为了与并型工件同时处理调度问题及串型工件同时处理调度问题区分开来, 用三参数表示法记为:  $1|c\text{-batch}, C|C_{\max}$ .

本文第 2 节介绍了问题的实际背景; 第 3 节给出了批的处理时间  $p^{(i)}$  的计算公式及  $1|c\text{-batch}, C|C_{\max}$  最优解的性质; 在第 4 节给出了一个复杂性为  $O(n^2)$  的动态规划算法, 并证明了这个算法的最优性; 在第 5 节用数值例子进一步说明了此算法的计算过程.

## 2 问题的背景

本文讨论问题的实际背景是钢铁工业中的加热炉. 下面以环形加热炉为例, 简单介绍一下其工作特点.

环形加热炉一般有 3 到 5 个加热区、一个预热区和一个清理段, 如图 1.

每根管坯从清理段进入加热炉, 经过预热区、各个加热区加热后再回到清理段, 然后从清理段出来离开加热炉, 进行下一道工序. 环形加热炉可同时处理管坯的个数称为机器的容量, 记为  $C$ . 这最多  $C$  根管坯被“平铺”在加热炉的底部一起加热, 加热到一定的温度后一起匀速地向前移动, 一根管坯从清理段出来, 同时又一根管坯进入加热炉. 一直到这批工件的最后一根管坯进入加热炉被加热完离开机器, 表示这一批工件处理结束. 随着其余管坯进入加热炉加热, 新一批工件开始处理, 工作过程与上述一致.

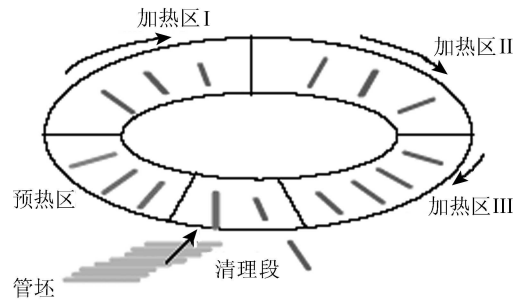


图1 环形加热炉的构造和工作特点

Fig. 1 Primary operations of rotary heating furnace

根据处理新批时工件进入处理机的初始方式不同,这类问题分为两种类型.一种是处理新批时工件先将炉底铺满,然后匀速地出一个工件、进一个工件;另一种类型是处理新批时并不铺满炉底,而是工件匀速地进入加热炉,在达到加热炉的容量后,再匀速地出一个工件、进一个工件.当没有工件进入加热炉时,炉内的工件还是按原来的速度进行加热直到结束.本文研究后一种形式,即批中的工件一直匀速地进入加热炉、加热和离开机器.

管坯的加热制度是由钢级根据热处理的原则而定的,钢级不同的管坯需要加热的温度不一样.我们把连续处理的管坯看成一批.则该批的大小为这批工件的个数.由于该机器的性质是连续进、连续出,所以批的容量为无限.

由于管坯在穿孔时温度必须达到一定的高度,所以把这批工件中要求达到的最高的温度作为该批中每个工件的温度.在这里假设温度越高处理的时间越长,即工件处理的时间与温度成正比.那么同一批中工件的处理时间是相同的,都等于这批中处理时间最长的工件的处理时间.批的处理时间为从该批中的第一个工件进入机器,到最后一个工件从机器中离开所用的时间.而这类问题不仅仅出现在环形加热炉的工作过程中,在钢铁工业的其他工作过程中也普遍存在,因此研究这种进入、处理和离开都连续进行、批的处理时间与批的大小、工件的处理时间及机器的容量等都有关系的连续型批调度问题具有广泛的应用价值和现实意义.

### 3 问题的批处理时间定义及最优解性质

在给出批的处理时间  $p^{(i)}$  的定义之前,先来看下面的例子.

**例 1.** 考虑 11 个工件、 $C = 4$  的一个例子,其中各工件的处理时间为:

$$p = (8, 8, 7, 6, 6, 5, 5, 4, 3, 2, 1)$$

若对工件分批为:  $B_1 = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7\}$ ,  $B_2 = \{T_8, T_9, T_{10}, T_{11}\}$ , 则批  $B_1$  中每个工件的处理时间都为 8; 批  $B_2$  中每个工件的处理时间都为 4. 处理过程如图 2. 其中设工件从零时刻开始处理. 工件按开始处理时间的先后依次进入、离开处理机.

从图 2 中可以看出,在任何时刻处理机最多可同时处理 4 个工件,只有当一批工件处理完后才可以处理下一批工件,并且第一批与第二批的处理时间分别为

$$p^{(1)} = \max\{8, 8, 7, 6, 6, 5, 5\}(1 + \frac{7-1}{4}) = 20, \quad p^{(2)} = \max\{4, 3, 2, 1\}(1 + \frac{4-1}{4}) = 7$$

注意到批  $B_i$  中每个工件的实际处理时间为  $p_{(i)}$ , 批处理机的容量为  $C$ , 并且工件是匀速地进入、离开处理机,连续地进行处理. 因此每增加一个工件,处理时间就要增加  $\frac{1}{C}p_{(i)}$ .

批  $B_i$  中工件的个数为  $|B_i|$  个, 除第一个工件需要  $p_{(i)}$  处理时间外, 还需要增加  $\frac{|B_i|-1}{C}p_{(i)}$  处理时间. 因此每批的处理时间  $p^{(i)}$  为

$$p^{(i)} = p_{(i)}\left(1 + \frac{|B_i|-1}{C}\right) = \max_{T_j \in B_i} \{p_j\}\left(1 + \frac{|B_i|-1}{C}\right), \quad i = 1, 2, \dots, m \quad (1)$$

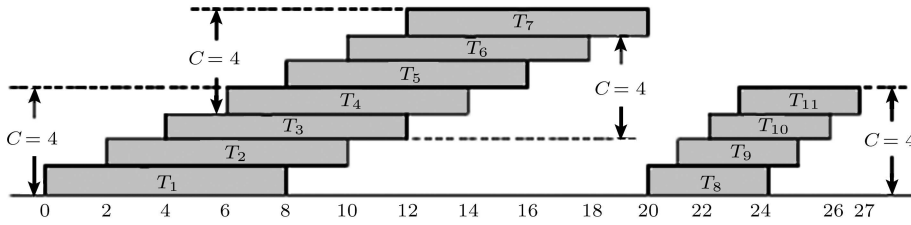


图 2 例 1 的调度图

Fig. 2 Gantt Charts of Example 1

下面讨论问题  $1|c\text{-batch}, C|C_{\max}$  的性质. 先给出下面的定义. 这里把工件按  $LPT$  编号, 即  $p_1 \geq p_2 \geq \dots \geq p_n$ .

**定义 1.** 若调度中的任意两批  $P$  和  $Q$ , 批  $P$  在批  $Q$  前处理, 意味着不存在这样的工件  $T_i, T_j$ , 使得  $T_i \in P, T_j \in Q$ , 且  $p_i < p_j$ , 则称此分批为批  $-LPT$  序的.

**定义 2.** 在批调度  $(B_1, B_2, \dots, B_r)$  中, 若第  $l$  批含有  $j_{l+1} - j_l$  ( $l = 1, 2, \dots, r$ ) 个工件, 批  $B_l = \{T_{j_l}, T_{j_l+1}, \dots, T_{j_{l+1}-1}\}$  且批内顺序也按下标从小到大排, 则称此调度为  $LPT$  序的.

**性质 1.** 在问题  $1|c\text{-batch}, C|C_{\max}$  的最优调度中, 处理时间相同的工件一定在同一批中.

**证明.** 用反证法. 设在最优调度  $\pi^*$  中, 目标函数值为  $C_{\max}(\pi^*)$ , 工件  $T_i, T_j$  有相同的处理时间  $p$ , 分别在批的处理时间为  $p^P, p^Q$  的两个批  $P, Q$  中. 设  $p_P = \max_{T_j \in P} \{p_j\}, p_Q = \max_{T_j \in Q} \{p_j\}$ ,

不失一般性, 设  $p_P < p_Q$ , 则  $p^P = p_P\left(1 + \frac{|P|-1}{C}\right), p^Q = p_Q\left(1 + \frac{|Q|-1}{C}\right)$ . 现将  $T_j$  从  $Q$  中

移入  $P$  中, 其余工件不变. 令  $P$  变为  $P', Q$  变为  $Q'$ , 其余批不变, 形成新的调度  $\pi'$ , 目标函数值为  $C_{\max}(\pi')$ . 由于  $T_i \in P, p_i = p_j = p$ , 所以  $p \leq p_P$ , 即  $p_{P'} = \max_{T_j \in P'} \{p_j\} = p_P$ ,

$p_{Q'} = \max_{T_j \in Q'} \{p_j\} = p_Q$ . 而  $|P'| = |P| + 1, |Q'| = |Q| - 1, p^{P'} = p_{P'}\left(1 + \frac{|P'|}{C}\right) = p_P\left(1 + \frac{|P|}{C}\right),$

$p^{Q'} = p_Q\left(1 + \frac{|Q|-2}{C}\right)$ , 所以

$$p^{P'} + p^{Q'} = p_P\left(1 + \frac{|P|}{C}\right) + p_Q\left(1 + \frac{|Q|-2}{C}\right) = p^P + p^Q + \frac{p_P - p_Q}{C}$$

而  $p_P < p_Q$ , 则  $p^{P'} + p^{Q'} < p^P + p^Q, C_{\max}(\pi') < C_{\max}(\pi^*)$ . 与  $\pi^*$  是最优调度矛盾.  $\square$

**性质 2.** 在问题  $1|c\text{-batch}, C|C_{\max}$  中, 一定存在这样的最优调度, 其同一批的工件是处理时间相邻的, 即若  $p_{j_1} \geq p_{j_2} \geq \dots \geq p_{j_k}$ , 其中  $k \leq n$ , 而  $T_{j_1}, T_{j_2}, \dots, T_{j_{i-1}}, T_{j_{i+1}}, \dots, T_{j_k}$  在同一批中, 则  $T_{j_i}$  必在此批中处理.

**证明.** 用反证法. 设  $\pi^*$  为最优调度, 目标函数值为  $C_{\max}(\pi^*)$ , 批  $P$  包含工件  $T_{j_1}, T_{j_2}, \dots, T_{j_{i-1}}, T_{j_{i+1}}, \dots, T_{j_k}$ , 而  $T_{j_i}$  在批  $Q$  中处理, 批  $P, Q$  的处理时间分别为  $p^P, p^Q$ , 设  $p_P = \max_{T_j \in P} \{p_j\}, p_Q = \max_{T_j \in Q} \{p_j\}$ . 由引理 1 可知  $p_P \neq p_Q$ , 则  $p_P$  与  $p_Q$  只能是下列情况之一:

1)  $p_P < p_Q$

2)  $p_P > p_Q$

我们首先来看 1):  $p_P < p_Q$ ,  $p^P = p_P(1 + \frac{|P|-1}{C})$ ,  $p^Q = p_Q(1 + \frac{|Q|-1}{C})$ . 现将  $T_{j_i}$  从  $Q$  中移入  $P$  中, 其余工件不变.  $P$  变为  $P'$ ,  $Q$  变为  $Q'$ , 其余批不变, 形成新的调度  $\pi'$ , 目标函数值为  $C_{\max}(\pi')$ . 由于  $p_{j_{i-1}} \geq p_{j_i}$ ,  $T_{j_{i-1}} \in P$ , 则  $P'$  中每个工件的处理时间不变. 由已知可得,  $p_{P'} = \max_{T_j \in P'} \{p_j\} = p_P$ ,  $p_{Q'} = \max_{T_j \in Q'} \{p_j\} = p_Q$ ,  $|P'| = |P| + 1$ ,  $|Q'| = |Q| - 1$ , 所以  $p^{P'} = p_P(1 + \frac{|P|}{C})$ ,  $p^{Q'} = p_Q(1 + \frac{|Q|-2}{C})$ , 因为  $p_P < p_Q$ , 则  $p^{P'} + p^{Q'} < p^P + p^Q$ ,  $C_{\max}(\pi') < C_{\max}(\pi^*)$ . 与  $\pi^*$  是最优调度矛盾.

2): 若  $p_P > p_Q$ , 将  $T_{j_{i+1}}, \dots, T_{j_k}$  从  $P$  中移入  $Q$  中, 其余工件不变,  $Q$  变为  $Q''$ ,  $P$  变为  $P''$ , 其余批不变, 形成新的调度  $\pi''$ , 目标函数值为  $C_{\max}(\pi'')$ . 由于  $p_{j_i} \geq p_{j_{i+1}} \geq \dots \geq p_{j_m}$ , 则  $p_{P''} = \max_{T_j \in P''} \{p_j\} = p_P$ ,  $p_{Q''} = \max_{T_j \in Q''} \{p_j\} = p_Q$ ,  $|P''| = |P| - (k - i)$ ,  $|Q''| = |Q| + k - i$ ,  $p^{P''} = p_P(1 + \frac{|P| - (k - i) - 1}{C})$ ,  $p^{Q''} = p_Q(1 + \frac{|Q| + k - i - 1}{C})$ , 因为  $p_P > p_Q$ , 则  $p^{P''} + p^{Q''} = p_P(1 + \frac{|P| - (k - i) - 1}{C}) + p_Q(1 + \frac{|Q| + k - i - 1}{C}) < p^P + p^Q$ ,  $C_{\max}(\pi'') < C_{\max}(\pi^*)$ . 与  $\pi^*$  是最优调度矛盾.  $\square$

性质 2 说明一定存在这样的最优分批, 其为批  $-LPT$  序的.

**性质 3.** 在问题  $1|c\text{-batch}, C|C_{\max}$  中, 必定存在一个最优调度  $(B_1, B_2, \dots, B_r)$  是  $LPT$  序的.

**证明.** 在任意一个最优调度中, 由于单机调度极小化最大完工时间问题与工件的处理顺序无关, 不妨设各批间按批  $LPT$  序的排列处理; 在各批的内部, 实际处理时间都相同, 不妨按下标从大到小排列, 其目标函数值不变. 这个最优调度就是  $LPT$  序的.  $\square$

性质 3 说明, 对于求问题  $1|c\text{-batch}, C|C_{\max}$  的最优调度, 可以首先把工件按  $LPT$  编号, 然后对这个  $LPT$  序分批即可.

## 4 动态规划算法

由问题  $1|c\text{-batch}, C|C_{\max}$  的性质, 我们可以得到下面的求解这个问题的动态规划算法. 工件按  $LPT$  序编号, 即  $T_1, T_2, \dots, T_n$ , 使得  $p_1 \geq p_2 \geq \dots \geq p_n$  后, 这个算法实际上求出了从第 1 个工件  $T_1$  到任一工件的最优分批.

如下的事实是经常用到的, 如果  $B_1, B_2, \dots, B_l$  (其中批  $B_i$  的最后一个工件为  $T_{j_i}$ ,  $1 \leq j_1 < j_2 < \dots < j_l = k$ ,  $i = 1, 2, \dots, l$ ) 是工件从  $T_1$  到  $T_k$  对应的最优分批, 那么工件从  $T_1$  到  $T_{j_i}$  按  $B_1, B_2, \dots, B_i$  分批最优. 事实上如果这个结论不成立, 设  $B'_1, B'_2, \dots, B'_p$  是工件从  $T_1$  到  $T_{j_i}$  的最优分批, 令工件从  $T_1$  到  $T_{j_i}$  按  $B'_1, B'_2, \dots, B'_p$  分批, 从  $T_{j_{i+1}}$  到  $T_k$  按  $B_{j_{i+1}}, \dots, B_l$  分批. 则后一种分批方法对应的目标函数值就比前一种分批方法对应的目标函数值要小, 这与  $B_1, B_2, \dots, B_l$  是工件从  $T_1$  到  $T_k$  对应的最优分批矛盾.

在下述动态规划算法中,  $f(k)$  表示工件从  $T_1$  到  $T_k$  对应的目标函数值,  $B_k$  是按  $f(k)$  分批的最后一批.  $\operatorname{argmin}\{\varphi(i)\}$  表示使  $\varphi(i)$  取最小值时对应的  $i$  值.

动态规划算法 (DP):

**步骤 1.** 把工件按  $LPT$  序编号, 使  $p_1 \geq p_2 \geq \dots \geq p_n$ ;

**步骤 2.** 令  $f(0) = 0$ ,  $B_0 = \Phi$ ,  $r_0 = 0$ ,  $k = 1$ ;

**步骤 3.** 按递归方程

$$f(k) = \min_{r_{k-1} \leq i \leq k-1} \{f(i) + p_{i+1,k}\} \quad (2)$$

计算  $f(k), k = 1, 2, \dots, n$ . 其中  $p_{i+1,k} = p_{i+1}(1 + \frac{k-i-1}{C})$ ; 令  $r_k = \arg \min_{r_{k-1} \leq i \leq k-1} \{f(i) + p_{i+1,k}\}$ ,  $B_k = \{T_{r_k+1}, \dots, T_k\}$ ;

**步骤 4.** 任意安排各批的次序.

求出  $f(n)$  后, 利用反向追踪就得到了所有工件的一个最优分批,  $f(n)$  就是最优值.

下面证明算法的最优性.

**定理 4.** 基于 *LPT* 的动态规划算法得到原调度问题的最优解.

**证明.** 由性质 3, 只要证明对 *LPT* 序分批后的工件进行分批就可. 下面用数学归纳法来证明这个结论.

当  $k = 1$  时, 显然成立.

当  $k = 2$  时, 只能有两种分批情况:  $T_1, T_2$  分在一批或各自组成一批处理. 比较它们的大小就可得到问题的解, 这在算法中已经体现. 结论成立.

假设  $k - 1$  时结论成立, 即利用上述动态规划算法能得到问题的最优分批. 不失一般性, 设工件按 *LPT* 排序后的顺序为  $T_1, T_2, \dots, T_{k-1}$ , 则  $p_1 \geq p_2 \geq \dots \geq p_{k-1}$ ; 设最优分批为  $\pi: B_1, B_2, \dots, B_l$ ; 且  $B_l = \{T_{r+1}, T_{r+2}, \dots, T_{k-1}\}$ .

下面证明结论对  $k$  个工件也成立.

设按上述动态规划算法得到  $k$  个工件的最优分批的最后一批为  $B_q$ , 其中  $B_q = \{T_{r'+1}, T_{r'+2}, \dots, T_k\}$ , 则一定有  $B_q \subseteq B_l \cup \{T_k\}$ , 也就是  $r' \geq r$ . 下面用反证法证明这一结论.

假设上述结论不成立, 即  $r' < r$ . 由假设各工件按 *LPT* 排列, 所以

$$p_{r'} > p_{r'+1} \geq p_r > p_{r+1} \quad (3)$$

此时设  $f'(k)$  表示工件从  $T_1$  到  $T_k$  对应的目标函数值, 则有

$$f'(k) = f(r') + p_{r'+1}(1 + \frac{k-r'-1}{C}) = f(r') + p_{r'+1}(1 + \frac{k-r'-2}{C}) + \frac{p_{r'+1}}{C} \quad (4)$$

在 (4) 式的右边中, 前两项是前  $k - 1$  个工件的目标函数值, 而

$$f(k-1) = f(r) + p_{r+1}(1 + \frac{k-r-2}{C}) \quad (5)$$

是前  $k - 1$  个工件的目标函数的最优值, 所以

$$f(r') + p_{r'+1}(1 + \frac{k-r'-2}{C}) \geq f(k-1) \quad (6)$$

由 (4), (6) 得

$$f'(k) \geq f(k-1) + \frac{p_{r'+1}}{C} \quad (7)$$

由 (3) 有

$$\frac{p_{r'+1}}{C} > \frac{p_{r+1}}{C} \quad (8)$$

把 (5)、(8) 代入 (7) 得

$$f'(k) > f(r) + p_{r+1}(1 + \frac{k-r-2}{C}) + \frac{p_{r+1}}{C} = f(r) + p_{r+1}(1 + \frac{k-r-1}{C}) \quad (9)$$

式 (9) 中等式的右端是将  $T_k$  直接并入  $B_l$  内处理, 而其余工件按  $B_1, B_2, \dots, B_{l-1}$  分批所对应的目标函数值. 从式 (9) 可以看出假设不成立, 即  $r' \geq r$ .

因此增加  $T_k$  工件后, 最优调度只能有下列三种情况:

- 1)  $T_k$  被分在  $B_l$  中处理;
- 2) 在  $B_l$  中某个工件  $T_q$  处断开,  $\{T_{q+1}, \dots, T_k\}$  为  $k$  个工件最优分批的最后一批;
- 3)  $T_k$  单独形成一批处理.

而上述动态规划算法是对这三种情况比较后取最小值. 所以结论成立.  $\square$

由于算法中动态规划过程的复杂性是  $O(n^2)$ , 而对工件进行  $LPT$  排序的时间复杂性是  $O(n \log n)$ , 所以此算法的复杂性为  $O(n^2)$ . 而且本算法在增加一个工件后进行分批时, 只从未增加这个工件的分批中最后一批工件进行讨论, 这样计算量减少了许多, 尤其当分的批数越多时, 算法的优势越明显.

由上述分批过程直接可得到下面的推论.

**推论 5.** 在问题  $1|c\text{-batch}, C|C_{\max}$  中, 工件按  $LPT$  序排列, 在最优调度中, 若  $p_j < \frac{p_i}{C}$ , 则  $T_i$  与  $T_j$  不能在一批中加工.

从此推论可知, 若  $p_{i+1} < \frac{p_i}{C}$ , 可直接从  $T_{i+1}$  处分批.

值得注意的是, 在这个问题中, 所有工件若按  $SPT$  规则排序, 可以得到类似的动态规划算法. 其不同点是初始值  $f(n+1) = 0$ , 而  $f(1)$  就是最优值.

## 5 数值例子

**例 2.** 考虑 10 个工件、 $C = 5$  的一个例子, 其中各工件的处理时间分别为:  $p = (10, 10, 3, 1.8, 1, 1, 1, 1, 1, 1)$ .

用上述动态规划算法计算, 过程如下:

$$\begin{aligned} f(0) &= 0, r_0 = 0, f(1) = 10, r_1 = 0, B_1 = \{T_1\} \\ f(2) &= \min\{20, 12\} = 12, r_2 = 0, B_2 = \{T_1, T_2\} \\ f(3) &= \min\{14, 22, 15\} = 14, r_3 = 0, B_3 = \{T_1, T_2, T_3\} \\ f(4) &= \min\{14, 24, 15.6, 15.8\} = 15.6, r_4 = 2, B_4 = \{T_3, T_4\}, \dots \end{aligned}$$

以此类推, 得:  $f(10) = 17.6, r_{10} = 4, B_{10} = \{T_5, T_6, T_7, T_8, T_9, T_{10}\}$ .

则最优值为 17.6; 通过反向追踪得到所有工件的一个最优分批为  $\{T_1, T_2\}, \{T_3, T_4\}, \{T_5, T_6, T_7, T_8, T_9, T_{10}\}$ .

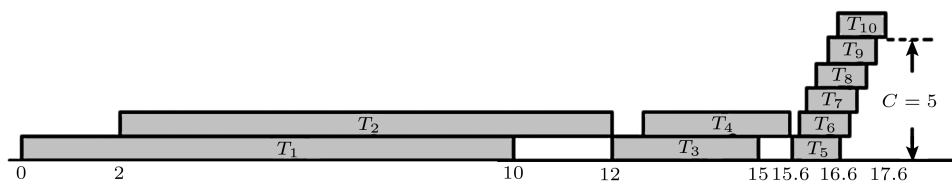


图 3 例 2 的调度图

Fig. 3 Gantt Charts of Example 2

## 6 结论

本文以钢铁工业中普遍采用的环形加热炉为背景, 提出的这种新型的连续型批调度问题, 在钢铁工业的其他处理过程中也普遍存在. 与传统的批处理机调度问题不同, 批中每个工件的开始加工时间及完工时间都不相同. 批中工件的进入、处理和离开都连续进行. 除极小化最大完工时间问题外, 其他目标函数如极小化总完工时间问题、工件有释放时间及工期限制等问题在实际中也经常遇到, 因此对于这一模型其他目标函数的研究值得进一步探讨.

### References

- 1 Tang G C, Zhang F, Luo S C, Liu L L. Theory of Modern Scheduling. Shanghai: Shanghai Popular Science Press, 2003. 83~113
- 2 Ikura Y, Gimple M. Scheduling algorithms for a single batch processing machine. *Operations Research Letters*, 1986, **5**(1): 61~65
- 3 Ahmadi J H, Ahmadi R H, Dasu S, Tang C S. Batching and scheduling jobs on batch and discrete processors. *Operations Research*, 1992, **40**(4): 750~763
- 4 Lee C Y, Uzsoy R, Martin-Vega L A. Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, 1992, **40**(4): 764~775
- 5 Uzsoy R, Lee C Y, Martin-Vega L A. Scheduling semiconductor test operations: Minimizing maximum lateness and number of tardy jobs on a single machine. *Naval Research Logistics*, 1992, **39**(3): 369~388
- 6 Brucker P, Garey N R, Johnson D S. Scheduling equal-length tasks under tree-like precedence constraints to minimize maximum lateness. *Mathematics of Operations Research*, 1977, **2**(2): 275~284
- 7 Lee C Y, Uzsoy R. Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research*, 1999, **37**(1): 219 ~236
- 8 Hochbaum D S, Landy D. Scheduling semiconductor burn-in operations to minimize total flowtime. *Operations Research*, 1997, **45**(6): 874~885
- 9 Uzsoy R. Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, 1994, **32**(7): 1615~1635
- 10 Albers S, Brucker P. The complexity of one-machine batching problems. *Discrete Applied Mathematics*, 1993, **47**(1): 87~107
- 11 Coffman E G Jr, Yannakakis M, Magazine M J, Bantos C. Batch sizing and job sequencing on a single machine. *Annals of Operations Research*, 1990, **26**(1): 135~147
- 12 Brucker P, Kovalyov M Y. Single machine batch scheduling to minimize the weighted number of late jobs. *Mathematical Methods of Operations Research*, 1996, **43**(1): 1~8
- 13 Ng C T, Cheng T C E, Yuan J J, Liu Z H. On the single machine serial batching scheduling problem to minimize total complete on time with precedence constraints, release dates and identical processing times. *Operations Research Letters*, 2003, **31**(6): 323~326
- 14 Brucker P, Gladky A, Hoogeveen H, Kovalyov M, Potts C, Tautenhahn T, Van S de Vdlde. Scheduling a batching machine. *Journal of Scheduling*, 1998, **1**(1): 31~54

**赵玉芳** 东北大学物流优化与控制研究所在读博士, 沈阳师范大学数学与系统科学学院副教授, 研究方向为生产调度与组合最优化.

(**ZHAO Yu-Fang** Ph. D. candidate in the Logistics Institute at Northeastern University. She is an associate professor in School of Mathematics and Systems Science at Shenyang Normal University. Her research interests include production scheduling and combinatorial optimization.)

**唐立新** 东北大学物流优化与控制研究所所长、教授、博士生导师. 研究方向为生产调度、物流与供应链管理和组合最优化.

(**TANG Li-Xin** Professor of the Logistics Institute at Northeastern University. His research interests include production scheduling, logistics and supply chain management, and combinatorial optimization.)