

# 一类新型批处理机调度问题的理论分析<sup>①</sup>

冯大光<sup>1,2</sup>, 唐立新<sup>1</sup>

(1. 东北大学物流优化与控制研究所, 辽宁省制造系统与物流优化重点实验室, 沈阳 110004;

2. 沈阳农业大学理学院, 沈阳 110866)

摘要: 钢卷在冷轧生产中,为了改进其性能,需要在罩式炉进行退火,退火过程由加热、保温和降温三段组成,而这三段处理时间由于工艺上的要求不能归结为一个时间,这与传统批处理机调度有明显的差别.对新型批处理机的总加权完成时间最小化问题建立了非线性整数规划模型,开发了基于动态规划的启发式算法.通过理论分析,获得该算法的误差性能比为 3.对于三段中的某一段板卷的处理时间相同的情况,证明了启发式算法的误差性能比是 2,而且证明是紧界.对于三段中的某二段板卷的处理时间相同的情况,证明了启发式算法是最优算法.对启发式算法扩展到带有任意段的加工时间的一般情况进行了性能分析.

关键词: 批处理机; 罩式退火炉; 三段加工时间; 动态规划

中图分类号: TP29 文献标识码: A 文章编号: 1007-9807(2012)06-0033-07

## 0 引言

传统的单机调度问题是指机器一次只能加工一个工件,而在半导体生产和钢铁企业中存在着多台机器可以同时加工多个工件的调度问题,这种机器称为批处理机,同时加工的多个工件称为一个批.单台批处理机的调度问题对于需要在机器上进行加工的多个工件如何组合,即如何分批,并确定批在机器上的加工次序,使得需要满足的目标函数最优.

在钢铁企业的各种金属材料和制品的生产过程中,热处理是不可缺少的重要环节之一.实际应用中的热处理工艺,都是由加热、保温和冷却三个阶段组成,整个工艺过程可以用加热速度、加热温度、保温时间、冷却速度以及总的延续时间(热处理周期)这几个基本工艺参数来描述<sup>[1]</sup>.

钢卷在罩式退火炉中进行热处理的加热、保温和冷却时间都是事先给定的,过程如图 1 所示.

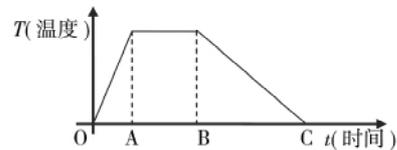


图 1 板卷热处理过程示意图

Fig. 1 Process of heat treatment for steel coils

在 OA 时间段,属于加热阶段,随着退火炉内温度的升高,板卷内原子的运动速度逐渐加快,原子间由于力的作用,调整和改变位置.当温度达到工艺要求时,停止升温;在 AB 时间段,属于保持温度的阶段,由于原子间的作用力,使得原子间的距离逐渐达到平衡状态;在 BC 时间段,属于缓慢降温的过程,在这个过程中,由于温度变化非常缓慢,原子间已经达到的平衡距离得以保持.在降温过程结束后,板卷的性能得到了改善,质量得到了提高.这三个过程必须连续进行,不能中断.

需要退火的板卷由于用途和要求不同,在每个过程中所要求的温度不同,即在罩式退火炉内各个过程的时间不同.虽然多个板卷可以同时

① 收稿日期: 2010-06-07; 修订日期: 2010-09-07.

基金项目: 国家自然科学基金资助项目(71032004).

作者简介: 冯大光(1972—),男,辽宁锦州人,博士生,副教授. Email: daguangf@126.com

罩式退火炉内进行退火,但是由于退火炉的数量有限,而需要进行退火处理的板卷往往又比较多,这就需要确定哪些板卷同时放在炉内进行退火(同时退火的板卷称为批),并确定退火的次序,使得所需要的目标函数达到最优.

### 1 一类新的批处理机问题

根据以上背景,提出如下的调度问题的模型:  
 $n$  个工件  $N = \{1, 2, \dots, n\}$  需要在一台能力为  $B$  的批处理机上加工,批处理机的能力是指机器最多同时加工的工件数量,当  $B \geq n$  时,称批处理机的能力是无限的.每个工件的加工时间为三步加工时间,即分为三段  $h_i, k_i, l_i (i = 1, 2, \dots, n)$ ,工件  $i$  的权值为  $w_i$ . $n$  个工件分成互斥的批  $B_1, B_2, \dots, B_r$ ,即  $B_i \cap B_j = \Phi (\forall i \neq j = 1, 2, \dots, r)$  且  $B_1 \cup B_2 \cup \dots \cup B_r = N$ .批  $B_i$  的加工时间  $p(B_i)$  为三步加工时间之和,批的三步加工时间分别记为  $h(B_i), k(B_i)$  和  $l(B_i)$ ,定义分别为  $h(B_i) = \max_{j \in B_i} \{h_j\}, k(B_i) = \max_{j \in B_i} \{k_j\}$  和  $l(B_i) = \max_{j \in B_i} \{l_j\}$ ,从而  $p(B_i) = h(B_i) + k(B_i) + l(B_i)$ .批  $B_i$  的权值  $W(B_i)$  定义为  $W(B_i) = \sum_{j \in B_i} \{w_j\}$ .批  $B_i$  的完成时间记为  $C(B_i)$ ,也可以简记为  $C_i$ ,目标函数是总加权完成时间最小化,即  $\min \sum_{i=1}^r W(B_i) C(B_i)$ .

关于单机调度题,冯大光<sup>[1]</sup>进行了 dynasearch 算法的研究与对比.Brucker<sup>[2]</sup>对能力无限的单台批处理机的总加权完成时间最小化问题的性质进行了分析,并提出了动态规划算法求得最优解.Dang<sup>[3]</sup>对工件具有准备时间,批处理机能力无限的总加权完成时间最小化问题建立了有向图,并用 Dijkstra 算法求给定工件次序的最优解,并证明了算法的最差性能比为 2.Hochbaum<sup>[4]</sup>对能力有限的单台批处理机总完成时间最小化问题最优解的性质进行了分析,并给出了最差性能比为 2 的动态规划算法.Uzsoy<sup>[5]</sup>对能力有限的单台批处理机总加权完成时间最小化问题进行了分析,并提出了一些启发式算法.Oulamara<sup>[6]</sup>对具有两台批处理机工件具有无等待

约束的 makespan 问题进行了研究,其中第一台为并行批处理机,第二台为串行批处理机,分析了问题的 NP 性,并给出了启发式算法和最差性能分析.Oulamara 等<sup>[7]</sup>对工件具有无等待约束,两台批处理机都为并行批处理机的流水车间 makespan 问题进行了研究,提出了多项式算法,并推广到具有  $m$  台并行批处理机的情形.Oulamara 和 Finke<sup>[8]</sup>对工件不具有无等待约束,第一台为并行批处理机,第二台为串行批处理机流水车间的 makespan 问题进行了研究,对于并行批处理机能力无限情形提出了多项式算法,当并行批处理机能力有限时,证明该问题是 NP 难的.Lin 和 Cheng<sup>[9]</sup>对具有两台串行批处理机流水车间的 makespan 问题进行了研究,证明该问题是 NP 难的.Cheng 和 Wang<sup>[10]</sup>指出一台为传统单机,另一台为串行批处理机且批处理机上具有准备时间的流水车间 makespan 最小化问题是 NP 完全的,并提出若干多项式可解的情形.Cheng 等<sup>[11]</sup>对  $m$  台同构并行机的总完成时间最小化问题进行了研究,其中每台机器都是串行批处理机,提出了该问题的动态规划算法.Cheng 等<sup>[12]</sup>对两台批处理机的 makespan 问题进行了研究,每台批处理机都需要准备时间,指出该问题是强 NP 难的.Potts<sup>[13]</sup>对调度问题给出了综述,对批处理机的研究现状做了详尽细致的回顾.

基于板卷热处理问题的调度模型创新之处在于,把每个工件的加工时间分成三个不可间断的部分,该模型不同于流水车间的三台批处理机问题,也与工件之间具有链式约束的问题有区别,这是第一次提出该类型的问题.

### 2 整数非线性规划模型

$n$  个工件在能力无限单台批处理机上的总加权完成时间最小化问题可以用整数非线性规划模型描述如下

$$\text{设 } x_{ij} = \begin{cases} 1, & \text{第 } i \text{ 个工件分配到第 } j \text{ 批} \\ 0, & \text{否则} \end{cases}$$

$$\min \sum_{j=1}^n W(B_j) C(B_j)$$

s. t.

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (1)$$

$$\sum_{i=1}^n x_{ij} \leq n \quad j = 1, 2, \dots, n \quad (2)$$

$$h(B_j) \geq h_i x_{ij} \quad i, j = 1, 2, \dots, n \quad (3)$$

$$k(B_j) \geq k_i x_{ij} \quad i, j = 1, 2, \dots, n \quad (4)$$

$$l(B_j) \geq l_i x_{ij} \quad i, j = 1, 2, \dots, n \quad (5)$$

$$p(B_j) = h(B_j) + k(B_j) + l(B_j) \quad j = 1, 2, \dots, n \quad (6)$$

$$W(B_j) = \sum_{i=1}^n w_i x_{ij} \quad j = 1, 2, \dots, n \quad (7)$$

$$C(B_1) = p(B_1) \quad (8)$$

$$C(B_{j+1}) = C(B_j) + p(B_{j+1}) \quad j = 2, 3, \dots, n-1 \quad (9)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, 2, \dots, n \quad (10)$$

$$W(B_j), p(B_j), C(B_j) \geq 0 \quad j = 1, 2, \dots, n \quad (11)$$

1) 表示每个工件只能属于一批; 2) 表示每批所含有的工件可以任意多; 3) 表示批的加热时间为批中工件的最大加热时间; 4) 表示批的保温时间为批中工件的最大保温时间; 5) 表示批的降温时间为批中工件的最大降温时间; 6) 表示批的加工时间为批的三部分处理时间之和; 7) 表示批权值为批中工件的权值之和; 8) 第一批的完成时间等于第一批的加工时间; 9) 第二到第  $n$  批的完成时间为其前面批的完成时间与该批的加工时间之和; 10) 表示决策变量为 0-1 变量; 11) 表示批权值、加工时间和完成时间非负.

批处理机能力无限且工件具有准备时间的总完成时间最小化问题的 NP 性到目前为止仍然没有确定<sup>[2]</sup>, 所以工件具有三步加工时间的批处理机总加权完成时间最小化问题的 NP 性还是未知的. 虽然问题可以通过上面的整数非线性规划描述, 并通过一些已有的软件, 如 Lingo, 求得最优解, 但当问题的规模较大时, 求解的速度比较慢, 因而研究该问题的启发式算法是具有实际意义的.

### 3 动态规划算法

动态规划算法是求解调度问题中的一类常用

算法, 对工件具有三步加工时间的总加权完成时间最小化问题, 实现了相应的算法 DP, 步骤如下:

1) 工件按  $h_i + k_i + l_i$  的非减次序排列, 当  $h_i + k_i + l_i$  相同时, 按权值的非增顺序排列;

2) 动态规划的迭代方程为

$$f(n+1) = 0;$$

$$f(i) = \min_{i+1 \leq k \leq n} \{f(k) + [\max_{i \leq j \leq k} (h_j) +$$

$$\max_{i \leq j \leq k} (k_j) + \max_{i \leq j \leq k} (l_j)] \sum_{r=i}^n w_r\},$$

$$i = n, n-1, \dots, 1;$$

3)  $f(1)$  即为给定工件次序的最优解.

**定理 1** 应用动态规划算法 DP 得到的解至多是最优解的 3 倍.

**证明** 假设  $f(O^*(N))$  是工件集  $N$  的最优调度  $O^*$  所对应的目标函数值,  $f(D^*(N))$  是应用算法 DP 得到解  $D^*$  所对应的目标函数值, 即要证明  $f(D^*(N)) / f(O^*(N)) \leq 3$ .

如果工件  $1, 2, \dots, m$  是任意一个调度  $S$  中的第一批, 记  $N_1 = N \setminus \{1, 2, \dots, m\}$ , 则有如下的表达式

$$f(S(N)) = \left( \max_{1 \leq i \leq m} h_i + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(S(N_1))$$

并且如果  $S(N)$  是工件集  $N$  的最优调度, 则  $S(N_1)$  是工件集  $N_1$  的最优调度.

应用数学归纳法对该定理进行证明.

1) 当  $|N| = 1$  时, 显然有

$$f(D^*(N)) / f(O^*(N)) = 1 \leq 3, \text{ 定理成立.}$$

2) 当  $|N| > 1$  时, 假设对于  $N$  的任意一个子集  $K, K \subset N$ , 有  $f(D^*(K)) / f(O^*(K)) \leq 3$ , 分两种情况对定理进行证明.

1) 在最优调度中, 工件  $1, 2, \dots, m$  作为第一批  $N_1$  与上面的表达方式相同, 则最优调度的目标函数为

$$f(O^*(N)) = \left( \max_{1 \leq i \leq m} h_i + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \times \sum_{r=1}^n w_r + f(O^*(N_1)).$$

**算法 DP** 一定能够搜索到以工件  $1, 2, \dots, m$

作为第一批的调度 则有

$$f(D^*(N)) \leq \left( \max_{1 \leq i \leq m} h_i + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \times \sum_{r=1}^n w_r + f(D^*(N_1)) ,$$

并且由假设  $\frac{f(D^*(N_1))}{f(O^*(N_1))} \leq 3$ .

对于实数  $a > b > 0$  函数  $\frac{x+a}{x+b}$  是  $x$  的递减函数

这是因为  $\left( \frac{x+a}{x+b} \right)' = \frac{b-a}{(x+b)^2} < 0$  , 从而有

$$\frac{x+a}{x+b} \leq \frac{a}{b} .$$

所以  $\frac{f(D^*(N))}{f(O^*(N))} \leq \frac{\left( \max_{1 \leq i \leq m} h_i + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(D^*(N_1))}{\left( \max_{1 \leq i \leq m} h_i + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_1))} \leq 3$  , 即当最优解的第一批工件为  $1, 2, \dots, m$  时 定理成立.

2) 在最优调度中 , 以工件  $i_1, i_2, \dots, i_m$  作为第一批 , 其中  $m < i_m$  , 记  $J_1 = \{i_1, i_2, \dots, i_m\}$  ,  $N_1 = N - J_1$  , 从而有  $f(O^*(N)) = \left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_1))$  .

由于批处理机的能力是无限的 , 把  $O^*(N)$  中序号小于  $i_m$  且不在第一批中的工件都移到第一批 , 设得到的新的调度为  $O(N)$  , 由于  $O^*(N)$  是最优调度 , 可以得到  $f(O^*(N)) \leq f(O(N))$  .

设  $J_2 = \{i \mid i \leq i_m\}$  并记  $N_2 = N - J_2$  , 于是有  $N_2 \subset N_1$  则  $f(O(N)) = \left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \times \sum_{r=1}^n w_r + f(O(N_2))$  .

由于  $O(N)$  是由  $O^*(N)$  把下标小于  $i_m$  且不在第一批中的工件都移到第一批而得到的 , 而  $N_1$  中其他工件的分批没有改变得到了  $N_2$  , 所以有  $f(O^*(N_1)) \geq f(O(N_2))$  , 并且由于  $O^*$  是最优调度 , 所以有  $f(O(N_2)) \geq f(O^*(N_2))$  , 从而

得到

$$f(O^*(N)) \geq \left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) \sum_{r=1}^n w_r +$$

$$f(O(N_2)) \geq \left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_2)) .$$

由于  $i_m$  是工件集  $J_2$  中下标最大的工件 , 所以有

$$\max \left\{ \max_{i \in J_2} h_i, \max_{i \in J_2} k_i, \max_{i \in J_2} l_i \right\} \leq h_{i_m} + k_{i_m} + l_{i_m} ,$$

从而

$$\max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \leq 3(h_{i_m} + k_{i_m} + l_{i_m}) .$$

又因为  $i_m \in J_1$  , 所以

$$h_{i_m} + k_{i_m} + l_{i_m} \leq \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i ,$$

从而

$$\max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \leq 3 \left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) ,$$

$$\text{即 } \frac{1}{3} \left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \leq$$

$$\left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) ,$$

$$\text{从而有 } f(O^*(N)) \geq$$

$$\left( \max_{i \in J_1} h_i + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_2)) \geq \frac{1}{3} \left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_2)) .$$

算法 DP 可以搜索到以工件集  $J_2$  为第一批的情况 , 所以有

$$f(D^*(N)) \leq \left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r +$$

$$f(D^*(N_2)) .$$

由于  $N_2 \subset N$  根据假设  $\frac{f(D^*(N_2))}{f(O^*(N_2))} \leq 3$  所以

$$\frac{f(D^*(N))}{f(O^*(N))} \leq$$

$$\frac{\left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r + f(D^*(N_2))}{\left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_2))} \leq 3 .$$

$$\frac{1}{3} \left( \max_{i \in J_2} h_i + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_2))$$

综上所述 , 该定理成立.

### 4 若干特殊情况

如果工件的三步加工时间中,某些部分的取值相同,上面的启发式算法的性能又有所增强,分各种情况进行如下的讨论.

**定理 2** 如果工件的三步加工时间满足一致性规则,即如果  $h_i \leq h_{i+1}$ , 则必有  $k_i \leq k_{i+1}$  和  $l_i \leq l_{i+1}$  ( $\forall i = 1, 2, \dots, n-1$ ), 则存在  $h+k+l$  非减顺序的最优调度,并且应用上面的动态规划算法可以求得一个最优调度.

**证明** 假设在最优调度中存在着工件  $i$  和  $j$ , ( $i < j$ ) 且  $i \in B_r, j \in B_s$  ( $s < r$ ), 即批  $B_r$  在批  $B_s$  后加工,把工件  $i$  移到批  $B_s$  中时,批  $B_s$  的加工时间不会增加,并且工件  $i$  和批  $B_r$  的完成时间也不增加,所以目标函数没有增加,即移动工件  $i$  后的解仍然是最优解,所以结论成立.

**定理 3** 对于所有工件而言,如果三个时间中,有两个时间相同,则工件存在  $h+k+l$  非减顺序的最优调度,并且应用上面的动态规划算法可以求得一个最优调度.

**证明** 不妨假设对于任意工件  $i$ , 有  $h_i = h$  和  $l_i = l$ . 假设在最优调度中存在着工件  $i$  和  $j$ , ( $i < j$ ) 且  $i \in B_r, j \in B_s$  ( $s < r$ ), 即批  $B_r$  在批  $B_s$  后加工,由于  $k_i \leq k_j$ , 把工件  $i$  移到批  $B_s$  中时,批  $B_s$  的加工时间不会增加,并且工件  $i$  和批  $B_r$  的完成时间也不增加,所以目标函数没有增加,即移动工件  $i$  后的解仍然是最优解,所以结论成立.

注: 实际上, 这种情况包含了 Brucker<sup>[5]</sup> 单台批处理机能力无限的总加权完成时间最小化问题.

**定理 4** 对于所有工件而言,如果三个时间中,有一个时间相同,则当工件按  $h+k+l$  非减顺序排列时,算法 DP 的最差性能比为 2, 并且该界是紧界.

**证明** 假设对于任意工件  $i$ , 有  $h_i = h$ . 下面应用数学归纳法对该定理进行证明.  $f(O^*(N))$ 、 $f(D^*(N))$  与上面定义相同.

1) 当  $|N| = 1$  时, 显然有  $f(D^*(N)) /$

$f(O^*(N)) = 1 \leq 2$ , 即定理成立.

2) 当  $|N| > 1$  时, 假设对于  $N$  的任意一个子集  $K, K \subset N$ , 有

$f(D^*(K)) / f(O^*(K)) \leq 2$ , 分两种情况进行讨论.

1) 在最优调度中, 工件  $1, 2, \dots, m$  作为第一批  $N_1$  与上面的表达方式相同, 则最优调度的目标函数表示为

$$f(O^*(N)) = \left( h + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_1)).$$

算法 DP 能够搜索到以工件  $1, 2, \dots, m$  作为第一批的调度, 则有

$$f(D^*(N)) \leq \left( h + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(D^*(N_1)), \text{ 并且 } \frac{f(D^*(N_1))}{f(O^*(N_1))} \leq 2. \text{ 于是有}$$

$$\frac{f(D^*(N))}{f(O^*(N))} \leq \frac{\left( h + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(D^*(N_1))}{\left( h + \max_{1 \leq i \leq m} k_i + \max_{1 \leq i \leq m} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_1))},$$

即当最优解的第一批的工件为  $1, 2, \dots, m$  时, 定理成立.

2) 在最优调度中, 以工件  $i_1, i_2, \dots, i_m$  作为第一批, 其中  $m < i_m$ , 记  $J_1 = \{i_1, i_2, \dots, i_m\}, N_1 = N - J_1$ , 从而有  $f(O^*(N)) = \left( h + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i \right) \sum_{r=1}^n w_r + f(O^*(N_1))$ .

由于批处理机的能力是无限的, 把  $O^*(N)$  中序号小于  $i_m$  且不在第一批中的工件都移到第一批, 设得到的新的调度为  $O(N)$ , 由于  $O^*(N)$  是最优调度, 可以得到  $f(O^*(N)) \leq f(O(N))$ .

设  $J_2 = \{i \mid i \leq i_m\}$ , 并记  $N_2 = N - J_2$ , 于是有  $N_2 \subset N_1$ , 则

$$f(O(N)) = \left( h + \max_{i \in J_2} k_i + \max_{i \in J_2} l_i \right) \sum_{r=1}^n w_r + f(O(N_2)).$$

由于  $O(N)$  是由  $O^*(N)$  把下标小于  $i_m$  且不在第一批中的工件都移到第一批而得到的, 而  $N_1$

中其他工件的分批没有改变得到了  $N_2$ , 所以有  $f(O^*(N_1)) \geq f(O(N_2)) \geq f(O^*(N_2))$ .

由于  $f(O^*(N)) \geq (h + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i) \sum_{r=1}^n w_r + f(O(N_2))$ , 所以  $f(O^*(N)) \geq (h + \max_{i \in J_1} k_i + \max_{i \in J_1} l_i) \sum_{r=1}^n w_r + f(O^*(N_2))$  即

$$f(O^*(N)) \geq f(O(N_2)) + h \sum_{k=1}^n w_k + (\max_{i \in J_1} k_i + \max_{i \in J_1} l_i) \sum_{r=1}^n w_r.$$

由于  $i_m$  是工件集  $J_2$  中下标最大的工件, 所以有  $\max\{\max_{i \in J_2} k_i, \max_{i \in J_2} l_i\} \leq k_{i_m} + l_{i_m}$  从而,  $\max_{i \in J_2} k_i + \max_{i \in J_2} l_i \leq 2(k_{i_m} + l_{i_m})$ . 又因为  $i_m \in J_1$ , 所以  $k_{i_m} + l_{i_m} \leq \max_{i \in J_1} k_i + \max_{i \in J_1} l_i$  从而

$$\max_{i \in J_2} k_i + \max_{i \in J_2} l_i \leq 2(\max_{i \in J_1} k_i + \max_{i \in J_1} l_i), \text{ 即}$$

$$\frac{1}{2}(\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \leq \max_{i \in J_1} k_i + \max_{i \in J_1} l_i, \text{ 从而有}$$

$$\frac{1}{2}(\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \leq (\max_{i \in J_1} k_i + \max_{i \in J_1} l_i) \text{ 所以}$$

$$f(O) \geq f(O^*(N_2)) +$$

$$h \sum_{k=1}^n w_k + \frac{1}{2}(\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \sum_{r=1}^n w_r.$$

算法 DP 能够搜索到以工件集  $J_2$  为第一批的情况, 所以有

$$f(D^*(N)) \leq (\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \sum_{r=1}^n w_r + h \sum_{r=1}^n w_r + f(D^*(N_2)).$$

由于  $N_2 \subset N$  根据假设有  $\frac{f(D^*(N_2))}{f(O^*(N_2))} \leq 2$  所以

$$\frac{f(D^*(N_2)) + h \sum_{r=1}^n w_r}{f(O^*(N_2)) + h \sum_{r=1}^n w_r} \leq 2 \text{ 从而 } \frac{f(D^*(N))}{f(O^*(N))} \leq$$

$$\frac{(\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \sum_{r=1}^n w_r + h \sum_{r=1}^n w_r + f(D^*(N_2))}{\frac{1}{2}(\max_{i \in J_2} k_i + \max_{i \in J_2} l_i) \sum_{r=1}^n w_r + h \sum_{r=1}^n w_r + f(O^*(N_2))} \leq 2$$

综上所述, 该定理成立.

下面的例子 (见表 1) 说明该界是紧界. 批处

理机的能力为 2.

表 1 紧界例子工件数据

Table 1 Data of jobs for worst case

工件	$J_1$	$J_2$
加热时间 $h$	$\varepsilon$	$\varepsilon$
保温时间 $k_i$	1	$10\varepsilon$
降温时间 $l_i$	$\varepsilon$	1
权值 $w_i$	$\varepsilon$	2

注: 工件已经按  $h + k + l$  的非减次序排列.

算法 DP 过程如下

第 1 阶段:  $f(3) = 0$ ;

第 2 阶段:  $f(2) = f(3) + 2(\varepsilon + 10\varepsilon + 1) = 2 + 22\varepsilon$ ;

第 3 阶段:  $f(1) = \min\{f(2) + (2 + \varepsilon)(2\varepsilon + 1), f(3) + (2 + \varepsilon)[\varepsilon + \max(10\varepsilon, 1) + \max(\varepsilon, 1)]\}$   
 $= \min\{\varepsilon^2 + 27\varepsilon + 4, 4 + 4\varepsilon + \varepsilon^2\}$   
 $= 4 + 4\varepsilon + \varepsilon^2$ ;

即算法 DP 的解为两个工件作为一批,

$$f(D^*(J_1, J_2)) = 4 + 4\varepsilon + \varepsilon^2.$$

该例子的最优解为工件  $J_2$  作为第一批,  $J_1$  作为第二批进行加工, 目标函数值为

$$f(O^*(J_1, J_2)) = (2 + \varepsilon)(1 + 11\varepsilon) + (1 + 2\varepsilon)\varepsilon = 13\varepsilon^2 + 24\varepsilon + 2; \text{ 所以}$$

$$\lim_{\varepsilon \rightarrow 0} \frac{f(D^*(J_1, J_2))}{f(O^*(J_1, J_2))} = \lim_{\varepsilon \rightarrow 0} \frac{4 + 2\varepsilon + \varepsilon^2}{13\varepsilon^2 + 24\varepsilon + 2} = 2.$$

定理 5 如果工件的加工时间分为  $s$  个部分, 算法 DP 得到的解至多是最优解的  $s$  倍.

说明 该证明过程的难点在于归纳法的第二步. 在进行放缩时, 把最优解的第一批的加工时间缩小为以连续工件下标为第一批的  $1/s$  就可以了.

## 5 结束语

提出了一种新型批处理机的调度问题, 即三步加工时间的批处理机问题, 建立了该问题的数学模型, 提出了动态规划算法, 并证明了算法的最差性能比. 对特殊情况进行了讨论. 将来的进一步研究是, 对于容量有限的批处理机问题寻找有效的启发式算法, 并且该问题的计算复杂性也是值得探讨的问题.

## 参考文献:

- [1]冯大光,唐立新. 具有最大总加权满意度的单机调度问题的 dynasearch 算法[J]. 管理科学学报,2006,9(4): 40-50.  
Feng Daguang, Tang Lixin. Dynasearch algorithms for single machine scheduling problem with total weighted satisfaction [J]. Journal of Management Sciences in China, 2006, 9(4): 40-50. (in Chinese)
- [2]Brucker P, Gladky A, Hoogeveen H, et al. Scheduling a batching machine [J]. Journal of Scheduling, 1998, 1(1): 31-54.
- [3]Dang C Y, Kang L Y. Batch-processing scheduling with setup times [J]. Journal of Combinatorial Optimization, 2004, 8(2): 137-146.
- [4]Hochbaum D S, Landy D. Scheduling semiconductor burn-in operations to minimize total flowtime [J]. Operations Research, 1997, 45(6): 874-885.
- [5]Uzsoy R, Yang Y. Minimizing total weighted completion time on a single batch processing machine [J]. Production and Operations Management, 1997, 6(1): 57-73.
- [6]Oulamara A. Makespan minimization in a no-wait flow shop problem with two batching machines [J]. Computers & Operations Research, 2007, 34(1): 1033-1050.
- [7]Oulamara A, Kovalyov M Y, Finke G. Scheduling a no-wait flowshop with unbounded batching machines [J]. IIE Transactions, 2005, 37(8): 685-696.
- [8]Oulamara A, Finke G. Flowshop problems with batch processing machines [J]. International Journal of Mathematical Algorithms, 2001, 2(1): 269-287.
- [9]Lin B M T, Cheng T C E. Batch scheduling in the no-wait two-machine flowshop to minimize the makespan [J]. Computers and Operations Research, 2001, 28(1): 613-624.
- [10]Cheng T C E, WANG G Q. Batching and scheduling to minimize the makespan in the two-machine flowshop [J]. IIE Transactions, 1998, 30(1): 447-453.
- [11]Cheng T C E, Chen Z L, Kovalyov M Y, et al. Parallel-machine batching and scheduling to minimize total completion time [J]. IIE Transactions, 1996, 28(1): 953-956.
- [12]Cheng T C E, Lin B M, Toker A. Makespan minimization in the two-machine flowshop batch scheduling problem [J]. Naval Research Logistics, 2000, 47(1): 128-144.
- [13]Potts C N, Mikhail Y, Kovalyov M. Scheduling with batching: A review [J]. European Journal of Operational Research, 2000, 120(2): 228-249.

## Theoretical analysis of scheduling of a new batching machine

FENG Da-guang<sup>1 2</sup>, TANG Li-xin<sup>1</sup>

1. The Logistics Institute, Northeastern University, Liaoning Key Laboratory of Manufacturing System and Logistics, Shenyang 110004, China;
2. Science of Institute, Shenyang Agricultural University, Shenyang 110866, China

**Abstract:** The processing time of the steel coils, which are to be processed in the bell type annealing furnace to improve quality, is composed of three operations steps, that is, heating, keeping and lowering temperature. The three-step processing of the jobs cannot be regarded as a whole process or three independent processes for  
(下转第 48 页)

- Chen Shou , Li Shuangfei , Li Chuanguo. Stock price response to order imbalance and change of volume [J]. Journal of Management Sciences in China , 2010 , 13( 9) : 68 - 75. ( in Chinese)
- [18] Parlour C A , Seppi D J. Limit order markets: A survey [J]. Handbook of Financial Intermediation and Banking , 2008 , 5.

## Order allocation model: A model combining microstructure theory and asset allocation theory

ZHENG Zhen-long , LIU Yang-shu

Department of Finance , School of Economics , Xiamen University , Xiamen 361005 , China

**Abstract:** This paper extends asset allocation model to order allocation model , which bridges the gap between microstructure theory and asset allocation theory. In particular , by maximizing investor's utility of order submission problem in the same way with solving asset allocation problem , we receive a close-form solution on allocation about order submission. In addition , we prove that CAPM is a special case of our model when submission is constrained to be marginal market order.

**Key words:** order submission; order allocation; asset allocation

~~~~~  
( 上接第 39 页)

technical reasons , and is different from the classical batching machine. An integer nonlinear programming is proposed and a heuristic algorithm based on dynamic programming is applied to the total weighted completion time for the new batching machine. The worst case performance of the heuristic algorithm is proved to be at most 3. If any two steps' processing times are the same , the heuristic algorithm can obtain the optimal solution. If any one step's processing time of all the jobs is the same , the worst performance of the heuristic algorithm is proved to be at most 2 and the bound is tight. We also analyze the worst case of the heuristic algorithm for the general case where jobs processing are composed of any step-processing.

**Key words:** batching machine; bell type annealing furnace; three-step processing time; dynamic programming