

# 批处理机上具有两类释放时间的工件集竞争调度问题

赵晓丽<sup>1</sup> 官华<sup>2</sup> 车平<sup>3</sup>

**摘要** 研究了两个工件集合竞争在一台批处理机上加工的调度问题, 其中每个集合的工件具有一个共同的释放时间. 批处理机可以同时加工多个工件作为一批, 每批的加工时间为该批工件中加工时间的最大值. 基于两类释放时间的大小, 针对无界批处理机上最小化一个集合工件的最大完工时间、最大延迟以及总完工时间, 使得另一个集合工件的最大完工时间不超过给定上界问题, 分别给出了最优求解方法. 针对有界批处理机上最小化一个集合工件的最大完工时间, 使得另一个集合工件的最大完工时间不超过给定上界问题, 证明为一般意义 NP-难问题, 并给出伪多项式时间最优求解方法.

**关键词** 调度, 竞争工件集合, 释放时间, 批处理机

**引用格式** 赵晓丽, 官华, 车平. 批处理机上具有两类释放时间的工件集竞争调度问题. 自动化学报, 2020, 46(1): 168–177

**DOI** 10.16383/j.aas.2018.c170536

## The Competing Job Sets Scheduling Problems With Two Types of Release Dates on Batching-Machine

ZHAO Xiao-Li<sup>1</sup> GONG Hua<sup>2</sup> CHE Ping<sup>3</sup>

**Abstract** We consider two sets of jobs competing a single parallel-batching machine where the jobs of each set have a common release date. The parallel-batching machine can process several jobs simultaneously as a batch, and the processing time of a batch is the largest of the job processing times in the batch. Based on the sizes of the two types of release dates, the optimal solutions are presented on minimizing the makespan, the maximum lateness and the total completion time of one set of jobs, respectively, such that the makespan of the other set of jobs is maintained under a fixed value on an unbounded batching machine. For the bounded batching machine, minimizing the makespan of one set of jobs subject to an upper bound on the makespan of the other set of jobs is proved to be ordinary NP-hard, and pseudo-polynomial time optimal solutions are given.

**Key words** Scheduling, competing job sets, release date, batching machine

**Citation** Zhao Xiao-Li, Gong Hua, Che Ping. The competing job sets scheduling problems with two types of release dates on batching-machine. *Acta Automatica Sinica*, 2020, 46(1): 168–177

本文主要研究两个工件集合同时在一台批处理机上竞争加工的生产调度问题, 其中每个集合的工件具有共同的释放时间. 传统的生产调度研究主要集中在所有工件对应一个客户需求, 作为一个整体考虑一致的性能指标. 随着经济的发展和人们消费水平的提高, 顾客对商品的个性化和多样化的需求越来越高, 所以这种传统的所有工件对应一致目标

的生产调度问题已不能再满足多客户需求情况下的生产调度问题, 因此迫切需要研究考虑多客户为了自己的利益竞争使用共同资源的生产调度问题. 此外, 在实际工业生产中为了提高机器利用率、降低能耗等目的, 经常将具有相似特征的工件组批加工. 本文从钢铁企业中提炼出这类满足两个顾客需求的批处理机生产调度问题, 同时工件到达批处理机生产时具有两类不同的到达时间. 在钢铁工业的初轧厂中, 经过模铸生成的钢锭要先进入均热炉中加热或保温一段时间, 使钢锭内部温度均匀达到轧制的要求, 而均热炉能同时加热多个钢锭可以看作一台批处理机, 均热炉中一批的加工时间为炉中钢锭最长的均热时间. 均热后的钢锭经轧制生成的钢产品一部分直接销售给顾客, 一部分进入热轧阶段进行热轧(如图 1 所示). 这里直接销售给顾客与进行热轧的钢产品由于需求目的的不同, 对钢级的要求也不同, 所以在均热炉中均热的温度与时间往往是不同的, 因此一般情况下目的相同的钢锭可以组成批进行均热, 而目的不同的钢锭不组成批, 即批是不可

收稿日期 2017-09-22 录用日期 2018-04-16  
Manuscript received September 22, 2017; accepted April 16, 2018

国家自然科学基金项目(71402021), 辽宁省科技厅自然科学基金计划重点项目(20170540790), 沈阳市科技计划项目(17231131)资助  
Supported by National Natural Science Foundation of China (71402021), Key Natural Science Foundation Project of Liaoning Province (20170540790), and Science and Technology Project of Shenyang City (17231131)

本文责任编辑 乔俊飞  
Recommended by Associate Editor QIAO Jun-Fei  
1. 沈阳航空航天大学理学院 沈阳 110136 2. 沈阳理工大学理学院 沈阳 110159 3. 东北大学理学院数学系 沈阳 110819  
1. School of Science, Shenyang Aerospace University, Shenyang 110136 2. School of Science, Shenyang Ligong University, Shenyang 110159 3. Department of Mathematics, College of Sciences, Northeastern University, Shenyang 110819

兼容的. 此外, 两个不同目的的钢锭集合到达批处理机上加工的到达时间可以看作两类不同的释放时间. 基于上述初轧阶段的生产特点, 为了满足两个不同目的的生产需求, 同时也为了提高初轧阶段的生产效率、降低能耗, 如何在批处理机上对工件进行组批, 调度两个工件集合的加工顺序是至关重要的研究问题之一.

下面将给出一个具体的实例来解释我们所研究的一台批处理机上具有两类释放时间的工件集竞争调度问题. 在上面描述的钢锭初轧生产过程中, 假设热轧阶段需要 3 个钢产品, 对应的钢锭工件集合为  $J^A = \{J_1^A, J_2^A, J_3^A\}$ . 顾客需要直接购买 2 个钢产品, 对应的钢锭工件集合为  $J^B = \{J_1^B, J_2^B\}$ . 每个钢锭的加热时间分别为  $p_1^A = 1, p_2^A = 2, p_3^A = 3; p_1^B = 1, p_2^B = 2$ . 集合  $J^A$  中工件的共同释放时间为  $r^A = 0$ , 集合  $J^B$  中工件的共同释放时间为  $r^B = 2$ . 批处理机的能力为  $b = 2$ , 不同集合的工件不能组成批加工. 所考虑的目标函数为在满足集合  $J^B$  中工件的最大完工时间  $C_{\max}^B$  不超过一个给定上界  $Q_B = 7$  的约束下, 最小化集合  $J^A$  中工件的最大完工时间  $C_{\max}^A$ . 具体的参数介绍见第 1 节中问题描述. 对于上述实例, 我们能获得一个最优调度  $\pi$  满足  $C_{\max}^B = 6 < Q_B$  的条件下集合  $J^A$  工件的最大完工时间最小值为  $C_{\max}^A = 4$  (如图 2 所示). 最优调度不仅满足了顾客的需求, 同时通过最小化热轧阶段所需工件的最大完工时间, 为热轧阶段提高了生产效率.

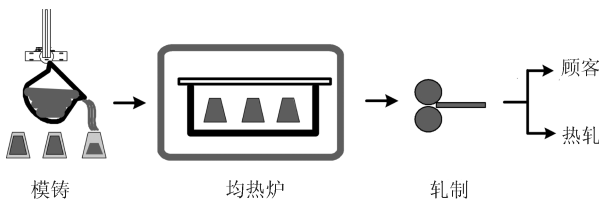


图 1 钢铁工业钢锭初轧过程

Fig. 1 Ingot rolling process in the steel industry

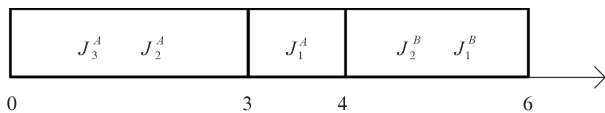


图 2 最优调度  $\pi$

Fig. 2 The optimal schedule  $\pi$

近几年来, 两个或多个工件集竞争调度的问题作为调度领域中多目标问题的特殊情况备受学者们的关注, 其中每个工件集有各自的优化目标. 不同目标工件集的存在使得问题的求解比所有工件对应一个工件集的情况更加困难. 多个工件集竞争的调度问题最初由 Baker 等<sup>[1]</sup> 以及 Agnetis 等<sup>[2]</sup> 提出, 前者研究了单机上最小化两个工件集目标函数线性

组合形式问题的复杂性, 后者研究了单机上依赖两个工件集不同目标函数的有约束优化形式及帕累托形式问题的复杂性. 关于多个工件集竞争的调度问题, 读者可参阅 Perez-Gonzalez 等<sup>[3]</sup> 以及 Agnetis 等<sup>[4]</sup>. 本文主要研究两个工件集目标函数有约束优化的形式, 即找到一个最优调度最小化一个工件集的目标函数使得另一个工件集的目标函数不超过一个给定上界的问题. 下面主要针对两个工件集竞争的有约束优化调度问题进行简单综述.

关于单机批处理机上两个工件集竞争的调度问题, Li 等<sup>[5]</sup> 提出了多项式或伪多项式时间的算法来求解无界批处理机上批不可兼容与批可兼容两种情况下各种正则目标函数结合的调度问题, 其中针对批不可兼容问题  $1|p - batch, b > n, IF|f_{\max}^A : f_{\max}^B \leq Q_B$  与  $1|p - batch, b > n, IF|\sum f_j^A : f_{\max}^B \leq Q_B$  分别提出了时间复杂度为  $O(nn_A n_B \log(Q_U - Q_L))$  与  $O(nn_A n_B P)$  的算法, 这里  $IF$  表示批不可兼容, 具体参数请见文献 [5]. Agnetis 等<sup>[4]</sup> 针对无界批处理机上批不可兼容的问题  $1|p - batch, b > n|C_{\max}^A : C_{\max}^B \leq Q_B$  与  $1|p - batch, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  分别提出了时间复杂度为  $O(n)$  与  $O(n_A + n_B^4)$  的多项式时间算法. Fan 等<sup>[6]</sup> 针对有界批处理机上批不可兼容问题  $1|p - batch, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  提出了时间复杂度为  $O(n \log n)$  的最优算法, 同时证明了上述问题当批可兼容情况下以及批不可兼容情况下问题  $1|p - batch, b < n|\sum C_j^A : C_{\max}^B \leq Q_B$  均为 NP-难问题. 最近, Wang 等<sup>[7]</sup> 研究了有界批处理机上工件具有不同尺寸大小、加工时间相同的两个工件集竞争调度问题, 证明出目标函数为最小化一个工件集的最大完工时间, 使得另一个工件集的最大完工时间不超过给定上界问题, 不存在多项式时间算法具有常数的最坏情况比, 同时提出了一个有效算法获得最优解的下界以及两个启发式算法对问题进行求解. 虽然上面的文献考虑了批处理机上两个工件集竞争的调度问题, 但所考虑的问题都是工件具有相同的释放时间. Tang 等<sup>[8]</sup> 研究了单机无界批处理机与有界批处理机上工件具有恶化加工时间的两个工件集竞争调度问题, 基于批的兼容性与工件释放时间的差异性, 作者对一些目标函数的结合给出了时间复杂性, 其中当批不可兼容且工件存在不同释放时间时, 针对批无界问题  $1|p - batch; p_j^X = \alpha_j^X(a + bt); r_j^X; c = \infty; IF|C_{\max}^A : C_{\max}^B$  提出了时间复杂度为  $O(n_A n_B n \log(Q'_U - Q'_L))$  的最优求解算法, 同时证明出批有界问题  $1|p - batch; p_j^X = \alpha_j^X(a + bt); r_j^X; c < n|C_{\max}^A : C_{\max}^B$  对于批可兼容与不可兼容两种情况均为 NP-难问题, 具体参数请见文献 [8].

关于单机上具有释放时间的两个工件集竞争调度问题, Yin 等<sup>[9]</sup> 证明了问题  $1|r_j^X|\sum T_j^A : L_{\max}^B$  为强 NP-难问题, 并提出了分支定界算法进行求解. Lee 等<sup>[10]</sup> 对问题  $1|r_j^X|\sum T_j^A : T_{\max}^B$  提出了分支定界算法与遗传算法进行求解. Wu 等<sup>[11]</sup> 证明了问题  $1|r_j^X|\sum C_j^A : \sum C_j^B$  为强 NP-难问题, 同时提出了分支定界算法、蚁群算法以及遗传算法对问题进行求解. Cheng 等<sup>[12]</sup> 对问题  $1|r_j^X|\sum w_j C_j^A : L_{\max}^B$  提出了分支定界算法以及模拟退火算法进行求解. 以上文献虽然考虑了工件带有不同的释放时间, 但都是利用智能算法对问题提出近优算法进行求解, 没有从理论分析的角度对具有不同释放时间的两个工件集竞争调度问题进行研究. Dover 等<sup>[13]</sup> 研究了单机上工件具有相同加工时间与不同释放时间的两个工件集竞争调度问题, 对各种正则目标函数的结合给出了时间复杂性.

另一方面, 关于批处理机上的调度问题也有许多学者研究. 一台批处理机可以同时加工多个工件, 一批的加工时间为该批中工件加工时间最大者, 同一批的所有工件同时开始加工同时结束, 一旦一批开始加工就不能中断, 也不能再加入其他的工件到该批. Potts 等<sup>[14]</sup> 对批处理机上的调度问题进行了综述. 根据批处理机的能力, 可以分为无界批处理机与有界批处理机. Brucker 等<sup>[15]</sup> 研究了单机批无界与批有界两种情况下工件具有相同释放时间的各种目标函数的时间复杂性. Lee 等<sup>[16]</sup> 研究了单机批处理机上工件具有不同释放时间, 目标函数为最小化工件最大完工时间问题的时间复杂性, 对批无界情况提出了多项式时间动态规划算法, 对批有界情况当工件具有两类释放时间时提出了伪多项式时间的动态规划算法进行求解, 对一般情况提出了启发式算法. Liu 等<sup>[17]</sup> 同样对于上述目标函数证明了批有界情况下问题为 NP-难问题, 并对工件具有有限个释放时间情况提出了伪多项式时间算法, 对一般情况设计了贪婪算法. Liu 等<sup>[18]</sup> 针对有界批处理机上工件具有不同释放时间的最小化总完工时间问题提出了一个多项式时间近似策略, 针对无界批处理机上工件具有不同释放时间的最小化总加权完工时间问题提出了一个伪多项式时间近似策略.

综上, 目前还没有存在关于工件具有两类释放时间的两个工件集在批处理机上竞争调度问题的研究, 虽然文献 [8] 中针对无界批处理机上批不可兼容情况下工件具有恶化加工时间与不同释放时间时, 目标函数为最小化一个工件集的最大完工时间, 使得另一个工件集最大完工时间不超过给定上界问题的一般情况提出了伪多项式时间求解算法, 但本文基于问题的自身特征针对该目标函数在批无界与不可兼容情况下工件具有两类释放时间的特殊情况提

出了多项式时间求解算法, 同时针对批无界情况下最小化最大延迟与总完工时间问题分别提出了最优求解方法. 此外, 对于批有界情况下工件具有不同释放时间时, 目标函数为最小化一个工件集的最大完工时间, 使得另一个工件集的最大完工时间不超过给定上界问题, 文献 [8] 针对一般情况证明了问题的 NP-难性, 而本文进一步证明了当两个工件集具有两类释放时间的特殊情况也均为 NP-难问题, 同时分别提出了伪多项式时间的动态规划算法对问题进行求解. 这样的研究不仅为实际工业生产提供了理论指导依据, 而且从理论上丰富了多个工件集竞争调度的理论研究基础.

## 1 问题描述

设有两个工件集合  $J^A$  和  $J^B$ , 其中  $J^A = \{J_1^A, J_2^A, \dots, J_{n_A}^A\}$ ,  $J^B = \{J_1^B, J_2^B, \dots, J_{n_B}^B\}$ . 设  $n$  表示两个集合中所有的工件数, 即  $n = n_A + n_B$ . 两个集合中的工件同时竞争在一台批处理机上不可中断地加工, 批处理机能同时加工  $b$  个工件作为一批. 本文假设只有来自同一集合的工件能组成批加工, 即批不具有兼容性. 假设工件  $J_j^X$  具有加工时间  $p_j^X$ , 工期  $d_j^X$ ,  $j = 1, 2, \dots, n_X$ ,  $X \in \{A, B\}$ . 批处理机上一批  $B_x$  的加工时间定义为  $p(B_x) = \max_{J_j^X \in B_x} \{p_j^X\}$ , 工期为  $d(B_x) = \min_{J_j^X \in B_x} \{d_j^X\}$ . 每个集合的工件具有共同的释放时间  $r^X$ ,  $X \in \{A, B\}$ , 即如果工件  $J_j^X \in J^X$ , 那么它的释放时间为  $r^X$ . 由于两个集合的工件具有两类不同的释放时间, 为了便于问题的分析, 不妨假设其中较小的释放时间记为 0, 较大的释放时间记为  $r$  ( $r > 0$ ). 此外, 假设所有问题中参数均为非负整数. 给定一个加工顺序, 定义工件  $J_j^X$  的完工时间为  $C_j^X$ , 延迟为  $L_j^X = C_j^X - d_j^X$ . 本文基于两类释放时间的大小比较, 分别考虑了批处理机能力无界 ( $b > n$ ) 情况, 即批处理机能同时加工任意多个工件作为一批, 与批处理机能力有界 ( $b < n$ ) 情况, 即批处理机最多能同时加工  $b$  个工件作为一批. 主要研究的目标函数分别为最小化集合  $J^A$  中工件的最大完工时间  $C_{\max}^A = \max\{C_j^A\}$ , 最大延迟  $L_{\max}^A = \max\{L_j^A\}$ , 总完工时间  $\sum C_j^A$ , 使得集合  $J^B$  中工件的最大完工时间  $C_{\max}^B = \max\{C_j^B\}$  不超过一个给定的上界  $Q_B > 0$ . 利用文献 [2] 提出的表示法, 所研究问题表示为

$$\begin{aligned} & 1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|\gamma^A : \\ & C_{\max}^B \leq Q_B, \gamma^A \in \{C_{\max}^A, L_{\max}^A, \sum C_j^A\}; \\ & 1|p - batch, r_j^A = r^A, r_j^B = r^B, b < n|C_{\max}^A : \\ & C_{\max}^B \leq Q_B, \text{ 其中 } p - \text{batch 表示批处理机.} \end{aligned}$$

## 2 批处理机能力无界

本节将考虑批处理机能力无界的情形. 针对两个工件集各自释放时间  $r^A$  与  $r^B$  的大小关系, 分别研究了最小化工件集  $J^A$  中工件的最大完工时间、最大延迟以及总完工时间, 使得另一个工件集  $J^B$  中工件的最大完工时间不超过一个给定上界问题的时间复杂性.

2.1  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|C_{\max}^A : C_{\max}^B \leq Q_B$

首先对于求解本节的问题提出如下最优解性质.

**性质 1.** 对于问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么在最优调度中工件集  $J^A$  与  $J^B$  的工件分别形成一批生产加工.

**证明.** 由于一批的容量是无限大的, 所以将集合  $J^A$  与  $J^B$  的工件分别在各自释放时间之后形成一批生产加工, 将不会增加各自工件的最大完工时间.  $\square$

下面基于性质 1, 同时针对  $r^A$  与  $r^B$  的大小比较, 分别对两种不同情况问题给出多项式时间最优求解算法.

1)  $r^A = 0, r^B = r$

**算法 1.**

**步骤 1.** 如果  $\max\{\max\{p_j^A\}, r\} + \max\{p_j^B\} \leq Q_B$ , 那么当  $\max\{p_j^A\} \leq r$  时, 在最优调度中从 0 时刻开始加工集合  $J^A$  中工件构成的一批, 从  $r$  时刻开始加工集合  $J^B$  中工件构成的一批; 当  $\max\{p_j^A\} > r$  时, 在最优调度中从 0 时刻开始加工集合  $J^A$  中工件构成的一批, 从  $\max\{p_j^A\}$  时刻开始加工集合  $J^B$  中工件构成的一批. 如果  $\max\{\max\{p_j^A\}, r\} + \max\{p_j^B\} > Q_B$ , 那么转到步骤 2.

**步骤 2.** 如果  $\max\{p_j^A\} \leq r$ , 那么问题没有解. 如果  $\max\{p_j^A\} > r$ , 那么在最优调度中从  $r$  时刻开始加工集合  $J^B$  中工件构成的一批, 从  $r + \max\{p_j^B\}$  时刻开始加工集合  $J^A$  中工件构成的一批.

2)  $r^A = r, r^B = 0$

**算法 2.**

**步骤 1.** 如果  $r + \max\{p_j^A\} + \max\{p_j^B\} \leq Q_B$ , 那么在最优调度中从  $r$  时刻开始加工集合  $J^A$  中工件构成的一批, 从  $r + \max\{p_j^A\}$  时刻开始加工集合  $J^B$  中工件构成的一批. 如果  $r + \max\{p_j^A\} + \max\{p_j^B\} > Q_B$ , 那么转到步骤 2.

**步骤 2.** 如果  $\max\{p_j^B\} \leq Q_B$ , 那么在最优调度中从 0 时刻开始加工集合  $J^B$  中工件构成的一批, 从  $\max\{\max\{p_j^B\}, r\}$  时刻开始加工集合  $J^A$  中工件构成的一批. 如果  $\max\{p_j^B\} > Q_B$ , 那么问题无解.

**定理 1.** 算法 1 与 2 均可在  $O(n)$  时间内分别找到问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|C_{\max}^A :$

$C_{\max}^B \leq Q_B$  不同释放时间大小的最优调度.

**证明.** 由性质 1 可知, 工件集  $J^A$  与  $J^B$  中的工件分别在各自的释放时间之后构成一批加工. 当  $r^A = 0 < r^B = r$  时, 如果问题存在最优解, 那么: a) 当集合  $J^A$  中工件的最大加工时间  $\max\{p_j^A\} \leq r$  时, 所有集合  $J^A$  中工件能够从 0 时刻开始构成一批加工, 完工时间即为  $\max\{p_j^A\}$ , 此时集合  $J^B$  中的工件最早可以从  $r$  时刻开始构成一批加工, 其完工时间为  $r + \max\{p_j^B\}$ ; b) 当  $\max\{p_j^A\} > r$  时, 如果仍从 0 时刻先加工工件集  $J^A$  再加工工件集  $J^B$ , 能够满足工件集  $J^B$  的最大完工时间  $C_{\max}^B = \max\{p_j^A\} + \max\{p_j^B\} \leq Q_B$ , 则能获得问题的最优解; 反之, 如果  $\max\{p_j^A\} + \max\{p_j^B\} > Q_B$ , 则最优解中只能最早从释放时间  $r$  开始先加工工件集  $J^B$ , 再接着加工工件集  $J^A$ .

同理, 当  $r^A = r > r^B = 0$  时, 如果问题存在最优解, 那么此时工件集  $J^A$  的最早开始加工时间为  $r$ , 如果先加工工件集  $J^A$  再加工工件集  $J^B$ , 满足目标约束条件  $C_{\max}^B = r + \max\{p_j^A\} + \max\{p_j^B\} \leq Q_B$ , 则能获得问题的最优解; 反之, 如果  $r + \max\{p_j^A\} + \max\{p_j^B\} > Q_B$ , 则最优解中只能从 0 时刻开始先加工工件集  $J^B$ , 再接着加工工件集  $J^A$ .

算法 1 与 2 中均需求出工件集  $J^A$  与  $J^B$  中最大加工时间, 因此时间复杂性均为  $O(n_A) + O(n_B) = O(n)$ .  $\square$

下面给出一个实例来应用算法 1. 假设工件集合  $J^A = \{J_1^A, J_2^A, J_3^A\}$ , 集合  $J^B = \{J_1^B, J_2^B\}$ . 工件的加工时间分别为  $p_1^A = 1, p_2^A = 2, p_3^A = 3; p_1^B = 1, p_2^B = 2$ . 共同释放时间分别为  $r^A = 0, r^B = r = 1$ . 集合  $J^B$  工件的最大完工时间上界为  $Q_B = 4$ .

在算法 1 的步骤 1 中, 由于  $\max\{\max\{p_j^A\}, r\} + \max\{p_j^B\} = 5 > Q_B = 4$ , 转到步骤 2. 又由于  $\max\{p_j^A\} = 3 > r = 1$ , 所以最优调度  $\pi'$  为从  $r = 1$  时刻开始加工集合  $J^B$  中工件构成的一批, 然后再从  $r + \max\{p_j^B\} = 3$  时刻开始加工集合  $J^A$  中工件构成的一批, 最优调度如图 3 所示.

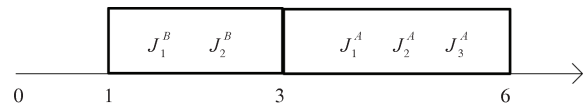


图 3 最优调度  $\pi'$

Fig. 3 The optimal schedule  $\pi'$

2.2  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$

针对问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$ , 给出一个最优解性质.

**性质 2.** 对于问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^B$  中所有工件在时刻  $r^B$  或者  $r^B$  之后形成一批生产加工, 集合  $J^A$  中所有工件按照最短加工时间 (Shortest processing time, SPT) 规则生产加工.

**证明.** 类似于性质 1 的证明, 很容易得出集合  $J^B$  中所有工件在时刻  $r^B$  或者  $r^B$  之后形成一批生产加工, 这样既不会增加集合  $J^B$  中工件的最大完工时间, 也不会增加集合  $J^A$  中每个工件的完工时间, 进而不会增加集合  $J^A$  中工件的最大延迟. 下面假设存在一个最优调度  $\pi$ , 其中批  $B_x$  与  $B_y$  是关于集合  $J^A$  中工件的任意两批, 且  $x < y$ , 工件  $J_i^A \in B_x, J_j^A \in B_y$ , 同时  $p_i^A > p_j^A$ . 将工件  $J_j^A$  从批  $B_y$  中移动到  $B_x$  中, 经过这样的转移后得到一个新的调度  $\pi'$ . 我们可以看到在  $\pi'$  中工件  $J_j^A$  的完工时间减少, 其他工件的完工时间都不会增加. 因此得到  $L_{\max}^A(\pi') \leq L_{\max}^A(\pi)$ .  $\square$

由性质 2 可知, 如果集合  $J^A$  中存在两个工件  $J_i^A$  和  $J_j^A$ , 并且满足  $p_i^A \leq p_j^A$ , 同时  $d_i^A \geq d_j^A$ , 那么我们可以将工件  $J_i^A$  转移到与工件  $J_j^A$  同一批中, 而不会增加集合  $J^A$  中工件的最大延迟. 由此, 对于问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$ , 我们可以假设集合  $J^A$  中所有工件已经按照  $p_1^A \leq p_2^A \leq \dots \leq p_{n_A}^A$  及  $d_1^A \leq d_2^A \leq \dots \leq d_{n_A}^A$  的顺序排列.

1)  $r^A = 0, r^B = r$

针对问题  $1|p - batch, r^A = 0, r^B = r, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$ , 由性质 2 设  $X_B$  为集合  $J^B$  所有工件形成的一批. 设  $S_1$  为在  $X_B$  之前加工的集合  $J^A$  中工件的子集,  $S_2$  为在  $X_B$  之后加工的集合  $J^A$  中余下工件的集合, 即问题  $1|p - batch, r^A = 0, r^B = r, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  具有最优调度形式  $(S_1, X_B, S_2)$ , 其中  $S_1$  与  $S_2$  中任何一个都可以是空集.

对于调度形式  $(S_1, X_B, S_2)$ , 集合  $J^B$  的最大完工时间  $C_{\max}^B$  由集合  $S_1$  中工件个数  $l$  及  $S_1$  中最后一批工件的完工时间  $t$  所决定,  $0 \leq l \leq n_A$ . 因此如果给定集合  $S_1$  中工件个数  $l$ , 问题就分解为在零时刻分别调度子集  $S_1$  与  $S_2$  中的工件, 分别使得  $S_1$  中工件对于每个  $t$  的最大延迟最小化以及  $S_2$  中工件的最大延迟最小化, 其中  $S_1 = \{J_1^A, J_2^A, \dots, J_l^A\}, S_2 = \{J_{l+1}^A, J_{l+2}^A, \dots, J_{n_A}^A\}$ . 因此, 调度形式  $(S_1, X_B, S_2)$  的最优调度对应的最小化最大延迟值为

$$\max\{f(l, t), C_{\max}^B + L_{\max}^A(S_2)\} \quad (1)$$

使得  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$  成立, 其中  $f(l, t)$  为子集  $S_1$  中调度  $l$  个工件当最后一批工件的完工时间为  $t$  时的最大延迟最小值.

在式 (1) 中最大延迟最小值  $L_{\max}^A(S_2)$ ,  $S_2 = \{J_{l+1}^A, J_{l+2}^A, \dots, J_{n_A}^A\}, l = 0, 1, \dots, n_A - 1$  所对应的最优调度可以利用 Brucker 等<sup>[15]</sup> 求解问题  $1|p - batch, b > n|L_{\max}$  的动态规划算法在  $O(n_A^2)$  时间内获得. 而最大延迟最小值  $f(l, t), 0 \leq l \leq n_A$  所对应的最优调度可以通过下面构造的动态规划算法获得.

**算法 DP1.**

$$\text{初始条件: } f(0, t) = \begin{cases} -\infty, & \text{若 } t = 0 \\ +\infty, & \text{否则} \end{cases}$$

对于每个  $l = 1, \dots, n_A$  对应在不同的最后一批完工时间  $t$  下,  $t = p_l^A, \dots, \sum_{j=1}^l p_j^A$  有如下递归关系:  $f(l, t) = \min_{0 \leq h \leq l-1} \{\max\{f(h, t - p_l^A), t - d_{h+1}\}\}$ , 如果满足  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 其中当前调度中最后一批的工件为  $\{J_{h+1}^A, \dots, J_l^A\}$  对于任意的  $h, 0 \leq h < l$ .

此时对于每个  $l$ , 对应在不同的  $t$  值下, 所有的值  $f(l, t)$  以及对应的调度均可在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A)$  时间内找到.

综上, 针对不同的  $l$  以及对应不同的  $t$  值, 在所有得到的  $f(l, t)$  中, 找到满足条件  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 同时最大延迟值  $\max\{f(l, t), C_{\max}^B + L_{\max}^A(S_2)\}$  最小的所对应的调度即为问题的最优调度.

**定理 2.** 问题  $1|p - batch, r^A = 0, r^B = r, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  能够在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$  时间内求得最优解.

**证明.** 由性质 2, 所研究问题的最优解具有调度形式  $(S_1, X_B, S_2)$ , 并且所有集合  $J^A$  中的工件按照  $p_1^A \leq p_2^A \leq \dots \leq p_{n_A}^A$  同时  $d_1^A \leq d_2^A \leq \dots \leq d_{n_A}^A$  的顺序排列, 因此最优调度只需确定  $S_1$  中  $l$  个工件对应不同最大完工时间的最大延迟最小值, 以及  $S_2$  中  $n_A - l$  个工件的最大延迟最小值. 为了确定  $S_2$  中工件最大延迟最小值, 我们可利用文献 [15] 中求解问题  $1|p - batch, b > n|L_{\max}$  的最优算法从零时刻开始调度  $S_2$  中工件, 其时间复杂度为  $O(n_A^2)$ . 由于  $S_2$  中工件的实际开始加工时间为集合  $X_B$  的完工时间  $C_{\max}^B$ , 而  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\}$ , 受  $S_1$  中最后一批工件完工时间  $t$  的影响, 因此, 对于  $S_1$  中每个给定的工件个数  $l$  对应不同最大完工时间  $t$  构造动态规划算法 DP1, 得出  $S_1$  中  $l$  个工件在不同最大完工时间  $t$  下对应的最大延迟最小值  $f(l, t)$ . 进而对不同的  $l$  对应不同的  $t$ , 找到满足条件  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 同时最大延迟  $\max\{f(l, t), C_{\max}^B + L_{\max}^A(S_2)\}$  最小的所对应的调度即为问题的最优调度.

对于算法 DP1, 由性质 2 可知, 从零时刻开始第一批工件  $\{J_1^A, J_2^A, \dots, J_l^A\}$  按照 SPT 规则

以及工期不减的顺序在机器上加工, 对应最大完工时间  $t$  时的最大延迟最小值为  $f(l, t)$ , 当前调度中最后一批工件  $\{J_{h+1}^A, \dots, J_l^A\}$  对应的最大延迟为  $t - d_{h+1}$ . 因此, 目标函数值为  $f(l, t) = \min_{0 \leq h \leq l-1} \{\max\{f(h, t - p_h^A), t - d_{h+1}\}\}$ . 为了使得调度形式  $(S_1, X_B, S_2)$  具有可行性, 需满足条件  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ . 此外, 算法 DP1 中还满足  $0 \leq h < l \leq n_A, 0 \leq t \leq \sum_{j=1}^{n_A} p_j^A$ , 所以 DP1 的时间复杂度为  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$ .

综上, 问题  $1|p - batch, r^A = 0, r^B = r, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  能够在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$  时间内求得最优解.  $\square$

2)  $r^A = r, r^B = 0$

如果问题  $1|p - batch, r^A = r, r^B = 0, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  存在可行调度, 那么有如下三种情况:

a) 如果  $\max\{p_j^B\} \leq r$ , 那么在最优调度中集合  $J^B$  的所有工件从零时刻开始组成一批生产加工, 集合  $J^A$  的所有工件从  $r$  时刻开始利用文献 [15] 中关于求解问题  $1|p - batch, b > n|L_{\max}$  的动态规划算法可在  $O(n_A^2)$  时间内求得最优解.

b) 如果  $r < \max\{p_j^B\} \leq Q_B - r - t$ , 其中  $0 < t \leq \sum_{j=1}^{n_A} p_j^A$ , 由性质 2 可知, 问题  $1|p - batch, r^A = r, r^B = 0, b > n|L_{\max}^A : C_{\max}^B \leq Q_B$  仍具有最优调度形式  $(S_1, X_B, S_2)$ , 其中集合  $S_1$  与  $S_2$  中工件对应的调度与上面  $r^A = 0, r^B = r$  时情况相同. 而组合后调度形式  $(S_1, X_B, S_2)$  的最优调度对应的最小化最大延迟值为

$$r + \max\{f(l, t), t + \max\{p_j^B\} + L_{\max}^A(S_2)\} \quad (2)$$

其中,  $0 < t \leq \sum_{j=1}^{n_A} p_j^A, r + t + \max\{p_j^B\} \leq Q_B$ .

因此, 该情况能够在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$  时间内求得问题的最优解.

c) 如果  $\max\{p_j^B\} \geq Q_B - r$ , 那么在最优调度中集合  $J^B$  的所有工件从零时刻开始组成一批生产加工, 集合  $J^A$  的所有工件从  $\max\{p_j^B\}$  时刻开始利用文献 [15] 中关于求解问题  $1|p - batch, b > n|L_{\max}$  的动态规划算法可在  $O(n_A^2)$  时间内求得最优解.

2.3  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$

类似于性质 2 的证明, 对于问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$ , 给出如下最优解性质.

**性质 3.** 对于问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^B$  中所有工件在时刻  $r^B$  或者  $r^B$  之后形成一批生产加工, 集合  $J^A$  中所有工件按照 SPT 规则生产加工.

1)  $r^A = 0, r^B = r$

针对问题  $1|p - batch, r^A = 0, r^B = r, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$ , 由性质 3, 设  $X_B$  为集合  $J^B$  中所有工件形成的一批, 同时有  $p_1^A \leq p_2^A \leq \dots \leq p_{n_A}^A$ . 设  $S_1$  为在  $X_B$  之前加工生产的集合  $J^A$  中工件的子集,  $S_2$  为在  $X_B$  之后加工生产的余下集合  $J^A$  中工件的集合, 即问题  $1|p - batch, r^A = 0, r^B = r, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$  具有最优调度形式  $(S_1, X_B, S_2)$ , 其中  $S_1$  与  $S_2$  中任何一个都可以是空集.

类似于第 2.2 节, 对于调度形式  $(S_1, X_B, S_2)$ , 集合  $J^B$  的最大完工时间  $C_{\max}^B$  由集合  $S_1$  中工件个数  $l$  及  $S_1$  中最后一批工件的完工时间  $t$  所决定. 因此如果给定集合  $S_1$  中工件个数  $l$ , 问题就分解为在零时刻分别调度子集  $S_1$  与  $S_2$  中的工件, 分别使得  $S_1$  中工件对于每个  $t$  的总完工时间最小化以及  $S_2$  中工件的总完工时间最小化, 其中  $S_1 = \{J_1^A, J_2^A, \dots, J_l^A\}$ ,  $S_2 = \{J_{l+1}^A, J_{l+2}^A, \dots, J_{n_A}^A\}$ . 因此, 对于问题  $1|p - batch, r^A = 0, r^B = r, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$ , 最优调度对应的最小化总完工时间为

$$f(l, t) + (n_A - l)C_{\max}^B + \sum_{j=l+1}^{n_A} C_j^A(S_2) \quad (3)$$

满足  $0 \leq l \leq n_A, C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 其中  $f(l, t)$  为子集  $S_1$  中调度  $l$  个工件当最后一批工件的完工时间为  $t$  时的最小总完工时间.

在式 (3) 中, 集合  $S_2$  中工件最小总完工时间所对应的最优调度可利用文献 [15] 提出的求解问题  $1|p - batch, b > n|\sum C_j$  的动态规划算法在  $O(n_A \log n_A)$  时间内获得. 而最小总完工时间  $f(l, t)$  可以通过下面的动态规划求出.

**算法 DP2.**

$$\text{初始条件: } f(0, t) = \begin{cases} 0, & \text{若 } t = 0 \\ \infty, & \text{否则} \end{cases}$$

对于每个  $l = 1, 2, \dots, n_A$  对应在不同的最后一批完工时间  $t$  下,  $t = p_l^A, \dots, \sum_{j=1}^l p_j^A$  有如下递归关系:  $f(l, t) = \min_{0 \leq h < l-1} \{f(h, t - p_l^A) + (l - h)t\}$ , 如果满足  $\max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 其中当前调度中最后一批的工件为  $\{J_{h+1}^A, \dots, J_l^A\}$ , 目标函数的增加值为该批的总完工时间  $(l - h)t$ .

对于每个  $l$ , 对应在不同的  $t$  值下, 所有的值  $f(l, t)$  以及对应的调度均可在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A)$  时间内找到.

综上, 针对不同的  $l$  以及对应不同的  $t$  值, 在所有得到的  $f(l, t)$  中, 找到满足条件  $C_{\max}^B = \max\{t, r\} + \max\{p_j^B\} \leq Q_B$ , 同时总完工时间  $f(l, t) + (n_A - l)C_{\max}^B + \sum_{j=l+1}^{n_A} C_j^A(S_2)$  最小的所对应的调度即为问题的最优调度.

**定理 3.** 问题  $1|p - batch, r^A = 0, r^B = r, b >$

$n|\sum C_j^A : C_{\max}^B \leq Q_B$  能够在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$  时间内求得最优解。

定理 3 的证明类似于定理 2 的证明, 这里不再详细赘述。

2)  $r^A = r, r^B = 0$

如果问题  $1|p - batch, r^A = r, r^B = 0, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$  存在可行调度, 那么有如下三种情况:

a) 如果  $\max\{p_j^B\} \leq r$ , 那么在最优调度中集合  $J^B$  的所有工件从零时刻开始组成一批生产加工, 集合  $J^A$  的所有工件从  $r$  时刻开始利用文献 [15] 中关于求解问题  $1|p - batch, b > n|\sum C_j$  的动态规划算法在  $O(n_A \log n_A)$  时间内求得最优解。

b) 如果  $r < \max\{p_j^B\} \leq Q_B - r - t$ , 其中  $0 < t \leq \sum_{j=1}^{n_A} p_j^A$ , 由性质 3 可知, 问题  $1|p - batch, r^A = r, r^B = 0, b > n|\sum C_j^A : C_{\max}^B \leq Q_B$  仍具有最优调度形式  $(S_1, X_B, S_2)$ , 其中集合  $S_1$  与  $S_2$  中工件对应的调度与上面  $r^A = 0, r^B = r$  时情况相同. 而组合后调度形式  $(S_1, X_B, S_2)$  的最优调度对应的最小化总完工时间值为

$$f(l, t) + (t + \max\{p_j^B\})(n_A - l) + \sum_{j=l+1}^{n_A} C_j^A(S_2) + n_A r \quad (4)$$

其中,  $0 < t \leq \sum_{j=1}^{n_A} p_j^A, r + t + \max\{p_j^B\} \leq Q_B$ .

因此, 该情况能够在  $O(n_A^2 \sum_{j=1}^{n_A} p_j^A + n_B)$  时间内求得问题的最优解。

c) 如果  $\max\{p_j^B\} \geq Q_B - r$ , 那么在最优调度中集合  $J^B$  的所有工件从零时刻开始组成一批生产加工, 集合  $J^A$  的所有工件从  $\max\{p_j^B\}$  时刻开始利用文献 [15] 中关于求解问题  $1|p - batch, b > n|\sum C_j$  的动态规划算法在  $O(n_A \log n_A)$  时间内求得最优解。

### 3 批处理机能力有界

本节只考虑批处理机能力有界的情形下, 目标函数为最小化集合  $J^A$  工件的最大完工时间使得集合  $J^B$  工件的最大完工时间不超过给定上界的复杂性. 针对两个工件集各自共同释放时间  $r^A$  与  $r^B$  的大小关系, 分别证明了两种情况下问题  $1|p - batch, r_j^A = r^A, r_j^B = r^B, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  均为 NP-难问题, 同时针对这两种情况分别设计了伪多项式时间算法进行求解。

1)  $r^A = 0, r^B = r$

定理 4. 问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  是一般意义 NP-难的。

证明. 我们将通过 NP-难的划分问题归约来证

明问题的 NP-难性. 划分问题 (Partition problem) 定义如下:

给定  $m$  个正整数  $a_1, a_2, \dots, a_m$ , 并且满足  $\sum_{j=1}^m a_j = 2H$ , 是否存在一个划分将下标集合  $I = \{1, 2, \dots, m\}$  分成两个不相交的子集  $I_1$  和  $I \setminus I_1$ , 使得  $\sum_{j \in I_1} a_j = \sum_{j \in I \setminus I_1} a_j = H$ ?

给定划分问题的一个实例, 构造问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  的判定问题的一个实例如下:

$$n_B = 1, r^B = r = H, p_1^B = H, Q_B = 2H;$$

批处理机一批的能力为  $b = 2$ ;

$$n_A = 2m, r^A = 0, p_j^A = p_{j+m}^A = a_j, j = 1, 2, \dots, m;$$

集合  $J^A$  目标的阈值为  $Q_A = 3H$ .

接下来证明对于构造的问题的实例, 当且仅当划分问题有解时, 能够找到一个调度满足  $C_{\max}^A \leq Q_A, C_{\max}^B \leq Q_B$ .

$\Rightarrow$  如果划分问题有解, 即存在子集  $I_1 \subset I$ , 使得  $\sum_{j \in I_1} a_j = \sum_{j \in I \setminus I_1} a_j = H$  成立. 对于上面问题构造的实例, 能够找到一个调度, 满足  $C_{\max}^A \leq Q_A, C_{\max}^B \leq Q_B$ , 调度如下: 工件  $J_1^B$  在  $r^B = H$  时刻开始调度, 具有相同加工时间的工件  $J_j^A$  与  $J_{j+m}^A$  组成一批加工, 即批  $B_j = \{J_j^A, J_{j+m}^A\}, j \in I$ , 其中对应下标  $j \in I_1$  的批从 0 时刻开始加工, 下标  $j \in I \setminus I_1$  的批从  $2H$  时刻开始加工. 容易验证  $C_{\max}^A = 3H = Q_A, C_{\max}^B = 2H = Q_B$ .

$\Leftarrow$  如果对于问题构造的实例存在一个调度满足  $C_{\max}^A \leq Q_A, C_{\max}^B \leq Q_B$ , 那么容易证明集合  $J^B$  唯一的工件  $J_1^B$  一定在  $r^B = H$  时刻开始调度. 此外, 由于  $\sum_{j=1}^{2m} p_j^A / b = 2H$ , 所以调度从 0 到  $3H$  时刻之间没有空闲时间, 集合  $J^A$  工件构成的每一批都是满批, 并且每一批的两个工件一定具有相同的加工时间. 记批  $B_j = \{J_j^A, J_{j+m}^A\}, j \in I$ , 因此在  $J_1^B$  之前加工的所有批的加工时间和为  $\sum_{j \in I_1} a_j = H$ , 之后加工的所有批的加工时间和为  $\sum_{j \in I \setminus I_1} a_j = H$ . 因此划分问题有解.  $\square$

下面给出求解问题的两个最优解性质。

性质 4. 对于问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^B$  中所有工件在时刻  $r$  或者  $r$  之后连续生产加工, 并且加工批满足满批最大加工时间 (Full batch largest processing time, FBLPT) 规则。

证明. 由于集合  $J^B$  中所有工件的释放时间为  $r$ , 所以集合  $J^B$  中工件只能在时刻  $r$  或者  $r$  之后加工生产. 在一个可行调度中, 将工件交换位置使得集合  $J^B$  中除最后一个工件外其他的工件向右移动到集合  $J^B$  中最后一个工件前与之连续加工, 再结合文

献 [15] 中提出的求解问题  $1|p - batch, b < n|C_{\max}^A$  的调度方法, 得到的调度中集合  $J^B$  的工件最大完工时间不会增加, 同时集合  $J^A$  的每个工件的完工时间也不会增加, 进而最大完工时间不会增加.  $\square$

设  $X_B$  为集合  $J^B$  工件按照 FBLPT 规则连续加工形成的集合.

**性质 5.** 对于问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^A$  中所有工件可以按照 FBLPT 规则产生所有加工批.

**证明.** 通过工件交换与工件转移, 能够证明性质的结论.  $\square$

由性质 4 和 5, 设集合  $J^A$  与  $J^B$  中的工件分别按照 FBLPT 规则获得所有加工批, 所需时间为  $O(n \log n)$ , 设批总长度为  $T_X, X \in \{A, B\}$ , 即  $T_X = \sum_{i=0}^{\lfloor \frac{n_X}{b} \rfloor} p_{ib+1}^X$ , 其中  $\lfloor x \rfloor$  表示不大于  $x$  的最大整数. 下面针对问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  提出如下算法进行求解.

### 算法 3.

**步骤 1.** 如果  $\max\{T_A, r\} + T_B \leq Q_B$ , 那么在最优调度中从 0 时刻开始按照 FBLPT 规则连续加工集合  $J^A$  的工件, 从  $\max\{T_A, r\}$  时刻开始按照 FBLPT 规则连续加工集合  $J^B$  的工件. 如果  $\max\{T_A, r\} + T_B > Q_B$ , 那么转到步骤 2.

**步骤 2.** 如果  $T_A \leq r$ , 那么问题没有解. 如果  $T_A > r$ , 那么问题的最优调度具有形式  $(S_1, X_B, S_2)$ , 其中  $S_1$  与  $S_2$  分别为在  $X_B$  前后加工生产的集合  $J^A$  工件的子集,  $S_1$  可能为空集, 而  $S_2$  一定不是空集. 针对这一情况, 设计如下伪多项式时间动态规划算法对问题进行求解.

### 算法 DP3.

由性质 5, 我们可以将集合  $J^A$  中所有工件按照 FBLPT 规则产生的每一个批看作一个工件. 因此, 此时关于集合  $J^A$  工件的批生产调度问题可以转化为单机生产调度问题, 即批处理机的能力为  $b = 1$ . 假设集合  $J^A$  的所有工件按照加工时间不增的顺序排列如下:  $p_1^A \geq p_2^A \geq \dots \geq p_{n_A}^A$ . 定义  $f(s)$  为调度  $n_A$  个集合  $J^A$  的工件与全部  $J^B$  工件后集合  $J^A$  工件最大完工时间的最小值, 其中  $s$  是  $X_B$  的开始加工时间, 满足  $r \leq s < r + P_{\max}^A$ , 其中,  $P_{\max}^A = \max\{p_j^A\}$ , 因为在集合  $S_1$  中至多一个工件在  $r$  之后加工完. 对于一个给定的  $s$ , 为了求出  $f(s)$  的最小值, 进一步定义  $F(h, t)$  为调度部分工件  $J_1^A, J_2^A, \dots, J_h^A$  时集合  $J^A$  工件最大完工时间的最小值, 其中  $t$  是集合  $S_1$  中最后一个工件的完工时间,  $t \leq s$ .

定义初始条件为

$$F(0, t) = \begin{cases} s + T_B, & \text{若 } t = 0 \\ +\infty, & \text{其他} \end{cases}$$

为了获得  $F(h, t)$  的最优值, 我们列举工件  $J_h^A$  的可能加工位置, 即工件  $J_h^A$  可能是集合  $S_1$  中最后一个加工的工件, 也可能是集合  $S_2$  中最后一个加工的工件, 于是有下面的递归关系:

$$F(h, t) = \min \begin{cases} F(h-1, t - p_h^A), & \text{若 } J_h^A \in S_1 \\ F(h-1, t) + p_h^A, & \text{若 } J_h^A \in S_2 \end{cases}$$

因此, 对于  $X_B$  的每个开始加工时间  $s, r \leq s < r + P_{\max}^A$ ,  $f(s)$  的最小值能够获得, 具有如下形式:

$$f(s) = \min\{F(n_A, t) : t \leq s\}$$

进而, 问题的最优解值为  $\min\{f(s) : r \leq s < r + P_{\max}^A\}$ .

**定理 5.** 算法 3 可以找到问题  $1|p - batch, r^A = 0, r^B = r, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  的最优调度, 其中算法 DP3 的时间复杂度为  $O(n_A(r + P_{\max}^A)^2)$ , 这里  $P_{\max}^A = \max\{p_j^A\}$ .

**证明.** 在算法 3 中, 如果问题存在最优解, 那么当  $\max\{T_A, r\} + T_B \leq Q_B$  时, 很显然利用性质 4 和 5 在最优解中从 0 时刻开始按照 FBLPT 规则连续加工集合  $J^A$  中工件, 从  $\max\{T_A, r\}$  时刻开始按照 FBLPT 规则连续加工集合  $J^B$  工件; 当  $T_A + T_B > Q_B$  时, 具有最优调度形式  $(S_1, X_B, S_2)$ . 由性质 5 可知, 此时可以将关于集合  $J^A$  工件的批生产调度看成单机生产调度问题. 对于一个给定的  $X_B$  的开始加工时间  $s$ , 工件  $J_h^A$  的完工时间  $F(h, t)$  有两种情况: a) 当  $J_h^A$  在集合  $S_1$  中最后一个加工时,  $F(h, t) = F(h-1, t - p_h^A)$ ; b) 当  $J_h^A$  在集合  $S_2$  中最后一个加工时,  $F(h, t) = F(h-1, t) + p_h^A$ . 最后对于每个  $s$  所获得的集合  $J^A$  全部工件的最大完工时间最小值  $f(s) = \min\{F(n_A, t)\}$ , 再求  $\min\{f(s)\}$  即为问题的最优解. 又由于  $r \leq s < r + P_{\max}^A$ , 所以  $s$  的大小界限为  $O(r + P_{\max}^A)$ . 另外  $0 \leq h \leq n_A, t$  是集合  $S_1$  中最后一个工件的完工时间, 满足  $t \leq s$ . 因此, 算法 DP3 的时间复杂度为  $O(n_A(r + P_{\max}^A)^2)$ .  $\square$

$$2) r^A = r, r^B = 0$$

**定理 6.** 问题  $1|p - batch, r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  是一般意义 NP-难的.

**证明.** 类似于定理 4 的证明, 通过 NP-难的划分问题归约来证明该问题的 NP-难性. 构造问题  $1|p - batch, r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  的判定问题的一个实例如下:

$$n_B = 2m, r^B = 0, p_j^B = p_{j+m}^B = a_j, j = 1, 2, \dots, m, Q_B = 3H;$$

批处理机一批的能力为  $b = 2$ ;

$$n_A = 1, r^A = r = H, p_1^A = H;$$

集合  $J^A$  目标的阈值为  $Q_A = 2H$ .

在此我们忽略详细的证明过程.  $\square$

针对问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 首先设计一个动态规划算法计算出集合  $J^B$  所有工件最大完工时间的最小值, 然后在所有这些最小值中找到满足集合  $J^B$  目标函数约束条件下集合  $J^A$  工件的最小开始加工时间. 类似于性质 4 的证明, 提出如下的最优解性质:

**性质 6.** 对于问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^A$  中所有工件在时刻  $r$  或者  $r$  之后连续生产加工, 并且加工批满足 FBLPT 规则.

设  $X_A$  为集合  $J^A$  工件按照 FBLPT 规则连续加工形成的集合, 类似于性质 5 有下面的最优解性质:

**性质 7.** 对于问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$ , 如果存在一个最优调度, 那么集合  $J^B$  中所有工件可以按照 FBLPT 规则产生所有加工批.

类似于情况  $r^A = 0, r^B = r$ , 设  $T_X, X \in \{A, B\}$  为集合  $J^X$  所有工件按照 FBLPT 规则获得所有加工批的批总长度. 在问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  中, 假设  $r < T_B < Q_B$ ; 否则, 当  $T_B \leq r$  或  $T_B = Q_B$  时, 如果存在一个最优调度, 那么可以将集合  $J^B$  所有工件从零时刻开始按照 FBLPT 规则连续生产加工, 集合  $J^A$  所有工件对应应在  $r$  或  $Q_B$  时刻按照 FBLPT 规则连续生产加工; 当  $T_B > Q_B$  时, 问题没有解. 因此, 问题具有调度形式  $(S_1, X_A, S_2)$ , 其中  $S_1$  与  $S_2$  分别为在  $X_A$  前后加工生产的集合  $J^B$  工件的子集合,  $S_1$  可能为空集, 而  $S_2$  一定不是空集.

根据以上的分析, 首先给出如下动态规划算法求出问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  中集合  $J^B$  所有工件最大完工时间的最小值.

#### 算法 DP4.

由性质 7, 我们可以将集合  $J^B$  中所有工件按照 FBLPT 规则产生的每一个批看作一个工件. 因此, 此时关于集合  $J^B$  工件的批生产调度问题可以转化为单机生产调度问题, 即批处理机的能力为  $b = 1$ . 假设集合  $J^B$  的所有工件按照加工时间不增的顺序排列如下:  $p_1^B \geq p_2^B \geq \dots \geq p_{n_B}^B$ . 定义  $f(s)$  为调度  $n_B$  个集合  $J^B$  工件与全部集合  $J^A$  工件后集合  $J^B$  工件最大完工时间的最小值, 其中  $s$  是  $X_A$  的开始加工时间, 满足  $r \leq s < r + P_{\max}^B$ . 对于一个给定的  $s$ , 为了求出  $f(s)$  的最小值, 进一步定义  $F(k, t)$  为调度部分工件  $J_1^B, J_2^B, \dots, J_k^B$  时集合  $J^B$  工件最大完工时间的最小值, 其中  $t$  是  $S_1$  中最后一个工件的

完工时间,  $t \leq s$ .

定义初始条件为

$$F(0, t) = \begin{cases} s + T_A, & \text{若 } t = 0 \\ +\infty, & \text{其他} \end{cases}$$

为了获得  $F(k, t)$  的最优值, 同样列举工件  $J_k^B$  的所有可能加工位置, 即工件  $J_k^B$  可能是集合  $S_1$  中最后一个加工的工件, 也可能是集合  $S_2$  中最后一个加工的工件, 于是有下面的递归关系:

$$F(k, t) = \min \begin{cases} F(k-1, t - p_k^B), & \text{若 } J_k^B \in S_1 \\ F(k-1, t) + p_k^B, & \text{若 } J_k^B \in S_2 \end{cases}$$

因此, 对于  $X_A$  的每个开始加工时间  $s, r \leq s < r + P_{\max}^B$ ,  $f(s)$  的最小值能够获得, 具有如下形式:

$$f(s) = \min\{F(n_B, t) : t \leq s\}$$

为了求出满足  $C_{\max}^B \leq Q_B$  条件下集合  $J^A$  工件最大完工时间的最小值, 我们在上面得到的所有  $f(s)$  中找到满足条件  $f(s) \leq Q_B$  的最小的  $s$ , 该调度即为问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  的最优调度.

**定理 7.** 算法 DP4 可以在  $O(n_B(r + P_{\max}^B)^2)$  时间内找到问题 1| $p$ -batch,  $r^A = r, r^B = 0, b < n|C_{\max}^A : C_{\max}^B \leq Q_B$  的最优调度, 其中  $P_{\max}^B = \max\{p_j^B\}$ .

**证明.** 算法 DP4 的最优性证明类似于定理 5 中最优性的证明. 由于  $r \leq s < r + P_{\max}^B$ , 所以  $s$  的大小界限为  $O(r + P_{\max}^B)$ . 另外  $0 \leq k \leq n_B, t$  是集合  $S_1$  中最后一个工件的最大完工时间, 满足  $t \leq s$ . 因此, 算法 DP4 的时间复杂度为  $O(n_B(r + P_{\max}^B)^2)$ .  $\square$

## 4 结论

本文研究了带有两类释放时间的两个工件集在一台批处理机上竞争生产加工的有约束优化调度问题. 针对两类释放时间的大小, 分别考虑了批无界与批有界情况下一些问题的时间复杂性. 当批无界时, 在满足一个工件集最大完工时间不超过一个给定上界的条件下, 分别针对最小化另一个工件集的最大完工时间、最大延迟以及总完工时间问题提出了最优解性质, 设计了多项式或伪多项式时间的最优求解算法. 当批有界时, 对于最小化一个工件集的最大完工时间使得另一个工件集最大完工时间不超过一个给定上界问题, 运用复杂性证明出问题为一般意义 NP-难问题, 同时给出了伪多项式时间动态规划算法进行求解. 批处理机上工件带有两类释放时间的两个工件集竞争调度问题的研究, 不仅为实际工业生产提高了机器生产效率、降低能耗、满足不同顾客需求, 也为多个工件集竞争调度问题的研究奠定了理论基础. 未来可以进一步研究此类问题中

NP-难问题的启发式算法, 并作出算法的性能分析, 也可以对问题的其他目标函数进行研究。

## References

- 1 Baker K R, Smith J C. A multiple-criterion model for machine scheduling. *Journal of Scheduling*, 2003, **6**(1): 7–16
- 2 Agnetis A, Mirchandani P B, Pacciarelli D, Pacifici A. Scheduling problems with two competing agents. *Operations Research*, 2004, **52**(2): 229–242
- 3 Perez-Gonzalez P, Framinan J M. A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: multi-agent scheduling problems. *European Journal of Operational Research*, 2014, **235**(1): 1–16
- 4 Agnetis A, Billaut J C, Gawiejnowicz S, Pacciarelli D, Soukhal A. *Multiagent Scheduling: Models and Algorithms*. Heidelberg: Springer, 2014.
- 5 Li S S, Yuan J J. Unbounded parallel-batching scheduling with two competitive agents. *Journal of Scheduling*, 2012, **15**(5): 629–640
- 6 Fan B Q, Cheng T C E, Li S S, Feng Q. Bounded parallel-batching scheduling with two competing agents. *Journal of Scheduling*, 2013, **16**(3): 261–271
- 7 Wang J Q, Fan G Q, Zhang Y Q, Zhang C W, Leung J Y T. Two-agent scheduling on a single parallel-batching machine with equal processing time and non-identical job sizes. *European Journal of Operational Research*, 2017, **258**(2): 478–490
- 8 Tang L X, Zhao X L, Liu J Y, Leung J Y T. Competitive two-agent scheduling with deteriorating jobs on a single parallel-batching machine. *European Journal of Operational Research*, 2017, **263**(2): 401–411
- 9 Yin Y Q, Wu W H, Cheng S R, Wu C C. An investigation on a two-agent single-machine scheduling problem with unequal release dates. *Computers & Operations Research*, 2012, **39**(12): 3062–3073
- 10 Lee W C, Chung Y H, Hu M C. Genetic algorithms for a two-agent single-machine problem with release time. *Applied Soft Computing*, 2012, **12**(11): 3580–3589
- 11 Wu C C, Wu W H, Chen J C, Yin Y Q, Wu W H. A study of the single-machine two-agent scheduling problem with release times. *Applied Soft Computing*, 2013, **13**(2): 998–1006
- 12 Cheng T C E, Chung Y H, Liao S C, Lee W C. Two-agent single-machine scheduling with release times to minimize the total weighted completion time. *Computers & Operations Research*, 2013, **40**(1): 353–361
- 13 Dover O, Shabtay D. Single machine scheduling with two competing agents, arbitrary release dates and unit processing times. *Annals of Operations Research*, 2016, **238**(1–2): 145–178
- 14 Potts C N, Kovalyov M Y. Scheduling with batching: a review. *European Journal of Operational Research*, 2000, **120**(2): 228–249
- 15 Brucker P, Gladky A, Hoogeveen H, Kovalyov M Y, Potts C N, Tautenhahn T, et al. Scheduling a batching machine. *Journal of Scheduling*, 1998, **1**(1): 31–54
- 16 Lee C Y. Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research*, 1999, **37**(1): 219–236
- 17 Liu Z H, Yu W C. Scheduling one batch processor subject to job release dates. *Discrete Applied Mathematics*, 2000, **105**(1–3): 129–136
- 18 Liu Z H, Cheng T C E. Approximation schemes for minimizing total (weighted) completion time with release dates on a batch machine. *Theoretical Computer Science*, 2005, **347**(1–2): 288–298

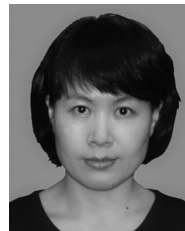


**赵晓丽** 博士, 沈阳航空航天大学理学院讲师. 主要研究方向为生产调度与组合优化. 本文通信作者.

E-mail: zhaoxiaoli824@163.com

(**ZHAO Xiao-Li** Ph.D., lecturer at the School of Science, Shenyang Aerospace University. Her research interest covers production scheduling and

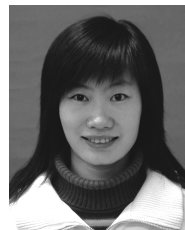
combinational optimization. Corresponding author of this paper.)



**宫华** 博士, 沈阳理工大学理学院教授. 主要研究方向为生产调度, 物流优化.

E-mail: gonghua1018@sina.com

(**GONG Hua** Ph.D., professor at the School of Science, Shenyang Ligong University. Her research interest covers production scheduling and logistics optimization.)



**车平** 博士, 东北大学理学院数学系讲师. 主要研究方向为发电调度, 生产计划, 生产调度.

E-mail: cheping@mail.neu.edu.cn

(**CHE Ping** Ph.D., lecturer in the Department of Mathematics, College of Sciences, Northeastern University. Her research interest covers generation

scheduling, production planning, and production scheduling.)