带有线性恶化工件和释放时间的两个代理单机调度问题

赵晓丽 [1] 唐立新 [1]

摘 要 研究了带有简单线性恶化工件和释放时间的两个代理单机调度问题. 所有工件在一台机器上加工, 每个代理有各自依赖于自己工件的优化目标. 针对工件释放时间相同与不同两种情况, 研究了有约束的优化模型, 即找到调度最小化一个代理的目标函数而使得另一个代理的目标函数不超过一个给定的上界. 当工件具有相同的释放时间, 我们主要考虑的目标函数有: 总加权完工时间和总加权拖期工件数. 当工件具有不同释放时间, 我们考虑的目标函数有: 最大完工时间、总完工时间以及拖期工件数. 对于每一个问题, 我们分析了问题的计算复杂性. 此外, 对于 NP 难问题的一些特殊情况本文分析了最优解性质, 基于这些性质给出了最优算法.

关键词 调度, 两个代理, 恶化工件, 释放时间, 单机

引用格式 赵晓丽, 唐立新. 带有线性恶化工件和释放时间的两个代理单机调度问题. 自动化学报, 2015, **41**(1): 104−112

DOI 10.16383/j.aas.2015.c140169

Two-agent Scheduling with Linear-deteriorating Jobs and Release Dates on a Single Machine

ZHAO Xiao-Li [1] TANG Li-Xin [1]

Abstract In this paper, we investigate the two-agent single-machine scheduling problem with simple linear-deteriorating jobs and release dates. All the jobs are processed on a common machine, and each agent has respective criterion depending on its own jobs to optimize. In view of identical or different job release dates, the constrained optimization model is studied, which is to schedule the jobs such that the objective of one agent is minimized while the objective of the other agent is less than a given upper bound. For the jobs with identical release dates, the objectives we consider in this paper are total weighted completion times and total weighted number of tardy jobs. For the jobs with distinct release dates, the objectives we consider are makespan, total completion times, and number of tardy jobs. For each problem, we analyze the computational complexity. Moreover, for several special cases of NP-hard problems, we present optimal properties and provide optimal algorithms on the basis of these properties.

Key words Scheduling, two-agent, deteriorating jobs, release date, single machine

Citation Zhao Xiao-Li, Tang Li-Xin. Two-agent scheduling with linear-deteriorating jobs and release dates on a single machine. *Acta Automatica Sinica*, 2015, **41**(1): 104−112

在经典的调度模型中, 所有工件在公共的机器资源上加工, 并且所有工件确定一个目标函数. 随着经济的发展和人们消费水平的提高, 顾客对产品的要求趋于多元化个性化, 因此在公共机器上加工的工件可能属于不同的顾客, 而不同的顾客有各自追求的目标, 从而经典的模型中所有工件对应一个目标的调度问题不再满足顾客的需要, 而多代理竞争调度模型或者是求多顾客目标的组合形式, 或者求多顾客的帕累托解, 或者求多顾客有约束条件的解, 这一模型恰能满足顾客个性化的需求. 本文主要研究带有线性恶化工件和释放时间的两个代理单机调度问题, 每个代理有各自的工件集合且所有工件在一台单机上不可中断地加工, 每个代理都希望最小化仅依赖于自己工件完工时间的目标函数, 工件具有与开工时间成比例的加工时间, 工件释放时间可能相同也可能不同. 我们的问题就是要找到调度最小化一个代理的目标函数使得另一个代理的目标函数不超过一个给定的上界.

多代理竞争调度问题首先由 Baker 等[1] 以及 Agnetis 等[2] 提出, 其中工件具有相同的释放时间, 前者研究了单机上两个代理目标函数的线性组合, 后者研究了单机与车间环境下两个代理的有约束优化与帕累托优化问题. 文献 [3] 研究了单机上最小化

收稿日期 2014-03-20 录用日期 2014-07-31
Manuscript received March 20, 2014; accepted July 31, 2014
国家自然科学基金重点项目 (71032004), 国家自然科学基金创新研究群体科学基金项目 (71321001) 资助
Supported by Supported by Key Project of National Natural Science Foundation of China (71032004), the Fund for Innovative Research Groups of the National Natural Science Foundation of China (71321001)
本文责任编委 王红卫
Recommended by Associate Editor WANG Hong-Wei
1. 东北大学工业工程与物流优化研究所 辽宁省制造系统与物流优化重点实验室 沈阳 110819
1. Liaoning Key Laboratory of Manufacturing System and Logistics, Institute of Industrial Engineering and Logistics Optimization, Northeastern University, Shenyang 110819

一个代理的总完工时间, 使得另一个代理的拖期工件数不超过一个给定的上界. 文献 [4−7] 研究了单机上多个代理的调度问题. Leung 等[8] 研究了并行机上两个代理的调度问题, 其中工件有不同的释放时间并允许中断. 文献 [9] 研究了单机与两台并行机上具有可控加工时间的两个代理的调度问题. Mor 等[10] 考虑了单机上最小化工件最大提前完工时间费用与最小化工件总 (加权) 提前完工时间费用和的两个代理调度问题. 文献 [11−13] 研究了单机批处理机上两个代理的调度问题. 上述多代理调度问题并没有考虑工件具有恶化的加工时间, 虽然文献 [8] 考虑了工件具有不同释放时间, 但工件却允许中断, 简化了所研究问题的难度.

工件具有依赖于开工时间的调度问题多年前就有许多学者研究, 其中 Cheng 等[14] 给出了关于工件依赖于开工时间调度问题的综述. 本文主要研究单机上工件与开工时间成比例的简单线性恶化模型, 故这里只对单机上工件具有简单线性恶化作简单介绍. Mosheiov[15] 最先研究了单机上工件具有简单线性恶化的调度问题. Ng 等[16] 考虑了单机上工件具有简单线性恶化且可中断的调度问题.

近年来也有一些关于多代理与工件加工时间依赖于开工时间集成考虑的文献, 如: Liu 等[17] 考虑了单机上工件具有简单线性恶化特征的两个代理的调度问题, 目标函数为: 最小化一个代理的最大拖期使得另一个代理的最大完工时间不超过一个给定的上界, 以及最小化一个代理的总完工时间使得另一个代理的工件最大完工时间费用不超过一个给定的上界. 文献 [18−21] 也研究了单机上工件具有依赖于开工时间的两个代理的调度问题. 上述文献虽然考虑了与开始加工时间相关的两个代理的调度问题, 但工件具有相同的释放时间, 而且大多数所考虑的目标函数都是多项式时间易于求解的.

尽管多代理竞争调度问题已经有大量文献研究, 但关于具有释放时间的多代理单机调度问题的文献却不多. 文献 [22−25] 研究了工件加工时间是常数并且有不同释放时间的两个代理单机调度问题, 上述文献所考虑的问题均为 NP 难的, 对于小规模问题作者利用分支定界求得最优解, 对于大规模问题设计了相应的智能算法求得近优解. 文献 [26−27] 考虑了具有依赖于位置的加工时间并且工件有不同释放时间的两个代理单机调度问题, 目标函数分别为最小化一个代理的总加权完工时间使得另一个代理的最大拖期不超过一个给定的上界, 以及最小化一个代理的总完工时间, 使得另一个代理的拖期工件数为零, 作者同样利用分支定界求得小规模的最优解, 分别利用模拟退火与遗传算法对大规模求得近优解. 本文考虑了同时具有依赖于开工时间线性恶化与释放时间的两个代理单机调度问题, 我们的调度问题较之经典调度问题更能反映部分生产过程, 如钢铁生产企业中对工件的加热存在恶化特征等. 对于相同释放时间模型, 我们延展了文献 [17] 中的目标函数, 考虑了两个 NP 难的加权目标函数. 具有加工时间恶化的两个代理调度问题较之加工时间为常数的两个代理调度问题更为复杂, 难于求解. 因此, 加工时间为常数的两个代理调度问题中, 现有的求解方法有时不再适用于我们的问题, 而已有文献中针对具有恶化工件与相同释放时间的两个代理调度问题或者给出问题的多项式时间算法或者采用智能算法求得 NP 难问题的近优解, 本文中我们针对自己问题的特征对两个 NP 难的问题设计了伪多项式时间动态规划算法, 同时对一些特殊情况给出最优算法, 从理论分析的角度扩充了具有恶化工件的两个代理调度问题的研究. 对于不同释放时间模型, 现有文献很少研究加工时间依赖开工时间恶化的两个代理调度问题, 虽然也考虑了具有不同释放时间的两个代理调度问题, 但多是利用分支定界与智能算法求解, 并没有从理论上对问题加以分析. 因此, 我们从理论上丰富了多代理单机调度问题的研究.

1 问题描述

假设有两个代理 A 和 B, 每个代理都有各自的工件集, 即 $J^A=\{J_1^A, J_2^A, \cdots, J_{n_A}^A\}$, $J^B=\{J_1^B, J_2^B, \cdots, J_{n_B}^B\}$. 设 n 表示所有的工件个数, 即 $n=n_A+n_B$. 所有工件不可中断地在一台单机上加工. 我们假设工件 J_j^X 的实际加工时间为 $p_j^X=\alpha_j^X t, X\in\{A,B\}$, 其中 $\alpha_j^X>0$ 与 t 分别是工件 J_j^X 的恶化率与开始加工时间. 设 $d_j^X\geq 0$ 与 $w_j^X\geq 0$ 分别表示工件 J_j^X 的工期与权值. 我们假设所有的工件或者在时刻 $t_0>0$ 同时可到达加工或者每个工件具有一个不同的释放时间 $r_j^X>0$. 给定一个加工顺序, 我们定义工件 J_j^X 的完工时间为 C_j^X, 如果 $C_j^X>d_j^X$, 我们称工件 J_j^X 是拖期的, 此时定义 $U_j^X=1$, 反之定义 $U_j^X=0$. 利用 Agnetis 等[2] 提出的表示法, 我们的问题表示为 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t; r_j|\gamma^A:\gamma^B\leq Q_B$, 其中 $Q_B\geq 0$ 为给定的上界, 当 r_j 省略时, 表示所有工件的释放时间相同. 我们主要考虑的目标函数有总 (加权) 完工时间 $\sum C_j$ ($\sum w_jC_j$), 总 (加权) 拖期工件数 $\sum U_j$ ($\sum w_jU_j$), 最大完工时间 $C_{\max}$, 以及工件最大完工时间费用 $f_{\max}=\max\{f_j(C_j)\}$.

2 相同释放时间模型

我们首先考虑所有工件在时刻 $t_0>0$ 同时可到达加工. Liu 等[17] 考虑了不加权值时的目标函数是多项式时间可解的, 这里我们考虑两个加权的目标

函数以及一些特殊情况. 首先利用文献 [17], 对于问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|C_{\max}$, 最大完工时间等于 $t_0\prod_{h=1}^{n_A}(1+\alpha_h^A)\prod_{k=1}^{n_B}(1+\alpha_k^B)$, 因此我们有如下的引理:

引理 1. 对于问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|C_{\max}$, 最大完工时间与工件加工顺序无关.

2.1 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$

我们首先通过 NP 难的 Subset Product 问题[28] 归结, 证明问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$ 是 NP 难的, 然后我们给出了一个伪多项式时间算法来求解该问题, 从而说明该问题是一般意义 NP 难的.

Subset product (SP) problem. 给定一个有限集合 $S=\{1,2,\cdots,m\}$ 与一个正整数 A, 对于每个元素 $i\in S$ 对应一个正整数 a_i, 是否存在一个子集 $S'\subseteq S$ 满足 $\prod_{i\in S'}a_i=A$?

设 $D=\prod_{i\in S}a_i=AB$. 在上面的 SP 问题中, 如果对于每个元素 $i\in S$ 有 $a_i=1$, 那么这样的元素可以从集合 S 中移除, 因为这样的元素将不影响任何集合的乘积. 因此我们假设对于每一个元素 $i\in S$ 有 $a_i\ge 2$.

定理 1. 问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$ 是 NP 难的.

证明. 给定 SP 问题的一个实例, 构造我们问题的一个实例如下:

代理 A 和 B 的工件数分别为: $n_A=m$; $n_B=1$.

代理 A 和 B 的工件恶化率分别为: $\alpha_h^A=a_h-1, h=1,2,\cdots,m$; $\alpha_1^B=A-1$.

代理 A 的工件的权值为: $w_h^A=(1-1/a_h)/A, h=1,2,\cdots,m$.

所有工件的到达时间为: $t_0=1$.

阈值分别为: $Q_A=1+D-A-1/A$; $Q_B=A^2$.

我们可以证明, 存在一个调度目标值满足 $\sum w_h^A C_h^A\le Q_A$ 与 $C_{\max}^B\le Q_B$, 当且仅当 SP 问题有解.

假设给定一个子集 $S'\subseteq S$ 满足 $\prod_{i\in S'}a_i=A$, 我们构造机器上工件的调度如下所示: 在工件 J_1^B 之前加工子集 S' 中代理 A 的工件, 余下的工件在 J_1^B 之后加工. 因此, 工件 J_1^B 的完工时间为 $C_{\max}^B=t_0\prod_{J_h^A\in S'}(1+\alpha_h^A)(1+\alpha_1^B)=A\prod_{h\in S'}a_h=A^2\le Q_B$, 代理 A 的工件的总加权完工时间为

$$\sum_{h=1}^{m}w_h^A C_h^A=\sum_{J_h^A\in S'}w_h^A C_h^A+\sum_{J_h^A\in S\setminus S'}w_h^A C_h^A=$$
$$\sum_{h=1}^{|S'|}\frac{1}{A}\left(1-\frac{1}{a_{[h]}}\right)t_0\prod_{i=1}^{h}(1+\alpha_{[i]}^A)+$$
$$\sum_{h=|S'|+1}^{m}\frac{1}{A}\left(1-\frac{1}{a_{[h]}}\right)C_{\max}^B\prod_{i=|S'|+1}^{h}(1+\alpha_{[i]}^A)=$$
$$\prod_{h\in S'}\frac{a_h}{A}-\frac{1}{A}+A\prod_{h\in S'}a_h\left(\prod_{h\in S\setminus S'}\frac{a_h}{A}-\frac{1}{A}\right)=$$
$$1+D-A-\frac{1}{A}\le Q_A \tag{1}$$

其中, $|S'|$ 表示子集 S' 中的工件个数, $(1-1/a_{[h]})/A$ 和 $\alpha_{[h]}^A$ $(h=1,2,\cdots,m)$ 分别为在第 h 个位置上加工工件的权值和恶化率.

因此, 我们得到一个可行调度满足 $\sum w_h^A C_h^A\le Q_A$ 与 $C_{\max}^B\le Q_B$.

反之, 若存在一个调度实例满足 $\sum w_h^A C_h^A\le Q_A$ 与 $C_{\max}^B\le Q_B$, 则 Subset product 问题有解. 假设集合 S' 是加工在工件 J_1^B 之前的代理 A 的工件集, 于是工件 J_1^B 的完工时间为 $C_{\max}^B=t_0\prod_{J_h^A\in S'}(1+\alpha_h^A)(1+\alpha_1^B)=A\prod_{h\in S'}a_h\le Q_B$. 因此, 我们有 $\prod_{h\in S'}a_h\le A$. 此外, 基于式 (1) 我们有 $\sum_{h=1}^m w_h^A C_h^A=\prod_{h\in S'}a_h/A-1/A+D-\prod_{h\in S'}a_h\le Q_A$, 即 $\prod_{h\in S'}a_h\ge A$. 因此, $\prod_{h\in S'}a_h=A$. 由此可知, 集合 S' 为 Subset product 问题提供了解. □

现在我们给出一个伪多项式时间动态规划算法来求解问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$.

引理 2. 对于问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$, 如果存在一个最优调度, 那么所有代理 B 的工件连续加工.

证明. 基于引理 1 中工件的最大完工时间与加工顺序无关, 将工件交换位置使得代理 B 中除最后一个工件外其他的工件向右移动到代理 B 中最后一个工件前与之连续加工, 这样得到的调度中代理 A 中每个工件的完工时间不会增加, 进而总加权完工时间也不会增加. □

基于引理 2, 将所有代理 B 的工件构成一个大的工件, 记为工件 B.

引理 3. 对于问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t|\sum w_h^A C_h^A : C_{\max}^B\le Q_B$, 如果存在一个最优调度, 那么在工件 B 之前生产与之后生产的代理 A 的工件集, 分别按比值 $\alpha_h^A/(1+\alpha_h^A)w_h^A$ 不减的顺序加工生产.

证明. 由引理 2 我们知道, 工件 B 将代理 A 中的工件分成前后两部分, 每部分工件集对应于问题 $1|p_h^A=\alpha_h^A t|\sum w_h^A C_h^A$. 利用文献 [15] 中最小化总加权完工时间的最优调度将所有工件按照比值

$\alpha_h^A/(1+\alpha_h^A)w_h^A$ 不减的顺序加工生产, 因此结论得以证明. □

以下我们假设所有的参数均为非负整数. 假设在工件 B 之前生产的代理 A 的工件集记为集合 S_1, 之后生产的代理 A 的工件集记为集合 S_2. 由引理 3 假设所有代理 A 的工件按比值 $\alpha_h^A/(1+\alpha_h^A)w_h^A$ 不减的顺序排列.

算法 1. 设 $f(h,t,s)$ 是调度一部分工件 $J_1^A, J_2^A, \cdots, J_h^A$ 时代理 A 的最小总加权完工时间, 其中 t 是集合 S_1 中工件的最大完工时间, s 是工件 B 的开始加工时间. 定义 $A_h = t_0\prod_{i=1}^{h}(1+\alpha_i^A)$. 当机器加工完工件 J_{h-1}^A 时, 加工工件 J_h^A 或者在集合 S_1 中, 此时有 $f(h,t,s)=f(h-1,\frac{t}{1+\alpha_h^A},s)+w_h^A t$; 或者在集合 S_2 中, 此时有 $f(h,t,s)=f(h-1,t,s)+s\prod_{k=1}^{n_B}(1+\alpha_k^B)w_h^A\frac{A_h}{t}$. 由此我们有如下动态规划:

初始条件: $f(0,t_0,s)=0$, $f(h,t,s)=+\infty$, 如果 $t<0$ 或者 $t>s$, $s=t_0,t_0+1,\cdots,\lfloor\frac{Q_B}{\prod_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor$.

递归关系: $f(h,t,s)=$

$$\min\begin{cases} f(h-1,\frac{t}{1+\alpha_h^A},s)+w_h^A t, & t_0\le t\le s,\\ f(h-1,t,s)+s\prod_{k=1}^{n_B}(1+\alpha_k^B)w_h^A\frac{A_h}{t}, & t_0\le t\le s.\end{cases}$$

最优解值: $\min\{f(n_A,t,s): t_0\le t\le s, s=t_0,t_0+1,\cdots,\lfloor\frac{Q_B}{\prod_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor\}$.

显然算法 1 的复杂性是 $\mathrm{O}(n_A Q_B^2)$.

2.2　$1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: C_{\max}^B\le Q_B$

我们首先通过 Subset product 问题归结证明问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: C_{\max}^B\le Q_B$ 是 NP 难的, 然后我们针对一些特殊情形给出了求解方法.

2.2.1　NP 难证明

定理 2. 问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: C_{\max}^B\le Q_B$ 是 NP 难的.

证明. 给定 SP 问题的一个实例, 构造我们问题的一个实例如下:

$n_A=m$, $n_B=1$; $\alpha_h^A=a_h-1$, $w_h^A=\ln a_h$, $d_h^A=A$, $h=1,2,\cdots,m$; $\alpha_1^B=A-1$; $t_0=1$; $Q_A=\ln B$, $Q_B=A^2$.

与定理 1 证明类似, 我们能证明存在一个调度目标值满足 $\sum w_h^A U_h^A\le Q_A$ 与 $C_{\max}^B\le Q_B$, 当且仅当 SP 问题有解, 这里省略详细证明. □

注 1. 从定理 2 的证明可以得出如下问题也是 NP 难的:

$1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: f_{\max}^B\le Q_B$
$1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: \sum U_k^B\le Q_B$
$1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: \sum w_k^B U_k^B\le Q_B$

2.2.2　$1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum U_h^A: f_{\max}^B\le Q_B$

由注 1 知道问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum w_h^A U_h^A: f_{\max}^B\le Q_B$ 是 NP 难的, 本节我们考虑它的一个特殊情况, 即当代理 A 中所有工件权值相等时问题是可解的. 首先对每一个工件 J_k^B, 定义一个最晚开始加工时间 LS_k^B 满足: 当 $t\le LS_k^B$ 时, $f_k^B((1+\alpha_k^B)t)\le Q_B$; 当 $t>LS_k^B$ 时, $f_k^B((1+\alpha_k^B)t)>Q_B$. 将代理 B 的所有工件按 LS_k^B 不减的顺序排列, 这时我们重新定义 LS_k^B, 设 $LS_{n_B}^B := LS_{n_B}^B$, $LS_k^B := \min\{LS_k^B, \frac{LS_{k+1}^B}{1+\alpha_k^B}\}$, $k=n_B-1,\cdots,1$.

我们定义可中断问题 $1|p_h^A=\alpha_h^A t;\ p_k^B=\alpha_k^B t, pmtn|\gamma^A:\gamma^B\le Q_B$, 其中 $pmtn$ 表示工件允许中断, 如下:

定义 1. 假设 $[s_i,t_i]$, $i=1,2,\cdots,k$ 是工件 J_j^X 的 k 个不相交的加工间隔, 其中 $0<r_j^X<s_1<t_1<s_2<t_2<\cdots<s_k<t_k$. 定义工件 J_j^X 在间隔 $[s_i,t_i]$, $i=1,2,\cdots,k$ 内的恶化率为 $I_{ji}^X=\frac{t_i-s_i}{s_i}$, 并且 $\prod_{i=1}^{k}(1+I_{ji}^X)=1+\alpha_j^X$ (见文献 [16]).

由文献 [2] 中问题 $1||\sum U_h^A: f_{\max}^B\le Q_B$ 的结论推广, 我们得到如下两个性质:

引理 4. 如果问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t, pmtn|\sum U_h^A: f_{\max}^B\le Q_B$ 存在一个最优解, 那么问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum U_h^A: f_{\max}^B\le Q_B$ 存在一个关于代理 A 工件相同数量拖期工件数的最优解.

引理 5. 如果问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t, pmtn|\sum U_h^A: f_{\max}^B\le Q_B$ 存在一个最优解, 那么最优解中所有代理 B 的工件不可中断地在时刻 LS_k^B 开始加工.

基于引理 5, 对于问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t, pmtn|\sum U_h^A: f_{\max}^B\le Q_B$ 的一个最优解, 我们可以先确定代理 B 每个工件的位置, 代理 A 中工件的位置可以通过一个辅助问题来确定. 在辅助问题中, 只包括不可中断的代理 A 中的工件, 对于工件 J_h^A, 它的修正工期 $\bar{d}_h^A$ 定义如下: 如果 d_h^A 落在间隔 $[LS_k^B,(1+\alpha_k^B)LS_k^B]$ 外部, 那么 $\bar{d}_h^A=d_h^A-\sum_{(1+\alpha_k^B)LS_k^B\le d_h^A}\alpha_k^B LS_k^B$; 如果 d_h^A 落在间隔 $[LS_k^B,(1+\alpha_k^B)LS_k^B]$ 内部, 那么 $\bar{d}_h^A=LS_k^B-\sum_{(1+\alpha_k^B)LS_k^B\le d_h^A}\alpha_k^B LS_k^B$. 最后代理 A 中工件的位置可以通过文献 [15] 中的问题 $1|p_h^A=\alpha_h^A t|\sum U_h^A$ 来确定. 由此基于引理 4, 我们得到解决问题 $1|p_h^A=\alpha_h^A t; p_k^B=\alpha_k^B t|\sum U_h^A: f_{\max}^B\le Q_B$ 的最优算法如下:

算法 2.

步骤 1. 利用修正的工期, 通过解问题 $1|p_h^A=\alpha_h^A t|\sum U_h^A$ 来确定代理 A 中工件的位置.

步骤 2. 从左到右在时刻 LS_k^B 不可中断地加工代理 B 中的每个工件, 得到问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t, pmtn|\sum U_h^A : f_{\max}^B \le Q_B$ 的一个最优调度, 此时设工件 J_h^A 的完工时间为 C_h^A.

步骤 3. 在间隔 $[\frac{C_h^A}{1+\alpha_h^A}, C_h^A]$ 内调度完整的工件 J_h^A, 并向后移动所有其他的工件, 最终得到问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum U_h^A : f_{\max}^B \le Q_B$ 的最优解.

算法 2 中排列代理 B 的工件需要 $\mathrm{O}(n_B \log n_B)$ 时间, 解决辅助问题 $1|p_h^A = \alpha_h^A t|\sum U_h^A$ 需要 $\mathrm{O}(n_A \log n_A)$ 时间, 故算法 2 的时间复杂性是 $\mathrm{O}(n_A \log n_A) + \mathrm{O}(n_B \log n_B)$.

2.2.3 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum w_k^B U_k^B \le Q_B$

通过注 1, 我们知道问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum w_k^B U_k^B \le Q_B$ 是 NP 难的. 下面给出一个伪多项式时间动态规划算法, 进而说明问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum w_k^B U_k^B \le Q_B$ 也是一般意义 NP 难的.

引理 6. 如果问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum w_k^B U_k^B \le Q_B$ 存在一个最优调度, 那么所有延迟工件在调度末尾连续加工, 所有提前工件在调度开头按最小工期 (Earliest due date, EDD) 优先规则连续加工.

证明. 将所有延迟工件移动到调度末尾连续加工, 那么所有提前工件将在调度开头连续加工, 因此所有提前工件的完工时间不会增加, 进而加权拖期工件数也不会增加. 将所有提前工件在调度开头按 EDD 规则连续加工, 这样也不会增加加权拖期工件数. □

算法 3.

由引理 6 我们假设所有的工件按 EDD 规则排列, 即 $d_1 \le d_2 \le \cdots \le d_n$. 设 $C(i, w_A, w_B)$ 是调度一部分工件 $J_1, J_2, \cdots, J_i$ 中最后一个提前工件的最小完工时间, 其中 w_A 与 w_B 分别是代理 A 与 B 中拖期工件的权值和.

初始条件: $C(0, 0, 0) = t_0$, $C(i, w_A, w_B) = +\infty$, 如果 $i < 0$ 或者 $w_A < 0$ 或者 $w_B < 0$ 或者 $w_B > Q_B$.

递归关系:

$$f(i, w_A, w_B) = \begin{cases} 0, & \text{如果 } (1+\alpha_i)C(i-1, w_A, w_B) \le d_i \\ +\infty, & \text{否则} \end{cases}$$

$$C(i, w_A, w_B) = \begin{cases} \min\{(1+\alpha_i)C(i-1, w_A, w_B) + f(i, w_A, w_B); \\ \quad C(i-1, w_A - w_i, w_B)\}, & \text{若 } J_i \in J^A \\ \min\{(1+\alpha_i)C(i-1, w_A, w_B) + f(i, w_A, w_B); \\ \quad C(i-1, w_A, w_B - w_i)\}, & \text{若 } J_i \in J^B \end{cases}$$

最优解值: $w_A^* = \min\{w_A : C(n_A + n_B, w_A, w_B) < +\infty, 0 \le w_A \le \sum_{h=1}^{n_A} w_h^A, w_B \le Q_B\}$.

容易证明算法 3 的时间复杂性为 $\mathrm{O}(nQ_B W_A)$, 其中 $W_A = \sum_{h=1}^{n_A} w_h^A$.

下面针对算法 3, 给出一个具体的数值实例.

例 1.

我们假设 $J^A = \{J_1^A, J_2^A\}$, 其中 $\alpha_1^A = 1$, $\alpha_2^A = 3$, $d_1^A = 4$, $d_2^A = 9$, $w_1^A = 1$, $w_2^A = 3$; $J^B = \{J_1^B\}$, 其中 $\alpha_1^B = 0.5$, $d_1^B = 6$, $w_1^B = 4$; $t_0 = 1$; $Q_B = 5$. 所有工件按照工期不减的顺序排列为 $d_1^A < d_1^B < d_2^A$, 我们有下面的结论:

$f(1,0,0) = 0$

$C(1,0,0) = \min\{(1+\alpha_1^A)C(0,0,0) + f(1,0,0); C(0,-w_1^A,0)\} = 2$

$f(1,w_1^A,0) = +\infty$

$C(1,w_1^A,0) = \min\{(1+\alpha_1^A)C(0,w_1^A,0) + f(1,w_1^A,0); C(0,0,0)\} = 1$

$f(2,0,0) = 0$

$C(2,0,0) = \min\{(1+\alpha_1^B)C(1,0,0) + f(2,0,0); C(1,0,-w_1^B)\} = 3$

$f(2,w_1^A,0) = 0$

$C(2,w_1^A,0) = \min\{(1+\alpha_1^B)C(1,w_1^A,0) + f(2,w_1^A,0); C(1,w_1^A,-w_1^B)\} = 1.5$

$f(2,0,w_1^B) = +\infty$

$C(2,0,w_1^B) = \min\{(1+\alpha_1^B)C(1,0,w_1^B) + f(2,0,w_1^B); C(1,0,0)\} = 2$

$f(2,w_1^A,w_1^B) = +\infty$

$C(2,w_1^A,w_1^B) = \min\{(1+\alpha_1^B)C(1,w_1^A,w_1^B) + f(2,w_1^A,w_1^B); C(1,w_1^A,0)\} = 1$

$f(3,0,0) = +\infty$

$C(3,0,0) = \min\{(1+\alpha_2^A)C(2,0,0) + f(3,0,0); C(2,-w_2^A,0)\} = +\infty$

$f(3,w_1^A,0) = 0$

$C(3,w_1^A,0) = \min\{(1+\alpha_2^A)C(2,w_1^A,0) + f(3,w_1^A,0); C(2,w_1^A - w_2^A,0)\} = 6$

$f(3,0,w_1^B) = 0$

$C(3,0,w_1^B) = \min\{(1+\alpha_2^A)C(2,0,w_1^B) + f(3,0,w_1^B); C(2,-w_2^A,w_1^B)\} = 8$

由于 $C(3,0,w_1^B) < +\infty$, 并且 $w_1^B < Q_B$, 因此最优调度是 $\{J_1^A, J_2^A, J_1^B\}$, 具有最优值 $w_A^* = 0$.

由注 1 知道问题 $1|p_h^A = \alpha_h^A t;\ p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum U_k^B \le Q_B$ 是 NP 难的, 利用

算法 3 同样可以在伪多项式时间 $O(nn_BW_A)$ 内求得最优解, 其中 $W_A = \sum_{h=1}^{n_A} w_h^A$. 因此, 问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t|\sum w_h^A U_h^A : \sum U_k^B \leq Q_B$ 也是一般意义 NP 难的.

当代理 A 中工件的权值相等以及代理 B 中工件的权值也相等时, 即问题 $1|p_h^A = \alpha_h^A t;\ p_k^B = \alpha_k^B t|\sum U_h^A : \sum U_k^B \leq Q_B$, 利用算法 3 仍可获得问题的最优解, 此时算法的时间复杂性为 $O(nn_An_B)$, 所以问题 $1|p_h^A = \alpha_h^A t;\ p_k^B = \alpha_k^B t|\sum U_h^A : \sum U_k^B \leq Q_B$ 是多项式时间可解的.

3 不同释放时间模型

由上一节我们知道, 当工件释放时间相同时, 最小化总加权完工时间和最小化总加权拖期工件数是 NP 难问题, 对于这两个问题的求解是非常难的. 所以当工件释放时间不同时最小化这两个目标函数也同样是 NP 难的, 求解方法会更加复杂. 因此, 对于这两个目标函数我们只考虑上一节中释放时间相同的特殊情况, 而不再研究释放时间不同的情况. 另外在上节中我们得到最小化一个代理的拖期工件数使得另一个代理的拖期工件数不超过一个给定的上界是多项式时间可解的, 同时由文献 [17] 知道最小化一个代理的最大完工时间或者最小化这个代理的总完工时间同时满足另一个代理的最大完工时间不超过一个给定的上界, 这两个问题也是多项式时间可解的, 但是当工件的释放时间不同时, 以上三个问题都是 NP 难的. 因此, 本节我们将对这三个问题给出 NP 难的理论证明, 以及提出它们一些可解的特殊情况.

3.1 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; r_j|C_{\max}^A : C_{\max}^B \leq Q_B$

我们首先证明问题 $1|p_h^A = \alpha_h^A t;\ p_k^B = \alpha_k^B t; r_j|C_{\max}^A : C_{\max}^B \leq Q_B$ 是 NP 难的, 然后针对一个特殊情况给出了最优算法.

3.1.1 NP 难证明

问题 $1|nr-a; p_j = \alpha_j t|C_{\max}$[29]. 给定 m 个工件 $J_1, \cdots, J_m$ 在一台单机上不可中断地加工, 所有工件在时刻 t_0 均可到达. 工件的恶化率为 α_j, $j = 1, 2, \cdots, m$. 机器在时间间隔 (b_1, b_2), $b_2 > b_1 > t_0$ 内不可利用. 是否存在一个调度满足 $C_{\max} \leq Q$?

定理 3. 问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; r_j|C_{\max}^A : C_{\max}^B \leq Q_B$ 是 NP 难的.

证明. 我们通过 NP 难问题 $1|nr-a;\ p_j = \alpha_j t|C_{\max}$ 来证明. 构造我们问题的一个实例如下:

$n_A = m$, $n_B = 1$; $\alpha_h^A = \alpha_h$, $r_h^A = t_0$, $h = 1, 2, \cdots, m$; $r_1^B = b_1$; $Q_A = Q$, $Q_B = b_1(1+\alpha_1^B) = b_2$.

显然, 构造的实例就是问题 $1|nr-a;\ p_j = \alpha_j t|C_{\max}$. 因此我们的问题是 NP 难的. □

3.1.2 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; r^A = r; r^B = t_0|C_{\max}^A : C_{\max}^B \leq Q_B$

假设代理 A 中工件有共同的释放时间 $r^A = r$, 代理 B 中工件有共同的释放时间 $r^B = t_0$, 且 $r > t_0$. 假设所有参数均为非负整数.

这种情况下不妨假设 $t_0 \prod_{k=1}^{n_B}(1+\alpha_k^B) > r$, 否则, 如果存在一个可行解, 那么我们能在时刻 r 之前调度代理 B 的所有工件. 与引理 2 证明相似, 我们有下面的性质:

引理 7. 对于问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; r^A = r; r^B = t_0|C_{\max}^A : C_{\max}^B \leq Q_B$, 如果存在一个最优解, 那么代理 A 的所有工件在时刻 r 或者 r 之后连续加工.

定理 4. 对于问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; r^A = r; r^B = t_0|C_{\max}^A : C_{\max}^B \leq Q_B$, 如果存在一个最优解满足条件: 1) $t_0 \prod_{k=1}^{n_B}(1+\alpha_k^B) = Q_B$, 那么最优调度是在时刻 t_0 以任意顺序先加工代理 B 中的所有工件, 再以任意顺序在时刻 Q_B 加工代理 A 中的所有工件; 2) $Q_B < C_{OPT}^B(s^A = r)$, 其中 $C_{OPT}^B(s^A = r)$ 表示机器具有不可利用间隔 $(r, r\prod_{h=1}^{n_A}(1+\alpha_h^A))$ 时, 最小化代理 B 中所有工件的最大完工时间的最优值, 这里代理 B 中工件如果在不可利用间隔之前没有加工完, 那么在遇到不可利用间隔后必须从头开始加工, 即工件不可中断, s^A 表示代理 A 的开始加工时间, 那么此时的最优调度在 $O(n_B(Q_B - r + 1)(Q_B - t_0))$ 时间内能获得; 3) $Q_B \geq C_{OPT}^B(s^A = r)$, 其中 $C_{OPT}^B(s^A = r)$ 的定义如 2) 中所示, 那么最优调度在 $O(n_B(r - t_0))$ 时间内能获得.

证明. 1) 的结论是显然的.

2) 对于每个 s^A, $s^A = r, r+1, \cdots, t_0\prod_{k=1}^{n_B}(1+\alpha_k^B)$, 构造一个对应的新问题 π, 其中机器具有不可利用间隔 $(s^A, s^A\prod_{h=1}^{n_A}(1+\alpha_h^A))$, 且代理 B 中工件不可中断. 利用问题 $1|nr-a; p_j = \alpha_j t|C_{\max}$[29] 的求解动态规划, 对于每个 s^A 我们得到一个最优值 $C_{\max}^B$, 从中选择满足条件 $C_{\max}^B \leq Q_B < C_{OPT}^B(s^A = r)$ 的最小的 s^A, 这个 s^A 所对应的调度即为最优调度. 问题 $1|nr-a; p_j = \alpha_j t|C_{\max}$[29] 的求解动态规划时间复杂度为 $O(m(b_1 - t_0))$, 并且 s^A 有 $t_0\prod_{k=1}^{n_B}(1+\alpha_k^B) - r + 1$ 个取值, $t_0\prod_{k=1}^{n_B}(1+\alpha_k^B) \leq Q_B$, 因此, 最优调度在 $O(n_B(Q_B - r + 1)(Q_B - t_0))$ 时间内能获得.

3) 如果 $Q_B \geq C_{OPT}^B(s^A = r)$, 那么我们能够在时刻 r 连续地生产代理 A 中的所有工件, 然后利用问题 $1|nr-a; p_j = \alpha_j t|C_{\max}$[29] 的求解动态规划调度具有不可利用间隔 $(r, r\prod_{h=1}^{n_A}(1+\alpha_h^A))$ 时代理 B 中所有工件. 因此, 最优调度在 $O(n_B(r - t_0))$ 时间内能获得. □

3.2 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum C_h^A:C_{\max}^B\leq Q_B$

我们首先证明问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum C_h^A:C_{\max}^B\leq Q_B$ 是 NP 难的, 然后针对一个特殊情况给出最优算法.

3.2.1 NP 难证明

问题$1|nr-a;p_j=\alpha_j t|\sum C_j$[29]. 给定 m 个工件 $J_1,\cdots,J_m$ 在一台单机上不可中断地加工, 所有工件在时刻 t_0 均可到达. 工件的恶化率为 α_j, $j=1,2,\cdots,m$. 机器在时间间隔 (b_1,b_2), $b_2>b_1>t_0$ 内不可利用. 是否存在一个调度满足 $\sum C_j\leq Q$?

定理 5. 问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum C_h^A:C_{\max}^B\leq Q_B$ 是 NP 难的.

证明. 通过 NP 难问题 $1|nr-a;p_j=\alpha_j t|\sum C_j$ 来证明我们的问题. 构造一个实例如下:

$n_A=m$, $n_B=1$; $\alpha_h^A=\alpha_h$, $r_h^A=t_0$, $h=1,2,\cdots,m$; $r_1^B=b_1$; $Q_A=Q$, $Q_B=b_1(1+\alpha_1^B)=b_2$.

显然, 构造的实例就是问题 $1|nr-a;p_j=\alpha_j t|\sum C_j$. 因此我们的问题是 NP 难的. □

3.2.2 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r^A=t_0;r^B=r|\sum C_h^A:C_{\max}^B\leq Q_B$

假设代理 A 中工件有共同的释放时间 $r^A=t_0$, 代理 B 中工件有共同的释放时间 $r^B=r$, 且 $r>t_0$. 假设所有参数均为非负整数. 同样类似于引理 2 的证明, 我们有下面的两个性质:

引理 8. 对于问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r^A=t_0;r^B=r|\sum C_h^A:C_{\max}^B\leq Q_B$, 如果存在一个最优调度, 那么代理 B 的所有工件在时刻 r 或者 r 之后连续加工.

由引理 8, 我们假设所有代理 B 中的工件构成一个大的工件, 记为工件 B.

引理 9. 对于问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r^A=t_0;r^B=r|\sum C_h^A:C_{\max}^B\leq Q_B$, 如果存在一个最优调度, 那么在工件 B 之前生产与之后生产的代理 A 的工件集分别按最小恶化率 (Smallest deterioration rate, SDR) 优先规则加工生产.

证明. 由引理 8 我们知道, 工件 B 将代理 A 中的工件分成前后两部分, 每部分工件集对应于问题 $1|p_h^A=\alpha_h^A t|\sum C_h^A$. 利用文献 [15] 中最小化总完工时间的最优调度为将所有工件按照 SDR 规则加工生产, 因此结论得以证明. □

由引理 9 假设所有代理 A 的工件按恶化率不减的顺序排列. 类似于算法 1, 有如下的动态规划:

算法 4.

设 $f(h,t,s)$ 是调度一部分工件 $J_1^A,J_2^A,\cdots,J_h^A$ 时代理 A 的最小总完工时间, 其中 t 是在工件 B 之前加工的代理 A 中工件的最大完工时间, s 是工件 B 的开始加工时间. 定义 $A_h=t_0\prod_{i=1}^{h}(1+\alpha_i^A)$.

初始条件: $f(0,t_0,s)=0$, $f(h,t,s)=+\infty$, 如果 $t<0$ 或者 $t>s$, $s=r,r+1,\cdots,\lfloor\frac{Q_B}{\prod_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor$.

递归关系:

$$f(h,t,s)=\min\begin{cases}f(h-1,\frac{t}{1+\alpha_h^A},s)+t, t_0\leq t\leq s,\\ \qquad s=r,r+1,\cdots,\lfloor\frac{Q_B}{\prod\limits_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor\\ f(h-1,t,s)+s\prod\limits_{k=1}^{n_B}(1+\alpha_k^B)\frac{A_h}{t},\\ \qquad t_0\leq t\leq s,\\ \qquad s=r,r+1,\cdots,\lfloor\frac{Q_B}{\prod\limits_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor\end{cases}$$

最优解值: $\min\{f(n_A,t,s):t_0\leq t\leq s,s=r,r+1,\cdots,\lfloor\frac{Q_B}{\prod_{k=1}^{n_B}(1+\alpha_k^B)}\rfloor\}$.

容易证明算法 4 的复杂性是 $\mathrm{O}(n_AQ_B^2)$.

3.3 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum U_h^A:\sum U_k^B\leq Q_B$

本节我们将研究问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum U_h^A:\sum U_k^B\leq Q_B$ 的复杂性. 对于该问题当代理 B 的工件只有一个释放时间和一个工期时, 那么问题至少是一般意义 NP 难的; 当代理 B 的工件具有任意数量的释放时间和工期时, 那么问题是强 NP 难的. 最后针对一个特殊情况提出了多项式时间算法.

3.3.1 NP 难证明

定理 6. 对于问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum U_h^A:\sum U_k^B\leq Q_B$, 如果代理 B 的工件只有一个释放时间和一个工期, 那么问题至少是一般意义 NP 难的.

证明. 我们通过 Subset product 问题归结来证明. 给定 SP 问题的一个实例, 构造我们问题的一个实例如下:

$n_A=m$, $n_B=1$; $\alpha_h^A=a_h-1$, $r_h^A=t_0=1$, $d_h^A=AD$, $h=1,2,\cdots,m$; $\alpha_1^B=A-1$, $r_1^B=A$, $d_1^B=A^2$; $Q_A=0$, $Q_B=0$.

与定理 3 证明类似, 在此省略详细证明. □

定理 7. 对于问题 $1|p_h^A=\alpha_h^A t;p_k^B=\alpha_k^B t;r_j|\sum U_h^A:\sum U_k^B\leq Q_B$, 如果代理 B 的工件具有任意数量的释放时间和工期, 那么问题是强 NP 难的.

证明. 我们通过强 NP 难的 4-Product (4-P) 问题[30] 归结来证明. 4-P 问题描述如下:

4-Product(4-P) problem. 给定正有理数 $a_1,a_2,\cdots,a_{4p}$ 以及 A, 满足 $A^{\frac{1}{5}}<a_i<A^{\frac{1}{3}}$, $i=1,2,\cdots,4p$, 并且 $\prod_{i=1}^{4p}a_i=A^p$, 是否存在一个划分将集合 $X=\{1,2,\cdots,4p\}$ 划分成 p 个不相交的子集 $X_1,X_2,\cdots,X_p$ 满足 $\prod_{i\in X_k}a_i=A$, $k=$

$1, 2, \cdots, p$?

给定 4-P 问题的一个实例, 构造我们问题的一个实例如下:

$n_A = 4p$, $n_B = p$; $\alpha_h^A = a_h - 1$, $r_h^A = t_0 = 1$, $d_h^A = A^{2p}$, $h = 1, 2, \cdots, 4p$; $\alpha_k^B = A - 1$, $r_k^B = A^{2k-1}$, $d_k^B = A^{2k}$, $k = 1, 2, \cdots, p$; $Q_A = 0, Q_B = 0$.

我们可以证明, 存在一个调度目标值满足 $\sum U_h^A \le Q_A$ 与 $\sum U_k^B \le Q_B$, 当且仅当 4-P 问题有解.

假设给定 p 个不相交的子集 $X_1, X_2, \cdots, X_p$ 满足 $\prod_{i \in X_k} a_i = A$, $k = 1, 2, \cdots, p$. 设 J_{X_k} 表示对应于集合 X_k 中元素的工件集. 我们构造机器上工件的调度如下所示: 首先调度代理 B 中的工件在它们各自的释放时间, 即工件 J_1^B 在 $r_1^B = A$ 时刻开始生产, 它的完工时间为 $C_1^B = A^2 = d_1^B$; 工件 J_2^B 在 $r_2^B = A^3$ 时刻开始生产, 它的完工时间为 $C_2^B = A^4 = d_2^B$; 直到最后一个工件 J_p^B 在 $r_p^B = A^{2p-1}$ 时刻开始生产, 它的完工时间为 $C_p^B = A^{2p} = d_p^B$. 然后在时间间隔 $[t_0 A^{2k-2}, t_0 A^{2k-1}]$ 内无空闲时间地调度集合 J_{X_k} $(k = 1, 2, \cdots, p)$ 中的工件. 因此, 我们得到 $\sum U_h^A = 0 \le Q_A$ 与 $\sum U_k^B = 0 \le Q_B$.

假设给定我们问题实例的一个调度满足 $\sum U_h^A \le Q_A$ 与 $\sum U_k^B \le Q_B$, 其中条件 $\sum U_k^B \le Q_B = 0$ 能够成立必须有每个工件 J_k^B 在各自的释放时间 $r_k^B = A^{2k-1}$ 开始加工, 这是由于 $\alpha_k^B = A - 1$, $r_k^B = A^{2k-1}$ 以及 $d_k^B = A^{2k}$, $k = 1, 2, \cdots, p$ 所决定的. 因此, 所有的工件 J_k^B 形成了 p 个不相交的时间间隔 $[t_0 A^{2k-2}, t_0 A^{2k-1}]$ $(k = 1, 2, \cdots, p)$. 由于 $C_p^B = d_h^A = A^{2p}$, 所以条件 $\sum U_h^A \le Q_A = 0$ 能够成立必须所有的工件 J_h^A $(h = 1, 2, \cdots, 4p)$ 在这些间隔内加工完成. 由于 $\prod_{h=1}^{4p}(1 + \alpha_h^A) = \prod_{h=1}^{4p} a_h = A^p$, 并且每个间隔有 A 时间单位长度, 因此这些间隔内的工件集就对应于集合 X_k $(k = 1, 2, \cdots, p)$. 因此, 这为 4-P 问题提供了解. □

3.3.2 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; agr(r_j, d_j)|\sum U_h^A : \sum U_k^B \le Q_B$

我们考虑所有工件释放时间与工期一致的调度问题, 即如果 $r_i < r_j$, 那么 $d_i \le d_j$, 表示为 $agr(r_j, d_j)$. 类似于引理 6 的证明, 我们能证明引理的结论对于问题 $1|p_h^A = \alpha_h^A t; p_k^B = \alpha_k^B t; agr(r_j, d_j)|\sum U_h^A : \sum U_k^B \le Q_B$ 仍然成立. 类似于算法 3, 有如下的动态规划算法:

算法 5.

假设所有的工件按 $r_1 \le r_2 \le \cdots \le r_n$ 以及 $d_1 \le d_2 \le \cdots \le d_n$ 顺序排列. 设 $C(i, h, k)$ 是调度一部分工件 $J_1, J_2, \cdots, J_i$ 中最后一个提前工件的最小完工时间, 其中 h 与 k 分别是当前调度中代理 A 与 B 至多能拖期的工件数.

初始条件: $C(0, 0, 0) = t_0$, $C(i, h, k) = +\infty$, 若 $i < 0$ 或 $h < 0$ 或 $k < 0$ 或 $k > Q_B$.

递归关系: $f(i, h, k) =$

$$\begin{cases} 0, & \text{如果 } \max\{C(i-1, h, k), r_i\}(1 + \alpha_i) \le d_i \\ +\infty, & \text{否则} \end{cases}$$

$$C(i,h,k) = \begin{cases} \min\{\max\{C(i-1,h,k), r_i\}(1+\alpha_i) + f(i,h,k); C(i-1,h-1,k)\}, & \text{若 } J_i \in J^A, \\ \min\{\max\{C(i-1,h,k), r_i\}(1+\alpha_i) + f(i,h,k); C(i-1,h,k-1)\}, & \text{若 } J_i \in J^B. \end{cases}$$

最优解值: $h^* = \min\{h : C(n, h, k) < +\infty, k \le Q_B\}$.

容易证明算法 5 的时间复杂性为 $O(n n_A n_B)$.

4　结论

本文研究了带有简单线性恶化工件和释放时间的两个代理单机调度问题, 其中工件释放时间可以相同也可以不同. 目的是找到一个调度使得满足第二个代理的目标函数不超过一个给定上界的约束下, 最小化第一个代理的目标函数. 对于一般问题我们证明了问题的复杂性, 对于难解的问题我们针对原问题或者原问题的一些特殊情况提出了伪多项式时间或多项式时间算法. 在本文的研究基础上, 我们以后将对本文中证明是一般意义 NP 难的问题分析设计全多项式时间近似算法, 同时对其他的目标函数或者其他的机器调度环境作进一步的研究.

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 Ng C T, Cheng T C E, Yuan J J. A note on the complexity of the problem of two-agent scheduling on a single machine. *Journal of Combinatorial Optimization*, 2006, **12**(4): 387−394

4 Cheng T C E, Ng C T, Yuan J J. Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs. *Theoretical Computer Science*, 2006, **362**(1−3): 273−281

5 Agnetis A, Pacciarelli D, Pacifici A. Multi-agent single machine scheduling. *Annals of Operations Research*, 2007, **150**(1): 3−15

6 Cheng T C E, Ng C T, Yuan J J. Multi-agent scheduling on a single machine with max-form criteria. *European Journal of Operational Research*, 2008, **188**(2): 603−609

7 Lee K B, Choi B C, Leung J Y T, Pinedo M L. Approximation algorithms for multi-agent scheduling to minimize total

weighted completion time. *Information Processing Letters*, 2009, **109**(16): 913−917

8 Leung J Y T, Pinedo M L, Wan G. Competitive two agent scheduling and its applications. *Operations Research*, 2010, **58**(2): 458−469

9 Wan G H, Vakati S R, Leung J Y T, Pinedo M. Scheduling two agents with controllable processing times. *European Journal of Operational Research*, 2010, **205**(3): 528−539

10 Mor B, Mosheiov G. Scheduling problems with two competing agents to minimize minmax and minsum earliness measures. *European Journal of Operational Research*, 2010, **206**(3): 540−546

11 Mor B, Mosheiov G. Single machine batch scheduling with two competing agents to minimize total flowtime. *European Journal of Operational Research*, 2011, **215**(3): 524−531

12 Li S S, Yuan J J. Unbounded parallel-batching scheduling with two competitive agents. *Journal of Scheduling*, 2012, **15**(5): 629−640

13 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

14 Cheng T C E, Ding Q, Lin B M T. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, 2004, **152**(1): 1−13

15 Mosheiov G. Scheduling jobs under simple linear deterioration. *Computers and Operations Research*, 1994, **21**(6): 653−659

16 Ng C T, Li S S, Cheng T C E, Yuan J J. Preemptive scheduling with simple linear deterioration on a single machine. *Theoretical Computer Science*, 2010, **411**(40−42): 3578−3586

17 Liu P, Tang L X. Two-agent scheduling with linear deteriorating jobs on a single machine. *Lecture Notes in Computer Science*, 2008, **5092**: 642−650

18 Liu P, Tang L X, Zhou X Y. Two-agent group scheduling with deteriorating jobs on a single machine. *The International Journal of Advanced Manufacturing Technology*, 2010, **47**(5−8): 657−664

19 Liu P, Yi N, Zhou X Y. Two-agent single-machine scheduling problems under increasing linear deterioration. *Applied Mathematical Modelling*, 2011, **35**(5): 2290−2296

20 Liu Peng, Zhou Xiao-Ye, Rong Nan. Two-agent scheduling with a learning effect and deteriorating jobs. *Journal of Systems Engineering*, 2012, **27**(6): 841−846
(刘鹏, 周晓晔, 荣楠. 带有学习效应和恶化工件的双代理调度问题. 系统工程学报, 2012, **27**(6): 841−846)

21 Yin Y Q, Cheng S R, Wu C C. Scheduling problems with two agents and a linear non-increasing deterioration to minimize earliness penalties. *Information Sciences*, 2012, **189**: 282−292

22 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 and Operations Research*, 2012, **39**(12): 3062−3073

23 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

24 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

25 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 and Operations Research*, 2013, **40**(1): 353−361

26 Li D C, Hsu P H. Competitive two-agent scheduling with learning effect and release times on a single machine. *Mathematical Problems in Engineering*, 2013, **2013**, Article ID 754826, doi: 10.1155/2013/754826

27 Kung J Y, Chao Y P, Lee K I, Kang C C, Lin W C. Two-agent single-machine scheduling of jobs with time-dependent processing times and ready times. *Mathematical Problems in Engineering*, 2013, **2013**, Article ID 806325, DOI: 10.1155/2013/806325

28 Johnson D S. The NP-complete columns: an ongoing guide. *Journal of Algorithms*, 1981, **2**(4): 393−405

29 Ji M, He Y, Cheng T C E. Scheduling linear deteriorating jobs with an availability constraint on a single machine. *Theoretical Computer Science*, 2006, **362**(1−3): 115−126

30 Kononov A. Combinatorial complexity of scheduling jobs with simple linear processing times. *Diskretny Analiz i Issledovanie Operatsii*, 1996, **3**(2): 15−32 (in Russian)

赵晓丽 东北大学工业工程与物流优化研究所, 辽宁省制造系统与物流优化重点实验室博士研究生. 主要研究方向为生产调度与组合最优化.
E-mail: zhaoxiaoli824@163.com
(**ZHAO Xiao-Li** Ph. D. candidate at Liaoning Key Laboratory of Manufacturing System and Logistics, Institute of Industrial Engineering and Logistics Optimization, Northeastern University. Her research interest covers production scheduling and combinational optimization.)

唐立新 东北大学工业工程与物流优化研究所, 辽宁省制造系统与物流优化重点实验室教授. 主要研究方向为生产调度, 物流与供应链管理和组合最优化. 本文通信作者.
E-mail: lixintang@mail.neu.edu.cn
(**TANG Li-Xin** Professor at Liaoning Key Laboratory of Manufacturing System and Logistics, Institute of Industrial Engineering and Logistics Optimization, Northeastern University. His research interest covers production scheduling, logistics and supply chain management, and combinational optimization. Corresponding author of this paper.)