

共同工期下的总权误工单机双代理排序问题^{*}

张新功, 崔同欣

(重庆师范大学 数学科学学院, 重庆 401331)

摘要:【目的】研究共同工期下与总权误工相关的单机双代理排序问题。【方法】通过动态规划方法分析了双代理模型, 即在第 2 个代理的总误工工件个数不超过一个给定值的前提下, 使得第 1 个代理的总权误工最小。【结果】分别给出了最优性质、伪多项式时间算法以及时间复杂度分析。【结论】通过算例实验分析说明了算法的可行性。

关键词:排序; 两个代理; 总权误工; 动态规划算法

中图分类号:O223

文献标志码:A

文章编号:1672-6693(2022)01-0035-06

工件误工问题一直是排序论中的研究热点之一, Lawler^[1]较早对此问题展开讨论, 证明了单机情况下总权误工问题是强 NP 难的, Potts 等人^[2]给出了总权误工问题的分支定界算法。张新功等人^[3]考虑把资源分配的公平价格问题应用到单机双代理排序中, 这里的双代理就是两个代理具有各自的工件集, 公平竞争的安排在单台机器上加工自己的工件。陈秋宏等人^[4]讨论了工件具有固定加工区间, 目标函数为总误工的排序问题。此外, 在排序论的讨论中, 工件的工期是一项重要指标, 例如松弛工期模型(SLK)、共同工期模型(CON)等, Liu 等人^[5]讨论了学习效应的单机资源分配排序问题, 考虑了双重成本和总权资源消耗成本, 目的是同时确定最佳加工顺序和资源分配, 使得排序成本最小化。Yang 等人^[6]在多任务环境下, 解决了一个和工期分配有关的双代理排序问题, 证明了问题是 NP 难的, 设计了伪多项式动态规划算法, 以及给出了完全多项式近似方案。Lu 等人^[7]研究了具有工期分配相关的工件学习效应和资源分配问题, 证明了对总权资源消耗成本约束的排序成本问题可以在多项式时间中解决。Wang^[8]研究了与共同工期、松弛工期模型有关的排序问题, 目标函数是提前完工工件、误工工件的数量和工期成本的线性加权之和, 分别给出了在多项式时间算法。Wang 等人^[9]讨论了单机的 CON/SLK 工期分配和工件依赖学习效果的资源分配排序问题, 目标是确定工件序列和资源分配及总成本函数, 证明了问题的最优性质和多项式时间算法。Yuan^[10]利用等划分方法, 证明了 CDDWT 模型的 NP 难证明。Zhang 等人^[11]在单机双代理的情况下, 研究了当 B 代理的最大费用函数不超过一个给定值的情况下, A 代理的总权误工损失最小, 并提出了多项式时间算法。Lü 等人^[12]考虑了 CON、SLK 模型下与位置依赖有关的流水作业问题, 并证明了两种模型都可以在多项式时间内解决。

本文讨论了在代理 B 工件的总误工工件个数满足小于等于一个特定值, 且代理 A 工件具有共同工期, 目标函数是总权误工, 并提出了该问题的最优排序, 给出相应的时间复杂度分析以及伪多项式时间动态规划算法, 最后用算例的分析来说明了算法的可行性。

1 问题的描述

本文研究的问题模型描述如下:

现有两个代理 A 和 B, 所有工件需要在同一台机器上进行加工, 令 $X \in \{A, B\}$, 代理 A 有 n_A 个工件, 代理 B 有 n_B 个工件, 它们有各自的工件集 $J^X = \{J_1^X, J_2^X, \dots, J_{n_X}^X\}$ 。令 $p^X = \{p_1^X, p_2^X, \dots, p_{n_X}^X\}$ 。假设 $n = n_A + n_B$ 个工件从 $t=0$ 时刻开始加工, 且不可中断。那么 $p_j^X, d_j^X, w_j^X, C_j^X$ 分别表示代理 X 中工件的加工时间、工期、权重以及完

^{*} 收稿日期: 2021-06-28 修回日期: 2021-09-17 网络出版时间: 2022-02-09 13:52

资助项目: 国家自然科学基金重大项目(No. 11991022); 国家自然科学基金面上项目(No. 11971443); 重庆市科委项目(No. cstc2021jcyj-msxmX0229); 重庆市教委自然科学基金重大项目(No. KJZD-K202000501); 重庆市教委研究生重点教改项目(No. YJG182019)

第一作者简介: 张新功, 男, 教授, 博士后, 研究方向为排序论, E-mail: zhangxingong@cqu.edu.cn

网络出版地址: <https://kns.cnki.net/kcms/detail/50.1165.N.20220209.1124.004.html>

工时间, σ 是一个可行排序。本文主要考虑如下两个目标函数:

1) 总权误工: $\sum w_j^X T_j^X(\sigma) = \sum_{j=1}^{n_X} w_j^X T_j^X(\sigma)$, 其中 $T_j^X(\sigma) = \max\{C_j^X(\sigma) - d_j^X, 0\}$ 。

2) 总误工工件个数: $\sum U_j^X(\sigma) = \sum_{j=1}^{n_X} U_j^X(\sigma)$, 且当 $C_j^X(\sigma) - d_j^X \leq 0$ 时, $U_j^X(\sigma) = 0$; 当 $C_j^X(\sigma) - d_j^X > 0$ 时, $U_j^X(\sigma) = 1$ 。

排序问题的目标是找到一个序列, 使得在代理 B 的目标函数在满足不超过给定上界 $H_B (H_B \geq 0)$ 前提下, 代理 A 的目标函数达到最小化。本文研究问题模型可表示为: $1 | d_j^A = d^A | \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 。

下面的性质将被用于本文证明与推理中。

定义 1^[13] WSPT 规则: 把工件 $J_1, J_2, \dots, J_n$ 按照权重与加工时间之比 w_j/p_j 非减顺序排列的排序规则即为加权最短加工时间优先规则, 简称为 WSPT 规则。

定义 2^[13] WLPT 规则: 把工件 $J_1, J_2, \dots, J_n$ 按照权重与加工时间之比 w_j/p_j 非增顺序排列的排序规则, 简称为 WLPT 规则。

定义 3^[13] EDD 规则: 把工件 $J_1, J_2, \dots, J_n$ 按照工期 d_j 非减顺序排列的排序规则即为最早工期优先规则, 简称为 EDD 规则。

引理 1^[13] 对于问题 $1 || \sum U_j$, 非误工的工件按照 EDD 规则排列而得到的序列是一个最优排序。

2 总误工工件个数问题

引理 2 对于问题 $1 | d_j^A = d^A | \sum w_j^A T_j^A : \sum U_j^B \leq H_B$, 存在工件 J_i 和工件 J_j , 满足 $p_i \leq p_j, w_i \geq w_j$ 。若这两个工件属于同一个代理, 则存在一个最优序列, 使得工件 J_i 排在工件 J_j 之前加工。

证明 (反证法) 假设该问题存在一个最优序列 σ , 工件没有按照上述规则排列, 至少存在工件 J_i 和工件 J_j , 工件 J_i 在工件 J_j 之后加工。序列 σ' 为交换工件 J_i 和工件 J_j 的位置, 其余工件不变而得到新的序列, 下面证明序列 σ' 优于序列 σ 。

假设工件 J_i 和工件 J_j 属于同一个代理(不妨设为代理 A), 则:

1) 在序列 σ 中, 工件 J_i 和工件 J_j 都为提前完工工件, 有: $w_i^A T_i^A(\sigma) = w_j^A T_j^A(\sigma) = 0$ 。

由于 $d_i^A = d_j^A = d^A$, 即 $d^A \geq C_i^A(\sigma)$, 且 $C_i^A(\sigma) = C_i^A(\sigma')$, 故 $d^A \geq C_i^A(\sigma')$, 即在序列 σ' 中, 工件 J_i^A 和工件 J_j^A 仍非误工, 则有 $w_i^A T_i^A(\sigma') = w_j^A T_j^A(\sigma') = 0$ 。但是 $p_i^A \leq p_j^A$, 故交换过后序列 σ' 中工件 J_i^A 和工件 J_j^A 之间的工件的完工时间不超过在序列 σ 中的完工时间, 即序列 σ' 优于序列 σ , 这与序列 σ 为最优序列矛盾。

2) 在序列 σ 中, 工件 J_i 和工件 J_j 都为误工工件。由于在序列 σ 中, 工件 J_i^A 和工件 J_j^A 都是误工的, 且 $d_i^A = d_j^A = d^A$, 故 $w_i^A T_i^A(\sigma) = w_i^A \cdot (C_i^A(\sigma) - d^A)$, $w_j^A T_j^A(\sigma) = w_j^A \cdot (C_j^A(\sigma) - d^A)$ 。显然, 交换后工件 J_i^A 序列 σ' 中仍误工, 故有 $w_j^A T_j^A(\sigma) = w_j^A \cdot (C_j^A(\sigma') - d^A)$, 且 $C_i^A(\sigma') = C_j^A(\sigma')$, $C_j^A(\sigma) \leq C_j^A(\sigma')$, 而对于工件 J_i^A , 则有 $T_i^A(\sigma') = \max\{0, C_i^A(\sigma') - d^A\}$ 。故:

$$\begin{aligned} & w_i^A T_i^A(\sigma) + w_j^A T_j^A(\sigma) - w_i^A T_i^A(\sigma') - w_j^A T_j^A(\sigma') = \\ & w_i^A \cdot (C_i^A(\sigma) - d^A) + w_j^A \cdot (C_j^A(\sigma') - d^A) - w_i^A \cdot \max\{0, C_i^A(\sigma') - d^A\} - w_j^A \cdot (C_j^A(\sigma') - d^A) \geq \\ & w_i^A \cdot (C_i^A(\sigma) - d^A) + w_j^A \cdot (C_j^A(\sigma') - d^A) - w_i^A \cdot (C_i^A(\sigma') - d^A) - w_j^A \cdot (C_j^A(\sigma') - d^A) = \\ & w_i^A \cdot C_i^A(\sigma) + w_j^A \cdot C_j^A(\sigma) - w_i^A \cdot C_i^A(\sigma') - w_j^A \cdot C_j^A(\sigma') \geq (w_i^A - w_j^A) \cdot (C_j^A(\sigma') - C_i^A(\sigma')). \end{aligned}$$

由于 $w_i^A \geq w_j^A, C_j^A(\sigma') \geq C_i^A(\sigma')$, 故上式大于等于 0。

综上所述, 有 $w_i^A T_i^A(\sigma) + w_j^A T_j^A(\sigma) \geq w_i^A T_i^A(\sigma') + w_j^A T_j^A(\sigma')$, 且 $p_i^A \leq p_j^A$, 故两两交换后, 工件 J_i^A 和工件 J_j^A 之间工件的完工时间不会增加, 所以序列 σ' 优于序列 σ , 这与序列 σ 为最优序列矛盾。

3) 在序列 σ 中, 工件 J_j^A 提前完工, 工件 J_i^A 误工, 又由于 $d_i^A = d_j^A = d^A, p_i^A \leq p_j^A$, 故交换过后, 必有工件 J_j^A 误工, 工件 J_i^A 提前完工, 即 $w_i^A T_i^A(\sigma') = 0$, 则有:

$$\begin{aligned} & w_i^A T_i^A(\sigma) + w_j^A T_j^A(\sigma) - w_i^A T_i^A(\sigma') - w_j^A T_j^A(\sigma') = \\ & w_i^A \cdot (C_i^A(\sigma) - d^A) - w_j^A \cdot (C_j^A(\sigma') - d^A) = (w_i^A - w_j^A) \cdot (C_i^A(\sigma') - d^A). \end{aligned}$$

由于 $w_i^A \geq w_j^A, C_i^A(\sigma') \geq d^A$, 故上式大于等于 0。且由于 $p_i^A \leq p_j^A$, 故交换之后, 工件 J_i^A 和工件 J_j^A 之间工件的完工时间不会增加, 所以序列 σ' 优于序列 σ , 这与序列 σ 为最优序列矛盾。证毕

等划分问题 给出含有 $2t$ 个元素的正整数集合 $A = \{a_1, a_2, \dots, a_{2t}\}$ 以及正整数 B , 满足 $\sum_{j=1}^{2t} a_j = 2B$, 那么是否有一个关于集合 A 的划分, 把它划分成两个子集合 A_1 和 A_2 , 且 $|A_1| = |A_2| = t$, 并且 $\sum_{j \in A_1} a_j = \sum_{j \in A_2} a_j = B$?

定理 1 问题 $1 | d_j^A = d^A | \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 是 NP 难的。

证明 首先给出问题 $1 | d_j^A = d^A | \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 的一个划分实例: 对于 A 代理: $n_A = 2t, p_j^A = a_j, w_j^A = 1, d_j^A = B (j=1, 2, \dots, n_A)$, 限制值 $H_A = (1+t)B$; 对于 B 代理: $n_B = 1, p_1^B = B, d_1^B = 2B$, 限制值 $H_B = 0$ 。目标函数满足 $\sum w_j^A T_j^A \leq H_A, \sum U_j^B \leq H_B$ 。

接下来证明存在一个可行排序 σ , 让划分问题有解的同时, 使目标函数也满足限制要求。假设有一个可行排序 σ , 使得划分问题有解, 既有划分 A_1 和 $A_2, |A_1| = |A_2| = t$, 使得 $\sum_{j \in A_1} a_j = \sum_{j \in A_2} a_j = B$ 。

在序列 σ 中, 集合 $\{J_j^A | j \in A_1\}$ 中 A 代理的工件排在 B 代理工件之前加工, 剩下的 A 代理的工件, 即集合 $\{J_j^A | j \in A_2\}$ 中 A 代理的工件排在 B 代理工件之后加工, 于是有: $\sum_{\{J_j^A | j \in A_1\}} p_j^A = \sum_{j \in A_1} a_j = B, \sum_{\{J_j^A | j \in A_2\}} p_j^A = \sum_{j \in A_2} a_j = B$, 即集合 $\{J_j^A | j \in A_1\}$ 中 A 代理的工件在区间 $[0, B]$ 上加工, B 代理的工件 J_1^B 在区间 $[B, 2B]$ 上加工, 集合 $\{J_j^A | j \in A_2\}$ 中 A 代理的工件在区间 $[2B, 3B]$ 上加工, 那么 A 代理中工件的总权误工为:

$$\sum_{j=1}^{2t} w_j^A T_j^A = \sum_{j=t+1}^{2t} w_j^A T_j^A = \sum_{j=t+1}^{2t} T_j^A = \sum_{j=t+1}^{2t} (tB + p_j^A) = tB + B = (t+1)B = H_A。$$

同时 B 代理中的工件 J_1^B 满足 $\sum U_j^B = 0 = H_B$, 故序列 σ 是一个可行排序。

相应地, 如果 $\sum w_j^A T_j^A \leq H_A, \sum U_j^B \leq H_B$ 成立, 那么给定一个可行的排序 σ , 由 $C_{\max}(\sigma) = p_1^A + p_2^A + \dots + p_{2t}^A + p_1^B = 3B$ 可得, 若 A 代理的工件都在区间 $[C_1^B(\sigma), 3B]$ 上加工, 则全部误工, 所以有 $C_1^B(\sigma) \geq p_1^B = B = d^A$, 即 $(t+1)B \geq \sum_{j=t+1}^{2t} w_j^A T_j^A = \sum_{j=t+1}^{2t} T_j^A \geq 3B - C_1^B + t d^A = 3B - C_1^B + tB \geq (t+1)B$, 即 $C_1^B(\sigma) = 2B$ 。

又因为 $p_1^B = B$, 故 B 代理中的工件在区间 $[B, 2B]$ 上加工, 同时 A 代理中集合 $\{J_j^A | j \in A_1\}$ 中的工件位于区间 $[0, B]$ 上加工, 集合 $\{J_j^A | j \in A_2\}$ 中的工件位于区间 $[2B, 3B]$ 上加工。

综上所述, 原问题是一般意义下的 NP 难问题。证毕

引理 3 对于问题 $1 | d_j = d | \sum w_j T_j$ 存在一个最优序列 σ , 当它的加工时间与权重满足反一致关系, 即 $p_i \leq p_j$ 时, 有 $w_i \geq w_j$, 那么误工工件按照 WSPT 规则进行排列。

证明 假设满足上述问题的最优序列为 σ_2 , 误工工件没有按照 WSPT 规则排列, 那么至少存在两个相邻近的工件 J_i 和工件 J_j , 工件 J_i 在工件 J_j 之前加工, 且 $p_i \leq p_j, w_i \geq w_j, \frac{w_i}{p_i} \geq \frac{w_j}{p_j}$, 而序列 σ'_2 为交换工件 J_i 和工件 J_j 位置, 并且保持其他工件加工位置不变而得到的序列, 记 t_2 为工件 J_i 在序列 σ_2 中开始加工的时间, 也即为工件 J_j 在序列 σ'_2 中开始加工的时间, 记工件 J_i 与工件 J_j 之间的工件加工时间之和为 L_2 。

因为 $p_i \leq p_j, d_i = d_j = d$, 故交换过后工件 J_i 和工件 J_j 在序列 σ'_2 中为误工工件, 即:

$$T_i(\sigma'_2) = \max\{C_i(\sigma'_2) - d, 0\} = C_i(\sigma'_2) - d, T_j(\sigma'_2) = \max\{C_j(\sigma'_2) - d, 0\} = C_j(\sigma'_2) - d。$$

故有:

$$\begin{aligned} & w_i T_i(\sigma_2) + w_j T_j(\sigma_2) - w_i T_i(\sigma'_2) - w_j T_j(\sigma'_2) = \\ & w_i \cdot (C_i(\sigma_2) - d) + w_j \cdot (C_j(\sigma_2) - d) - w_i \cdot (C_i(\sigma'_2) - d) - w_j \cdot (C_j(\sigma'_2) - d) = \\ & w_i \cdot C_i(\sigma_2) + w_j \cdot C_j(\sigma_2) - w_i \cdot C_i(\sigma'_2) - w_j \cdot C_j(\sigma'_2) = \\ & w_i \cdot (t_2 + p_i) + w_j \cdot (t_2 + p_i + L_2 + p_j) - w_i \cdot (t_2 + p_i + L_2 + p_j) - w_j \cdot (t_2 + p_i) = \\ & w_j \cdot (p_i + L_2) - w_i \cdot (p_j + L_2) \geq (w_j p_i - w_i p_j) + (w_j - w_i) \cdot L_2。 \end{aligned}$$

由于 $p_i \leq p_j, w_i \geq w_j, \frac{w_i}{p_i} \geq \frac{w_j}{p_j}$, 且 $L_2 \geq 0$, 故 $w_i T_i(\sigma_2) + w_j T_j(\sigma_2) \geq w_i T_i(\sigma'_2) + w_j T_j(\sigma'_2)$, 这与序列 σ_2 为最优序

列矛盾。

证毕

3 问题的动态规划算法

引理 4 对于问题 1 $|d_j^A = d^A| \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 存在一个时间复杂度为 $O(n^2(p^A + p^B)n_B H_B)$ 的伪多项式时间算法。

证明 假设代理 A 中工件按照 WSPT 规则排列,代理 B 中工件按照规则 EDD 规则排列, $f(y, t, x_1, e_1)$ 表示代理 A 的工件总权误工,且满足以下条件:已加工了 j 个代理 A 的工件 $J_1^A, J_2^A, \dots, J_j^A$, 已加工了 i 个代理 B 中的工件,即 $J_1^B, J_2^B, \dots, J_i^B$, 且 i, j 分别满足 $1 \leq j \leq n_A, 1 \leq i \leq n_B$ 。同时 y 表示已加工的代理 A 工件与代理 B 工件的个数之和, t 表示已加工的代理 A 的工件以及代理 B 的非误工工件的加工时间和, x_1 表示已加工的代理 B 中工件非误工的工件个数, e_1 表示已加工的代理 B 中件的误工工件个数。

该动态规划方法的基本思想是考虑当前序列中最后一个加工的工件是 J_j^A 或者是工件 J_i^B , 分以下几种情况讨论:

1) 最后一个工件为 J_j^A 。

当前已加工的代理 A 中的工件以及代理 B 中的非误工工件的加工时间和为 $t - p_j^A$, 代理 A 对于目标函数的增加值为 $w_j^A \cdot \max\{t - d^A, 0\}$, 代理 B 中的误工工件个数不变为 e_1 , 对于目标函数的贡献为 0。

2) 最后一个工件为 J_i^B , 且工件 J_i^B 非误工。

此时工件 J_i^B 对于目标函数的贡献为 0, 且代理 B 中非误工工件个数为 $x_1 - 1$, 当前已加工的代理 A 中的工件以及代理 B 中的非误工工件的加工时间和为 $t - p_i^B$ 。

3) 最后一个工件为 J_i^B , 且工件 J_i^B 误工。

利用 Moore 算法在前 j 个代理 A 中工件 $J_1^A, J_2^A, \dots, J_j^A$ 和前 $i-1$ 个代理 B 中工件 $J_1^B, J_2^B, \dots, J_{i-1}^B$ 即前 $y-1$ 个代理 A、B 中工件的最优序列中, 将所有按时完工的代理 B 工件移至工件 J_i^B 中, 将加工时间最长的一个工件 J_m^B 加入到误工的代理 B 工件的集合之中, 其余工件前移, 由于移除的工件 J_m^B 加工时间最长, 故代理 A 工件的总权误工时间不会增加, 为 $t - p_i^B + p_m^B$, 代理 B 中工件的非误工工件个数为 $x_1 + 1$, 误工工件个数为 $e_1 - 1$ 。

问题 1 $|d_j^A = d^A| \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 的动态规划算法叙述如下:

初始条件: $f(0, 0, 0, 0) = 0$ 。

$$\text{递推关系式为: } f(y, t, x_1, e_1) = \min \begin{cases} f(y-1, t-p_j^A, x_1, e_1) + w_j^A \cdot \max\{t-d^A, 0\} \\ f(y-1, t-p_i^B, x_1-1, e_1) \\ f(y-1, t-p_i^B+p_m^B, x_1+1, e_1-1) \end{cases},$$

最优函数值为: $\min f(n, p^A + p^B, n_B, H_B)$ 。

算法的目的是为了找到问题 1 $|d_j^A = d^A| \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 的一个最优排序, 使得代理 A 的总权误工最小, 通过选取适当的参数, 利用上述给出的动态规划算法, 可以得出与之相对应的 $f(y, t, x_1, e_1)$ 的值, 若得到的 $f(y, t, x_1, e_1)$ 的值为一个有限实数 R , 那么这个有限实数 R 是问题 1 $|d_j^A = d^A| \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 的最小加权总误工。

下面讨论该算法的时间复杂度:

由于 $y \in [0, n_A + n_B], t \in [0, P^A + P^B], x_1 \in [0, n^B], e \leq H_B$, 故该动态规划算法的时间复杂度为 $O(n^2(p^A + p^B)n_B H_B)$ 。

证毕

接下来, 通过一组数据实验来对该算法进行进一步的说明。

算例 对于问题 1 $|d_j^A = d^A| \sum w_j^A T_j^A : \sum U_j^B \leq H_B$, 已知工件 $J^A = \{J_1^A, J_2^A, J_3^A\}, J^B = \{J_1^B, J_2^B\}, d_j^A = d^A = 5$, 且代理 A、B 中的工件在机器上加工的信息见表 1, 当 $H_B = 1$ 时, 求该问题的最小总权误工。

分析 代理 A 中的工件按照 WSPT 规则排列, 代理 B 中的工件按照 EDD 规则排列。假设现有 M 和 N 两个集合, 而 $\langle M, N \rangle$ 代表代理 A、B 中工件的一个可行排序, 集合 M 中工件从 0 时刻以从左往右的顺

表 1 工件的基本信息

Tab. 1 The information of jobs

工件	取值		
J_j^A	J_1^A	J_2^A	J_3^A
w_j^A	1	2	3
p_j^A	3	4	2
J_j^B	J_1^B	J_2^B	
p_j^B	1	2	
d_j^B	2	5	

序开始加工,集合 N 中的工件都视为代理 B 中的误工工件,放到集合 M 之后加工。变量 (ϖ, y, t, x_1, e_1) 代表当前工件的加工状态,其中: ϖ 代表此加工状态下代理 A 中工件的总权误工, y 代表代理 A 和代理 B 的总加工工件个数, t 代表当前已加工的代理 A 中的工件以及代理 B 中非误工工件的加工时间和, x_1 代表当前已加工的代理 B 中提前完工的工件个数, e_1 代表当前已加工的代理 B 中误工的工件个数, F_i 则代表当前加工的第 i 个工件所对应的工作状态,则 $(\varpi, y, t, x_1, e_1) \in F_i$, 且有 $i = y$ 。

$$\begin{aligned}
 F_1 &= \{ \underbrace{(0, 1, 2, 0, 0)}_{\langle J_3^A; \emptyset \rangle}, \underbrace{(0, 1, 0, 0, 1)}_{\langle \emptyset; J_1^B \rangle}, \underbrace{(0, 1, 1, 1, 0)}_{\langle J_1^B; \emptyset \rangle} \}, \\
 F_2 &= \{ \underbrace{(2, 2, 6, 0, 0)}_{\langle J_3^A J_2^A; \emptyset \rangle}, \underbrace{(0, 2, 3, 1, 0)}_{\langle J_3^A J_1^B; \emptyset \rangle}, \underbrace{(0, 2, 3, 1, 0)}_{\langle J_1^B J_3^A; \emptyset \rangle}, \underbrace{(0, 2, 2, 0, 1)}_{\langle J_3^A; J_1^B \rangle} \}, \\
 F_3 &= \{ \underbrace{(6, 3, 9, 0, 0)}_{\langle J_3^A J_2^A J_1^A; \emptyset \rangle}, \underbrace{(2, 3, 7, 1, 0)}_{\langle J_3^A J_2^A J_1^B; \emptyset \rangle}, \underbrace{(4, 3, 7, 1, 0)}_{\langle J_3^A J_1^B J_2^A; \emptyset \rangle}, \underbrace{(0, 3, 5, 2, 0)}_{\langle J_3^A J_1^B J_2^B; \emptyset \rangle}, \underbrace{(4, 3, 7, 1, 0)}_{\langle J_1^B J_3^A J_2^A; \emptyset \rangle}, \\
 &\quad \underbrace{(0, 3, 5, 2, 0)}_{\langle J_1^B J_3^A J_2^B; \emptyset \rangle}, \underbrace{(2, 3, 6, 0, 1)}_{\langle J_1^B J_3^A J_2^A; J_1^B \rangle}, \underbrace{(0, 3, 2, 0, 2)}_{\langle J_1^B J_3^A J_2^B; J_1^B \rangle}, \underbrace{(0, 3, 3, 1, 1)}_{\langle J_1^B J_3^A J_2^A; J_1^B \rangle}, \underbrace{(0, 3, 3, 1, 1)}_{\langle J_1^B J_3^A J_2^B; J_1^B \rangle} \}, \\
 F_4 &= \{ \underbrace{(2, 4, 9, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^B; \emptyset \rangle}, \underbrace{(4, 4, 9, 2, 0)}_{\langle J_3^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(4, 4, 9, 2, 0)}_{\langle J_1^B J_3^A J_2^A J_2^B; \emptyset \rangle}, \underbrace{(8, 4, 9, 2, 0)}_{\langle J_1^B J_3^A J_2^B J_2^A; \emptyset \rangle}, \underbrace{(8, 4, 9, 2, 0)}_{\langle J_1^B J_3^A J_2^A J_2^A; \emptyset \rangle}, \underbrace{(8, 4, 9, 2, 0)}_{\langle J_1^B J_3^A J_2^B J_2^B; \emptyset \rangle}, \underbrace{(6, 4, 9, 0, 1)}_{\langle J_3^A J_2^A J_1^B; J_1^B \rangle}, \\
 &\quad \underbrace{(2, 4, 7, 1, 1)}_{\langle J_3^A J_2^A J_1^B; J_2^B \rangle}, \underbrace{(4, 4, 7, 1, 1)}_{\langle J_3^A J_2^A J_1^B; J_2^A \rangle}, \underbrace{(4, 4, 7, 1, 1)}_{\langle J_3^A J_2^A J_1^B; J_2^B \rangle}, \underbrace{(2, 4, 6, 0, 2)}_{\langle J_3^A J_2^A J_1^B; J_2^A \rangle}, \underbrace{(6, 4, 10, 1, 0)}_{\langle J_3^A J_2^A J_1^B; J_2^B \rangle}, \underbrace{(7, 4, 10, 1, 0)}_{\langle J_3^A J_2^A J_1^B; J_2^A \rangle}, \underbrace{(9, 4, 10, 1, 0)}_{\langle J_3^A J_2^A J_1^B; J_2^B \rangle}, \\
 &\quad \underbrace{(9, 4, 10, 1, 0)}_{\langle J_3^A J_2^A J_1^B; J_2^A \rangle}, \underbrace{(9, 4, 10, 1, 0)}_{\langle J_3^A J_2^A J_1^B; J_2^B \rangle} \}, \\
 F_5 &= \{ \underbrace{(6, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(7, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(9, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(11, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(9, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \\
 &\quad \underbrace{(15, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(9, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(11, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(15, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(15, 5, 12, 2, 0)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(6, 5, 11, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \\
 &\quad \underbrace{(8, 5, 11, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(12, 5, 11, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(12, 5, 11, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(6, 5, 9, 0, 2)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(6, 5, 10, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(7, 5, 10, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \\
 &\quad \underbrace{(9, 5, 10, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle}, \underbrace{(9, 5, 10, 1, 1)}_{\langle J_3^A J_2^A J_1^B J_2^A J_2^B; \emptyset \rangle} \}.
 \end{aligned}$$

由分析可得:当 $H_B = 1$ 时,问题 $1 \mid d_j^A = d^A \mid \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 的总权误工为 9, $\langle J_1^B J_3^A J_2^A J_1^A; J_2^B \rangle$ 是一个最优排序。

4 结束语

本文研究 A 和 B 两个代理在单机情况下的排序问题,对于 $1 \mid d_j^A = d^A \mid \sum w_j^A T_j^A : \sum U_j^B \leq H_B$ 模型,列举了模型的部分最优性质,并采用动态规划思想,给出了伪多项式时间算法,以及算法的时间复杂度分析,最后,通过数值算例,证明了该算法的有效性。在今后的讨论中可以考虑其他目标函数的双代理排序问题,或者把该问题扩展到不同类型的机器上。基于该问题的 NP 难性质,也可以考虑利用智能优化算法来进行求解。

参考文献:

- [1] LAWLER E L. A pseudo-polynomial algorithm for sequencing jobs to minimize total tardines [J]. Annals of Discrete Mathematics, 1977, 1: 331-342.
 - [2] POTTS C N, Van Wassenhove L N. A branch and bound algorithm for the total weighted tardiness problem [J]. Operations Research, 1985, 33(2): 363-377.
 - [3] 张新功,刘甲玉,崔同欣. 误工工件个数和最大费用函数的单机双代理 KS 公平定价问题 [J]. 重庆师范大学学报(自然科学版), 2020, 37(1): 16-21.
- ZHANG X G, LIU J Y, CUI T X. KS price of fairness in two-agent single-machine scheduling problem to minimize the number of tardy jobs and maximum cost function [J]. Journal of Chongqing Normal University (Natural Science), 2020, 37(1): 16-21.

- [4] 陈秋宏,张新功. 带有固定区间的单机双代理可中断总误工问题[J]. 运筹学学报, 2019, 23(1): 61-71.
CHEN Q H, ZHANG X G. Two-agent preemptive scheduling of jobs with fixed time windows problem about total tardiness on single machine[J]. Operations Research Transactions, 2019, 23(1): 61-71.
- [5] LIU W, JIANG C. Due-date assignment scheduling involving job-dependent learning effects and convex resource allocation[J]. Engineering Optimization, 2020, 52(1): 74-89.
- [6] YANG Y, YIN Y, SHI Q. Two competing agents with due date assignment scheduling under multitasking environment[C]// 2019 International Conference on Industrial Engineering and Systems Management (IESM). Piscataway: IEEE, 2019: 1-6.
- [7] LU Y Y, WANG T T, WANG R Q, et al. A note on due-date assignment scheduling with job-dependent learning effects and convex resource allocation[J]. Engineering Optimization, 2020, 53(7): 1273-1281.
- [8] WANG W L. Single-machine due-date assignment scheduling with generalized earliness-tardiness penalties including proportional setup times[EB/OL]. (2021-04-28)[2021-06-28]. <http://link.springer.com/article/10.1007/s12190-021-01555-4#citeas>.
- [9] WANG J B, GENG X N, LIU L, et al. Single machine CON/SLK due date assignment scheduling with controllable processing time and job-dependent learning effects[J]. The Computer Journal, 2018, 61(9): 1329-1337.
- [10] YUAN J J. The NP-hardness of the single machine common due date weighted tardiness problem[J]. Journal of Systems Science and Mathematical Sciences, 1992(4): 328-333.
- [11] ZHANG X G, WANG Y. Two-agent scheduling problems on a single-machine to minimize the total weighted late work[J]. Journal of Combinatorial Optimization, 2017, 33(3): 945-955.
- [12] LÜ D Y, WANG J B. Study on proportionate flowshop scheduling with due-date assignment and position-dependent weights [J]. Optimization Letters, 2021(1): 1-9.
- [13] PETER M. Scheduling algorithms[M]. New York: Springer-Verlag, 2007.

Operations Research and Cybernetics

Single-Machine Two-Agent Scheduling Problem with Common Due Date to Minimize Total Weighted Tardiness

ZHANG Xingong, CUI Tongxin

(College of Mathematics Sciences, Chongqing Normal University, Chongqing 401331, China)

Abstract: [Purposes] It aims to study the scheduling problems with two competing agents on a single machine about total weighted tardiness under common due date. [Methods] It solves the following model by dynamic programming method: the total weighted tardiness of the first agent is minimized under the premise that the total number of tardiness jobs of the second agent does not exceed a given value. [Findings] The optimal properties, pseudo-polynomial time algorithm and time-complexity analysis are presented, respectively. [Conclusions] Finally, the feasibility of the algorithm is illustrated by an example.

Keywords: scheduling; two-agent scheduling; total weighted tardiness; dynamical programming algorithm

(责任编辑 黄 颖)