

最小化总完工时间且工件可拒绝的单机双代理多任务排序问题*

张新功, 叶 爽

(重庆师范大学 数学科学学院, 重庆 401331)

摘要:研究了工件可拒绝的双代理单机多任务排序问题。所谓多任务环境即是指当某个工件(主工件)在加工时,会被其余未完成加工的工件(这里称为等待工件)所打扰。而双代理之间不可互相打扰,它们共同使用单台机器来完成各自工件的加工,第 1 个代理目标函数为最小化总完工时间,第 2 个代理最大完工时间不超过某个上界。给定总拒绝费用的允许上界,每个工件有 2 个选择:接受或拒绝。排序目的是为了第 2 个代理最大完工时间不超过某个上界的条件下,要使得第 1 个代理目标函数最小化。由于该问题是 NP 难问题,为该问题给出最优性质刻画和复杂度分析,以及设计了伪多项式动态规划算法。并用算例实验来说明了算法的可行性。

关键词:双代理排序;可拒绝;总完工时间;多任务排序

中图分类号:O223

文献标志码:A

文章编号:1672-6693(2024)04-0061-07

在实际生产中,每个客户可能有不同的任务和不同的目标,从而逐渐形成了多代理排序问题。Agnētis 等人^[1]首次提出双代理单机排序问题,排序的目标是找到 1 个序列,第 2 个代理目标函数值不超过某个值的情况下,最小化第 1 个代理目标函数值,考虑的目标函数有最大费用函数、总加权完工时间及误工工件数等。Cheng 等人^[2]考虑了目标函数是总权误工工件数和总完工时间的单机多代理模型,证明问题是 NP 难,提出问题的多项式或伪多项式时间算法。Leung 等人^[3]考虑双代理在单机和平行机上的排序问题,包括具有达到时间的可中断问题、固定区间问题、可用性约束问题。陈秋宏等人^[4]研究带有固定区间的双代理单机排序问题,是在 Leung 等人^[3]基础上扩展到总误工问题,利用分块原则提出伪多项式动态规划算法,并给出了固定区间大于加工时间情况下的时间复杂度分析。崔同欣等人^[5]研究了共同工期下的总权误工单机双代理排序问题,目标函数通过动态规划方法分析了模型,给出了最优性质、伪多项式时间算法以及时间复杂度分析。在可拒绝排序的研究中,Feng 等人^[6]研究了单机双代理可拒绝问题。目标函数有最大完工时间、最大延迟等,给出一些问题的伪多项式时间动态规划算法、2-近似算法和完全多项式时间近似方案。苗翠霞等人^[7]考虑了工件具有退化效应的 2 台机器流水作业可拒绝排序问题,考虑的目标是被接受工件的最大完工时间加上被拒绝工件的总拒绝费用之和。证明了问题是 NP 难的,并提出了 1 个动态规划算法。

多任务处理被定义为“同时执行多个任务”。Hall 等人^[8]首次将多任务概念引入排序领域,研究了多任务情况下的单机排序问题。Li 等人^[9]考虑了多任务情况下的单机双代理排序问题,目标函数包括最大费用函数、总完工时间和(加权)误工工件数,目的是为 2 个代理的所有任务确定 1 个可行排序,给相关问题提出了多项式算法或伪多项式算法。Min 等人^[10]考虑了多任务平行排序问题,为相关问题证明是多项式可解的,并给出特殊情况的多项式算法。Wang 等人^[11]考虑具有多个代理的多任务排序模型,计算考虑的每个问题的复杂性状态,并为它们设计解决方案。徐晨^[12]研究了工件可拒绝的双代理单机多任务排序问题,双代理不可打扰,考虑了最大完工时间为目标的模型,设计了多项式动态规划算法、近似算法以及完全多项式近似方案。徐晨等人^[13]研究工件可拒绝的单机多任务排序问题,目标函数有最大完工时间、总完工时间等,给出了伪多项式动态规划算法。Oron^[14]研究了第 1 个代理的排序目标是最大完工时间或总完工时间,第 2 个代理的排序目标是最大完工时间。

* 收稿日期:2024-06-17 网络出版时间:2024-10-12T16:38

资助项目:国家自然科学基金重大项目(No. 11991022);最优化理论与方法及其应用”创新创业示范团队项目(No. CQYC20210309536);重庆市自然科学基金创新发展联合基金项目(No. CSTB2023NSCQ-LZX0005)

第一作者简介:张新功,男,教授,博士,博士生导师,研究方向为排序论、供应链管理,E-mail:zhangxingong@cqnu.edu.cn;通信作者:叶爽,女,研究方向为排序论,E-mail:1418471514@qq.com。

网络出版地址:https://link.cnki.net/urlid/50.1165.n.20241012.1400.002

对于各自代理拥有各自拒绝费用上界和双代理拥有总的拒绝费用上界这 2 类问题均为 NP 难问题,并设计了伪多项式动态规划算法。

多任务双代理现象在日常生活、物流配送等方面都有见到。例如在工厂里,工厂同时服务于 2 个客户,他们的工件没有关联,某台机器用于加工不同类型的工件,工件也有特殊性,在每次加工工件时需一定时间维持未加工工件状态,此外因资源有限和客户的需求不同,工厂有权因为工件花费时间高且利润低而拒绝工件。本文研究了双代理间工件不可打扰、可拒绝的单机双代理多任务排序问题。在允许总拒绝费用不超过某个值的前提下,第 1 个代理的目标函数为总完工时间,保证第 2 个代理的最大完工时间不超过某个上界,尽可能使得第 1 个代理的目标函数值更小。本文给出了最优性质刻画,并提出了 1 个伪多项式时间动态算法,最后通过算例实验说明了算法的可行性。

1 问题描述及符号表示

存在工件集合 $J^X = \{J_1^X, J_2^X, \dots, J_{n_X}^X\}$ 来自于代理 $X, X \in \{A, B\}$, 代理 X 有 n_X 个工件, 且 $\sum_{X \in \{A, B\}} n_X = n$ 。

代理 A 和代理 B 需要在同一机器上加工, 所有工件在 0 时刻到达, 工件要么被加工要么被拒绝。对于任意问题实例的工件排序是由所有接受加工工件组成的, 对可行排序 π , 定义 $p_j^X(\pi), C_j^X(\pi)$ 分别为代理 X 的第 j 个工件的加工时间、完工时间。若工件 J_j^X 被拒绝, 将会产生 e_j^X 的费用。 A^X 为代理 X 的接受工件集合, R^X 为代理 X 的拒绝工件集合。令代理 X 的总加工时间为 P^X , 代理 X 的总拒绝费用为 E^X , 则所有工件总加工时间为:

$$P = P^A + P^B = \sum_{k=1}^{n_A} p_k^A + \sum_{k=1}^{n_B} p_k^B,$$

$$\text{总拒绝费用为 } E = E^A + E^B = \sum_{k=1}^{n_A} e_k^A + \sum_{k=1}^{n_B} e_k^B.$$

在多任务环境下, 有 2 个类型的工件: 主工件和等待工件。假设代理 A 与代理 B 的工件不能相互打扰。当主工件 J_j^X 在代理 X 中处于第 r 个位置加工时, 其余未加工的工件集合 S_j^X 称为 J_j^X 的等待工件, 且必须打扰主工件有且仅有一次。由于工件 J_j^X 作为第 r 个主工件, 意味着此工件已打扰了 $r-1$ 个工件。将打扰时间函数定义为 $\eta_j(\overline{p_j^X}) = \mu \overline{p_j^X}$, 其中 $0 < \mu < 1$, $\overline{p_j^X}$ 表示工件的剩余处理时间, 有 $\overline{p_j^X} = (1-\mu)^{r-1} p_j^X$, 则工件的所有等待工件的打扰时间总和为 $\mu \sum_{J_y^X \in S_j^X} \overline{p_y^X}$ 。除此之外, 处理主工件 J_j^X 时, 每 2 个等待工件将产生切换时间。假设任意 2 个等待工件的切换时间相同且都等于 1。

综上, 工件 J_j^X 的总加工时间可表示为: $(1-\mu)^{r-1} p_j^X + \mu \sum_{J_y^X \in S_j^X} \overline{p_y^X} + (n_X - r)$ 。

考虑了双代理间不可打扰可拒绝多任务排序问题, 不仅知道拒绝工件集合而且需要对接受工件进行排序。目标是在总拒绝费用的限制下最小化代理 A 的总完工时间, mt 表示在多任务情况下, 使用三参数表示法记为:

$$1 \mid mt, \sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \mid \sum C_j^A.$$

此处假设 $M \leq \sum_{k=1}^{n_A} p_k^A + \sum_{k=1}^{n_B} p_k^B + \frac{n_A(n_A-1)}{2} + \frac{n_B(n_B-1)}{2}$, $E \leq \sum_{k=1}^{n_A} e_k^A + \sum_{k=1}^{n_B} e_k^B$, 否则约束 $\sum_{J_j \in R^A \cup R^B} e_j \leq E$,

$C_{\max}^B \leq M$ 成为无效约束。

下面是相关定义将用于本文的证明和推理中。

定义 1 块: 没有被干扰的, 连续加工的工件集称为块。

2 问题的动态规划算法

不考虑多任务情形下, 文献[14]证明了 $1 \mid \sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \mid \sum C_j^A$ 为 NP 难, 这意味着多任务情

形下 $1 \mid mt, \sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \mid \sum C_j^A$ 也是 NP 难的。在这小节, 将介绍 1 个伪多项式动态规划算法。

为了方便求解,将给出下列最优性质。

定理 1 最优排序中,任意 2 个接受加工的工件之间不空闲且第 1 个工件加工时间为 0。

证明 假设在最优排序中,第 1 个工件加工时间在 t 处,将第 1 个工件移到 0 时刻,目标函数将会减少,故在最优排序中第 1 个工件加工时间为 0;同理可证最优排序中,任意 2 个接受加工的工件之间不空闲。证毕

定理 2 在最优排序中,代理 B 的接受工件连续加工,在一个块中。代理 A 的接受工件,至多有 2 个块并且一个在代理工件 B 之前,一个在代理 B 之后。

证明 假设代理 B 的接受工件不连续加工,主工件存在一个可行排序 $\sigma: (\sigma^1, J_i^A, J_j^B, J_{i+1}^A, J_{j+1}^B, \sigma^2)$, 其中 σ^1 与 σ^2 为 σ 的子排序。设工件 J_i^A 为代理 A 的第 i 个主工件, m 为代理 A 的加工数,工件 J_j^B 为代理 B 的第 j 个主工件, n 为代理 B 的加工数, t 为排序 σ 中工件 J_i^A 的开始时间。由于代理 A 与代理 B 并不互相干扰,交换 J_j^B 与 J_{i+1}^A 后,令 $\sigma': (\sigma^1, J_i^A, J_{i+1}^A, J_j^B, J_{j+1}^B, \sigma^2)$, 他们在各自代理的主工件位置并不改变,即所带来的切换时间不变,故 $C_{j+1}^B(\sigma) = C_{j+1}^B(\sigma')$, 则有:

$$C_i^A(\sigma) + C_{i+1}^A(\sigma) = t + p_i^A + (m-i) + C_j^B(\sigma) + p_{i+1}^A + (m-i-1),$$

交换 J_j^B 与 J_{i+1}^A 后,则得到:

$$C_i^A(\sigma') + C_{i+1}^A(\sigma') = t + p_i^A + (m-i) + C_i^A(\sigma') + p_{i+1}^A + (m-i-1)。$$

由于 $C_i^A(\sigma') = C_i^A(\sigma) < C_j^B(\sigma)$, 故目标函数减少,于是 σ' 排序比 σ 排序优,通过两两交换,代理 B 连续加工,为 1 个块。代理 A 的接受工件,至多有 2 个块并且 1 个在代理工件 B 之前,1 个在代理 B 之后。证毕

定理 3 在最优排序中,代理 B 的接受工件任意排序加工,代理 A 的接受工件按照 SPT 规则加工。

证明 由定理 2,代理 B 的接受工件连续加工,其中任意 2 个接受工件两两交换并不改变 $C_{\max}^B$ 的值与目标函数值 $\sum C_j^A$, 故代理 B 的接受工件任意排序加工。

假设一对相邻接受工件 J_i^A 与 J_j^A , 工件 J_i^A 在工件 J_j^A 之前处理,且 $p_i^A < p_j^A$, J_i^A 为代理 A 的第 i 个主工件, m 为代理 A 的加工数。则在工件 J_i^A 之后处理了 $m-i$ 个代理 A 的工件,令它的排序为 σ 。交换工件 J_i^A 与 J_j^A 后获得新排序 σ' , 设 t 为排序 σ 中工件 J_i^A 的开始时间,则有:

$$C_i^A(\sigma) = t + p_i^A + (m-i), C_j^A(\sigma) = t + p_i^A + (m-i) + p_j^A + (m-i-1)$$

和

$$C_j^A(\sigma') = t + p_j^A + (m-i), C_i^A(\sigma') = t + p_j^A + (m-i) + p_i^A + (m-i-1)。$$

可得 $C_j^A(\sigma) = C_i^A(\sigma')$ 。因此有:

$$C_i^A(\sigma) - C_j^A(\sigma') = p_i^A - p_j^A < 0。$$

所以代理 A 的接受工件按照 SPT 规则加工。

接下来介绍动态规划算法求解此问题,首先根据 LPT 规则,对代理 A 的工件进行重新标号,即 $p_1^A \geq p_2^A \geq \dots \geq p_{n_A}^A$ 。

设 $f(i, j, t, l_1, l_2, m, e)$ 为处理了工件集 $\{J_1^A, J_2^A, \dots, J_i^A\}$ 和 $\{J_1^B, J_2^B, \dots, J_j^B\}$ 后代理 A 的总完工时间, t 代表代理 B 的最大完工时间, l_1 为代理 A 在代理 B 前接受工件数量, l_2 为代理 A 在代理 B 后接受工件数量, m 为代理 B 接受工件数量, e 为总的拒绝费用。

在当下情形,有以下 7 种情况。

假设工件 J_{j+1}^B 来自代理 B, 受限于代理 B 的最大完工时间以及总拒绝费用,接着分 4 种情况:

1) 若满足 $t + p_{j+1}^B + m \leq M, e + e_{j+1}^B \leq E$ 。

此时有 2 种方案:

(a) 工件 J_{j+1}^B 安排在已经接受加工的代理 B 工件的首位, 工件 J_{j+1}^B 将产生正常加工时间 p_{j+1}^B 和工件 J_{j+1}^B 带来 m 的切换时间, 代理 A 的总完工时间将增加 $l_2(p_{j+1}^B + m)$;

(b) 拒绝工件 J_{j+1}^B , 总拒绝费用增加 e_{j+1}^B , 代理 A 的总完工时间不变。

2) 若满足 $t + p_{j+1}^B + m \leq M, e + e_{j+1}^B > E$ 。

此时有 1 种方案: 工件 J_{j+1}^B 安排在已经接受加工的代理 B 工件的首位, 代理 A 的总完工时间将增加 $l_2(p_{j+1}^B + m)$ 。

3) 若满足 $t + p_{j+1}^B + m > M, e + e_{j+1}^B \leq E$ 。

此时有 1 种方案:拒绝工件 J_{j+1}^B ,总拒绝费用增加 e_{j+1}^B ,代理 A 的总完工时间不变。

4) 若满足 $t + p_{j+1}^B + m > M, e + e_{j+1}^B > E$ 。

此时,不管工件是否接受加工均不可行,即已考虑的工件集合是不可行排序。

上述 4 种情况下,工件 J_{j+1}^B 有 2 种选择,若工件 J_{j+1}^B 接受加工,代理 B 的最大完工时间会增加,以至于代理 A 的总完工时间将增加,增加值为块 B 后的已经接受加工的代理 A 工件均增加工件 J_{j+1}^B 正常加工时间和工件 J_{j+1}^B 带来的切换时间,即增加值为 $l_2(p_{j+1}^B + m)$ 。若工件 J_{j+1}^B 被拒绝,总拒绝费用增加 e_{j+1}^B ,但代理 A 的总完工时间不变。

假设接受工件 J_{i+1}^A 来自于代理 A,分析产生条件有如下 2 种情况:

5) 若满足 $t + p_{i+1}^A + (l_1 + l_2) \leq M$,其中 $l_1 = 0$ 。

此时有 3 种方案:

(a) 工件 J_{i+1}^A 放在时刻 0 处,工件 J_{i+1}^A 将产生正常加工时间 p_{i+1}^A 和工件 J_{i+1}^A 带来 l_2 的切换时间,代理 A 的总完工时间将增加 $(p_{i+1}^A + l_2)(l_2 + 1)$;

(b) 工件 J_{i+1}^A 放在时刻 t 处,工件 J_{i+1}^A 将产生正常加工时间 p_{i+1}^A 和工件 J_{i+1}^A 带来 l_2 的切换时间,代理 A 的总完工时间将增加 $t + (l_2 + 1)(p_{i+1}^A + l_2)$;

(c) 拒绝工件 J_{i+1}^A ,总拒绝费用增加 e_{i+1}^A ,代理 A 的总完工时间不变。

6) 若满足 $t + p_{i+1}^A + (l_1 + l_2) \leq M$,其中 $l_1 \neq 0$ 。

此时有 2 种方案:

(a) 工件 J_{i+1}^A 放在时刻 0 处,工件 J_{i+1}^A 将产生正常加工时间 J_{i+1}^A 和工件 J_{i+1}^A 带来 $l_1 + l_2$ 的切换时间,代理 A 的总完工时间将增加 $(p_{i+1}^A + l_1 + l_2)(l_1 + l_2 + 1)$;

(b) 拒绝工件 J_{i+1}^A ,总拒绝费用增加 e_{i+1}^A ,代理 A 的总完工时间不变。

7) 若满足 $t + p_{i+1}^A + (l_1 + l_2) > M$ 。

此时有 2 种方案:

(a) 工件 J_{i+1}^A 放在时刻 t 处,代理 A 的总完工时间将增加 $t + (l_2 + 1)(p_{i+1}^A + l_2)$;

(b) 拒绝工件 J_{i+1}^A ,总拒绝费用增加 e_{i+1}^A ,代理 A 的总完工时间不变。

根据上述情况分析,工件 J_{i+1}^A 有 3 种选择。如果工件 J_{i+1}^A 在 0 时刻开始加工,总拒绝费用不变,代理 B 的最大完工时间会增加,代理 A 总完工时间将增加,工件 J_{i+1}^A 的完工时间 $p_{i+1}^A + l_1 + l_2$ (工件 J_{i+1}^A 正常加工时间 p_{i+1}^A 和工件 J_{i+1}^A 带来的切换时间 $l_1 + l_2$)和已经接受加工的 $l_1 + l_2$ 个代理 A 工件均增加 $p_{i+1}^A + l_1 + l_2$,即代理 A 总完工时间将增加 $(p_{i+1}^A + l_1 + l_2)(l_1 + l_2 + 1)$ 。如果工件 J_{i+1}^A 在 t 时刻加工(注意:为了保证最优性质的 SPT 序,在 t 时刻加工的前提是 $l_1 = 0$,即块 B 前没有代理 A 的工件接受加工),在总拒绝费用不变,代理 B 的最大完工时间不变,代理 A 总完工时间将增加 $t + (l_2 + 1)(p_{i+1}^A + l_2)$ 。如果工件 J_{i+1}^A 被拒绝,总拒绝费用增加 e_{i+1}^A ,但代理 A 总完工时间不变。证毕

结合工件来自代理 A 或代理 B 的情况分析,该算法最终得到的最优解对应的最优排序是满足定理 1、定理 2、定理 3。

根据上述分析,给出如下算法。

算法 H 初始条件: $f(0,0,0,0,0,0,0)=0$,

$$f(i,j,t,l_1,l_2,m,e)=+\infty, l_1+l_2>i \text{ 或 } m>j。$$

递推函数:

$$f(i,j,t,l_1,l_2,m,e)=\min \begin{cases} f(i,j-1,t-p_j^B-(m-1),l_1,l_2,m,e)+l_2 \cdot (p_j^B+m-1), t+p_j^B+m-1 \leq M, \\ f(i-1,j,t-(p_i^A+l_1+l_2-1),l_1-1,l_2,m,e)+(p_i^A+l_1+l_2-1) \cdot (l_1+l_2), t+p_i^A+l_1+l_2-1 \leq M, \\ f(i-1,j,t,l_1,l_2-1,m,e)+t+l_2 \cdot (p_i^A+l_2-1), l_1=0, \\ f(i-1,j,t,l_1,l_2,m,e-e_i^A), e+e_i^A \leq E, \\ f(i,j-1,t,l_1,l_2,m,e-e_j^B), e+e_j^B \leq E. \end{cases}$$

最优值: $\min f(n_A, n_B, M, l_1, l_2, m, E)$ 。

定理 4 问题 1 $\left| mt, \sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \right| \sum C_j^A$ 时间复杂度为 $O(nn_A n_B ME)$ 。

证明 代理 A 工件 SPT 排序, 能在 $O(n_A \log n_A)$ 得到, 算法每次迭代都是在常数时间内进行的, i 以 n_A 为界, j 以 n_B 为界。 t 以 $P^A + P^B + \frac{n_A(n_{A-1})}{2} + \frac{n_B(n_{B-1})}{2}$ 为界, 同时 l_1 与 l_2 以 n_A 为界, e 以 $E^A + E^B$ 为界。故算法的总体时间复杂度为 $O(nn_A n_B ME)$ 。 证毕

算例 对于问题 1 $\left| mt, \sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \right| \sum C_j^A$, 假设有 2 个代理 A 和 B, 已知代理 A 和 B 中的工件信息表分别如下表所示, 当 $M=12, E=7$ 时, 求该问题的最小总完工时间。

表 1 代理 A 的工件信息

Tab. 1 Jobs of agent A

J_j^A	J_1^A	J_2^A	J_3^A
p_j^A	5	7	10
e_j^A	4	6	5

表 2 代理 B 的工件信息

Tab. 2 Jobs of agent B

J_j^B	J_1^B	J_2^B	J_3^B
p_j^B	6	4	10
e_j^B	4	3	8

分析 先按照 LPT 规则排列代理 A 的工件, 设有 A, R 这 2 个集合, 而集合 $\langle A; R \rangle$ 代表代理 A, B 中所有工件的一个可行排序, 从 0 时刻开始加工集合 A 中的工件, 集合 R 中的工件为代理 A, B 中所有拒绝加工的工件。变量 (c, t, e) 为当前工件的加工状态, 其中 c 为代理 B 工件的最大完工时间, t 为代理 A 的总完工时间, e 为代理 A 与代理 B 的拒绝费用, 每次考虑 1 个工件会更新 1 次变量, 接受加工工件所带来的时间包括正常加工时间和切换时间。 F_i 则表示第 i 个当前加工的工件对应的工作状态。

$$\begin{aligned}
 F_1 = & \{ \underbrace{(0, 10, 0)}_{\langle J_3^A; \emptyset \rangle}, \underbrace{(6, 0, 0)}_{\langle J_1^B; \emptyset \rangle}, \underbrace{(4, 0, 0)}_{\langle J_2^B; \emptyset \rangle}, \underbrace{(10, 0, 0)}_{\langle J_3^B; \emptyset \rangle}, \underbrace{(0, 0, 5)}_{\langle \emptyset; J_3^A \rangle}, \underbrace{(0, 0, 4)}_{\langle \emptyset; J_1^B \rangle}, \underbrace{(0, 0, 3)}_{\langle \emptyset; J_2^B \rangle} \}, \\
 F_2 = & \{ \underbrace{(0, 26, 0)}_{\langle J_2^A J_3^A; \emptyset \rangle}, \underbrace{(0, 10, 6)}_{\langle J_3^A; J_2^A \rangle}, \underbrace{(6, 16, 0)}_{\langle J_1^B J_3^A; \emptyset \rangle}, \underbrace{(4, 14, 0)}_{\langle J_2^B J_3^A; \emptyset \rangle}, \underbrace{(0, 10, 4)}_{\langle J_3^A; J_1^B \rangle}, \underbrace{(0, 10, 3)}_{\langle J_3^A; J_2^B \rangle}, \underbrace{(11, 0, 0)}_{\langle J_2^B J_1^B; \emptyset \rangle}, \\
 & \underbrace{(6, 0, 3)}_{\langle J_1^B; J_2^B \rangle}, \underbrace{(6, 0, 5)}_{\langle J_1^B; J_3^A \rangle}, \underbrace{(4, 0, 5)}_{\langle J_2^B; J_3^A \rangle}, \underbrace{(11, 0, 0)}_{\langle J_1^B J_2^B; \emptyset \rangle}, \underbrace{(4, 0, 4)}_{\langle J_2^B; J_1^B \rangle}, \underbrace{(10, 20, 0)}_{\langle J_3^B J_3^A; \emptyset \rangle}, \underbrace{(10, 0, 5)}_{\langle J_3^B; J_3^A \rangle}, \underbrace{(10, 0, 4)}_{\langle J_3^B; J_1^B \rangle}, \underbrace{(10, 0, 3)}_{\langle J_3^B; J_2^B \rangle}, \underbrace{(0, 7, 5)}_{\langle J_2^A; J_3^A \rangle}, \underbrace{(0, 0, 7)}_{\langle \emptyset; J_1^B J_2^B \rangle} \}, \\
 F_3 = & \{ \underbrace{(0, 47, 0)}_{\langle J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(0, 26, 4)}_{\langle J_2^A J_3^A; J_1^A \rangle}, \underbrace{(6, 38, 0)}_{\langle J_1^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(4, 34, 0)}_{\langle J_2^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(10, 46, 0)}_{\langle J_3^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(0, 26, 4)}_{\langle J_2^A J_3^A; J_1^B \rangle}, \underbrace{(0, 26, 3)}_{\langle J_2^A J_3^A; J_2^B \rangle}, \underbrace{(0, 22, 6)}_{\langle J_1^A J_3^A; J_2^A \rangle}, \\
 & \underbrace{(6, 16, 6)}_{\langle J_1^B J_3^A; J_2^A \rangle}, \underbrace{(4, 14, 6)}_{\langle J_2^B J_3^A; J_2^A \rangle}, \underbrace{(10, 20, 6)}_{\langle J_3^B J_3^A; J_2^A \rangle}, \underbrace{(6, 38, 0)}_{\langle J_1^B J_2^B J_3^A; \emptyset \rangle}, \underbrace{(11, 21, 0)}_{\langle J_2^B J_1^B J_3^A; \emptyset \rangle}, \underbrace{(6, 16, 3)}_{\langle J_1^B J_3^A; J_2^B \rangle}, \underbrace{(11, 21, 0)}_{\langle J_1^B J_2^B J_3^A; \emptyset \rangle}, \underbrace{(4, 34, 0)}_{\langle J_2^B J_2^A J_3^A; \emptyset \rangle}, \\
 & \underbrace{(10, 20, 4)}_{\langle J_3^B J_3^A; J_1^B \rangle}, \underbrace{(10, 20, 3)}_{\langle J_3^B J_3^A; J_2^B \rangle}, \underbrace{(4, 14, 4)}_{\langle J_2^B J_3^A; J_1^B \rangle}, \underbrace{(0, 10, 7)}_{\langle J_3^A; J_1^B J_2^B \rangle}, \underbrace{(11, 0, 5)}_{\langle J_2^B J_1^B; J_3^A \rangle}, \underbrace{(6, 13, 5)}_{\langle J_1^B J_2^A; J_3^A \rangle}, \underbrace{(11, 0, 5)}_{\langle J_1^B J_2^B; J_3^A \rangle}, \underbrace{(11, 7, 5)}_{\langle J_2^A J_2^B; J_3^A \rangle}, \underbrace{(4, 11, 5)}_{\langle J_2^B J_2^A; J_3^A \rangle}, \\
 & \underbrace{(10, 17, 5)}_{\langle J_3^B J_2^A; J_3^A \rangle}, \underbrace{(10, 0, 7)}_{\langle J_3^B; J_1^B J_2^B \rangle}, \underbrace{(0, 19, 5)}_{\langle J_1^A J_2^A; J_3^A \rangle}, \underbrace{(12, 30, 0)}_{\langle J_2^A J_2^B J_3^A; \emptyset \rangle} \}, \\
 F_4 = & \{ \underbrace{(6, 65, 0)}_{\langle J_1^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(4, 59, 0)}_{\langle J_2^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(10, 77, 0)}_{\langle J_3^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(0, 47, 4)}_{\langle J_1^A J_2^A J_3^A; J_1^B \rangle}, \underbrace{(0, 47, 3)}_{\langle J_1^A J_2^A J_3^A; J_2^B \rangle}, \underbrace{(6, 38, 4)}_{\langle J_1^B J_2^A J_3^A; J_1^A \rangle}, \underbrace{(4, 34, 4)}_{\langle J_2^B J_2^A J_3^A; J_1^A \rangle}, \underbrace{(10, 46, 4)}_{\langle J_3^B J_2^A J_3^A; J_1^A \rangle}, \\
 & \underbrace{(0, 26, 7)}_{\langle J_2^A J_3^A; J_1^B J_2^B \rangle}, \underbrace{(11, 48, 0)}_{\langle J_2^B J_1^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(6, 38, 3)}_{\langle J_1^B J_2^A J_3^A; J_2^B \rangle}, \underbrace{(11, 48, 0)}_{\langle J_1^B J_2^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(11, 55, 0)}_{\langle J_1^A J_2^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(10, 46, 3)}_{\langle J_3^B J_2^A J_3^A; J_2^B \rangle}, \underbrace{(10, 46, 4)}_{\langle J_3^B J_2^A J_3^A; J_1^B \rangle}, \underbrace{(6, 34, 6)}_{\langle J_1^B J_1^A J_3^A; J_2^A \rangle}, \underbrace{(4, 30, 6)}_{\langle J_2^B J_1^A J_3^A; J_2^A \rangle}, \\
 & \underbrace{(10, 42, 6)}_{\langle J_3^B J_1^A J_3^A; J_2^A \rangle}, \underbrace{(11, 21, 6)}_{\langle J_2^B J_1^B J_3^A; J_2^A \rangle}, \underbrace{(12, 28, 6)}_{\langle J_1^A J_1^B J_3^A; J_2^A \rangle}, \underbrace{(11, 21, 6)}_{\langle J_1^B J_2^B J_3^A; J_2^A \rangle}, \underbrace{(10, 26, 6)}_{\langle J_1^A J_2^B J_3^A; J_2^A \rangle}, \underbrace{(4, 34, 4)}_{\langle J_2^B J_2^A J_3^A; J_1^B \rangle}, \underbrace{(10, 20, 7)}_{\langle J_3^B J_3^A; J_1^B J_2^B \rangle}, \underbrace{(12, 30, 4)}_{\langle J_2^A J_2^B J_3^A; J_1^B \rangle}, \underbrace{(0, 26, 7)}_{\langle J_2^A J_3^A; J_1^B J_2^B \rangle}, \\
 & \underbrace{(11, 18, 5)}_{\langle J_2^B J_1^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(12, 25, 5)}_{\langle J_1^A J_1^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(6, 31, 5)}_{\langle J_1^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(11, 18, 5)}_{\langle J_1^B J_2^B J_2^A J_3^A; \emptyset \rangle}, \underbrace{(10, 39, 5)}_{\langle J_3^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(6, 31, 5)}_{\langle J_1^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(4, 27, 5)}_{\langle J_2^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(12, 30, 4)}_{\langle J_2^A J_2^B J_3^A; J_1^B \rangle}, \underbrace{(10, 23, 5)}_{\langle J_2^A J_3^A; J_1^B J_2^B \rangle} \}.
 \end{aligned}$$

$$\begin{aligned}
F_5 = \{ & \underbrace{(11, 80, 0)}_{\langle J_2^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(6, 65, 3)}_{\langle J_1^B J_1^A J_2^A J_3^A; J_2^B \rangle}, \underbrace{(11, 80, 0)}_{\langle J_1^B J_2^B J_1^A J_2^A J_3^A; \emptyset \rangle}, \underbrace{(4, 59, 4)}_{\langle J_2^B J_1^A J_2^A J_3^A; J_1^B \rangle}, \underbrace{(10, 77, 3)}_{\langle J_3^B J_1^A J_2^A J_3^A; J_2^B \rangle}, \underbrace{(10, 77, 4)}_{\langle J_3^B J_1^A J_2^A J_3^A; J_1^B \rangle}, \\
& \underbrace{(0, 47, 7)}_{\langle J_1^A J_2^A J_3^A; J_1^B J_2^B \rangle}, \underbrace{(11, 48, 4)}_{\langle J_2^B J_1^B J_2^A J_3^A; J_1^A \rangle}, \underbrace{(6, 38, 7)}_{\langle J_1^B J_2^A J_3^A; J_1^A J_2^B \rangle}, \underbrace{(11, 48, 4)}_{\langle J_1^B J_2^B J_2^A J_3^A; J_1^A \rangle}, \underbrace{(10, 46, 7)}_{\langle J_3^B J_2^A J_3^A; J_1^A J_2^B \rangle}, \underbrace{(11, 55, 4)}_{\langle J_1^A J_2^B J_2^A J_3^A; J_1^B \rangle}, \\
& \underbrace{(10, 46, 7)}_{\langle J_3^B J_2^A J_3^A; J_1^B J_2^B \rangle}, \underbrace{(11, 44, 6)}_{\langle J_2^B J_1^B J_1^A J_2^A; J_2^A \rangle}, \underbrace{(11, 44, 6)}_{\langle J_1^B J_2^A J_3^A; J_2^A \rangle}, \underbrace{(11, 41, 5)}_{\langle J_2^B J_1^B J_1^A J_2^A; J_3^A \rangle}, \underbrace{(11, 41, 5)}_{\langle J_1^B J_2^B J_1^A J_2^A; J_3^A \rangle} \}, \\
F_6 = \{ & \underbrace{(10, 77, 7)}_{\langle J_3^B J_1^A J_2^A J_3^A; J_2^B J_1^B \rangle} \}.
\end{aligned}$$

由分析得, 当 $M = 12, E = 7$ 时, $1 \mid mt$, $\sum_{J_j \in R^A \cup R^B} e_j \leq E, C_{\max}^B \leq M \mid \sum C_j^A$ 的最小总完工时间是 77,

$\langle J_3^B J_1^A J_2^A J_3^A; J_2^B J_1^B \rangle$ 是最优排序。

3 总结

本文考虑了工件可拒绝的双代理单机多任务排序问题, 并且双代理不互相干扰, 切换时间为单位时间 1 的情况下, 在保证代理 B 最大完工时间不超过某个上界的前提下, 给定一定的总拒绝费用, 使得代理 A 目标函数总完工时间最小化, 给出模型的最优性质, 设计伪多项式动态规划算法, 时间复杂度为 $O(nm_A n_B ME)$ 。未来的研究可以考虑多代理排序、其他目标函数和机器类型。

参考文献:

- [1] AGNETIS A, MIRCHANDANI P B, PACCIARELLI D, et al. Scheduling problems with two competing agents[J]. Operations Research, 2004, 52(2): 229-242.
- [2] CHENG T C E, NG C T, YUAN J J. Multi-agent scheduling on a single machine with max-form criteria[J]. European Journal of Operational Research, 2008, 188(2): 603-609.
- [3] LEUNG J Y T, PINEDO M, WAN G. Competitive two-agent scheduling and its applications[J]. Operations Research, 2010, 58(2): 458-469.
- [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 a single machine[J]. Operations Research Transaction, 2019, 23(1): 61-71.
- [5] 张新功, 崔同欣. 共同工期下的总权误工单机双代理排序问题[J]. 重庆师范大学学报(自然科学版), 2022, 39(1): 35-40.
ZHANG X G, CUI T X. Single-machine two-agent scheduling problem with common due date to minimize total weighted tardiness[J]. Journal of Chongqing Normal University (Natural Science), 2022, 39(1): 35-40.
- [6] FENG Q, FAN B, LI S, et al. Two-agent scheduling with rejection on a single machine[J]. Applied Mathematical Modelling, 2015, 39(3/4): 1183-1193.
- [7] 苗翠霞, 孟凡晓. 基于退化效应的两台机器流水作业可拒绝排序[J]. 运筹学学报, 2017, 21(2): 66-72.
MIAO C X, MENG F X. Two-machine flow-shop scheduling with deterioration and rejection[J]. Operations Research Transactions, 2017, 21(2): 66-72.
- [8] HALL N G, LEUNG J Y T, LI C L. The effects of multitasking on operations scheduling[J]. Production and Operations Management, 2015, 24(8): 1248-1265.
- [9] LI S S, CHEN R X, TIAN J. Multitasking scheduling problems with two competitive agents[J]. Engineering Optimization, 2020, 52(11): 1940-1956.
- [10] JI M, ZHANG W, LIAO L, et al. Multitasking parallel-machine scheduling with machine-dependent slack due-window assignment[J]. International Journal of Production Research, 2019, 57(6): 1667-1684.
- [11] WANG D, YU Y, YIN Y, et al. Multi-agent scheduling problems under multitasking[J]. International Journal of Production Research, 2021, 59(12): 3633-3663.
- [12] 徐晨. 可拒绝的双代理多任务排序问题[EB/OL]. (2024-04-29)[2024-06-17]. https://sysmath.cjoe.ac.cn/jweb_xtkxysx/CN/10.12341/jssms23727.
XU C. Two-agent multitasking scheduling with job rejection[EB/OL]. (2024-04-29)[2024-06-17]. https://sysmath.cjoe.ac.

cn/jweb_xtkxysx/CN/10.12341/jssms23727.

[13] 徐晨, 徐寅峰, 郑斐峰, 等. 工件可拒绝的单机多任务排序问题研究[J]. 系统科学与数学, 2022, 42(8): 2198-2206.

XU C, XU Y F, ZHENG F F, et al. Multitasking scheduling problems with job rejection on single machine[J]. Journal of System Science and Mathematical Science Chinese Series, 2022, 42(8): 2198-2206.

[14] ORON D. Two-agent scheduling problems under rejection budget constraints[J]. Omega, 2021, 102: 102313.

The Single-Machine Two-Agent Multi-Task Scheduling Problem with Rejection to Minimize the Total Completion Time

ZHANG Xingong, YE Shuang

(School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China)

Abstract: It investigates the single-machine multi-task scheduling problem with two-agent which jobs can be rejected. In a multitasking environment, when a job (referred to here as the main job) is being processed, it can be disturbed by other unfinished jobs (referred to here as waiting jobs). The two agents cannot interfere with each other and share a single machine to process their respective jobs. The objective function for the first agent is to minimize the total completion time, while the maximum completion time for the second agent does not exceed a specified upper bound. Given an allowable upper bound of total rejection cost, each job has two options: accepted or rejected. The aim is to minimize the objective function of the first agent under the condition that the maximum completion time of the second agent does not exceed the specified upper bound. As the proposed problem is NP-hard, it provides the optimal property characterizations and complexity analysis and designs a pseudo-polynomial dynamic programming algorithm. Numerical experiments are conducted to demonstrate the feasibility of the proposed algorithm.

Keywords: two-agent scheduling; job rejection; total completion time; multi-task scheduling

特约稿作者介绍



张新功, 教授, 博士后, 博士生导师, 重庆市科学技术带头人。现任中国运筹学会理事、中国运筹学会排序专业委员会常务理事、中国仿真学会智能仿真优化与调度专委会委员、重庆市运筹学会常务理事、重庆市工业与应用数学学会理事、重庆市系统工程学会理事。长期从事组合最优化、排序论、物流与供应链管理等研究。在 *Expert Systems with Applications*、*Annals of operations research*、*Computers & Industrial Engineering*、*Swarm and Evolutionary Computation*、*Journal of the Operations Research Society of China*、系统科学与数学等国内外学术期刊发表论文 95 篇, 其中 SCI 收录 42 篇, 在科学出版社等出版学术专著 2 部, 申请发明专利 2 项。主持国家自然科学基金青年基金项目、中国博士后科学基金特别资助项目和面上项目、重庆市自然科学基金创新发展联合基金项目、重庆市自然科学基金面上项目等省部级以上基金项目。作为主研人员参与国家自然科学基金重大项目、科技部国家重点研发计划“数学和应用研究”重点专项、国家自然科学基金面上项目等。



1954-2024
重庆师范大学 70 周年校庆
弦歌七秩 桃李芬芳

(责任编辑 黄 颖)