

关于总加权提前损失的两个代理单机排序问题^{*}

张新功, 栗 苹

(重庆师范大学 数学科学学院, 重庆 401331)

摘要:【目的】研究与总加权提前损失有关两个代理单机排序的问题。【方法】第 1 个代理工件的工期相同, 目标函数是最小化总加权提前损失; 第 2 个代理的目标函数是最大正则函数, 它的特殊情形为最大完工时间。目标是寻找一个排序, 使得在满足第 2 个代理目标可行的情况下, 第 1 个代理目标函数值最小。【结果】利用背包问题证明了该问题是一般意义下 NP 难的。【结论】最终给出了总加权提前损失有关两个代理单机排序问题的一个最优算法, 并证明了该算法是拟多项式时间可解的。

关键词: 排序; 两个代理; 总加权提前损失

中图分类号: O223

文献标志码: A

文章编号: 1672-6693(2021)01-0064-05

分布式决策领域的最新发展引起了学者们对多代理排序问题的关注, 即不同的代理分享同一公共加工资源, 每个代理都希望根据自身的工件来最小化成本函数。Agnētis 等人^[1]最早提出了两个竞争代理的单机排序问题, 其中有两个重要模型: 约束优化模型和帕累托优化模型; 他们还给出了 9 个在不同的目标函数组合下的两个竞争代理排序问题的时间复杂性结果, 目标函数包括最大正则费用函数、总加权完工时间和误工工件个数。Cheng 等人^[2]考虑了目标函数为最大正则以及完工时间的多代理竞争单机排序问题; 对于可行性模型, 他们证明了多项式时间可解; 对于最小化模型, 他们的研究则表明问题是一般意义下强 NP 难的, 当代理是给定的常数时, 可以在拟多项式时间内可解。Leung 等人^[3]研究了两个竞争代理在单机和恒同机上的排序问题, 包含带有到达时间的中断模型以及在固定区间加工的排序模型, 总结了目标为最大正则函数、总加权完工时间以及总误工工件个数的相关结论, 解决了文献[1]中遗留的有关一个代理目标为总完工时间, 而另一个代理目标为总误工工件个数的复杂性问题。Perez-Gonzalez 等人^[4]对 2014 年前的多目标排序问题(包括干涉工件、竞争工件、多代理排序等)进行了详细的综述, 并对多目标排序的定义和名词进行了统一, 为该类问题提供了一个系统性的框架; 他们还通过研究相关问题的时间复杂性和解决方案的程序设计, 对有关研究成果系统地进行了评述与分类, 在讨论最新研究进展基础上指出了后续的研究方向。Zhang 等人^[5]考虑了目标函数为总加权误工损失和最大正则函数的两个代理单机排序问题, 提出了多项式时间算法和伪多项式时间算法; 之后 Zhang 等人^[6]又对文献[5]中关于目标函数为总加权误工损失和最大正则函数两个代理单机排序的问题进行了改进, 提出了相应的拟多项式时间算法。马露等人^[7]还将两个竞争代理模型扩展到总误工损失问题, 研究了一个代理的总完工时间小于等于一个常数的条件下最小化另一个代理的总误工损失问题, 证明了该问题是拟多项式时间可解的; 他们还讨论了其中一个代理的总误工工件数小于等于一个常数的条件下, 最小化另一个代理的总误工损失问题, 并给出了相应的动态规划算法。张新功等人^[8]研究了一个代理的到达时间与工期满足一致关系且目标函数为总误工或最大误工、而另一个代理的目标函数为总误工工件个数或为关于完工时间的正则函数或者它所有工件排在固定区间的情形, 提出了总误工问题的一个拟多项式时间动态规划算法, 给出了最大误工问题时间复杂度的证明。Ben-Yehoshua 等人^[9]首次提出了单机总提前损失问题, 工件将根据工期前完成部分工件的持续时间进行惩罚, 一个工件的最大惩罚等于它的加工时间; 他们还利用二分法证明了单机总提前损失问题是 NP 难的, 并推导了拟

* 收稿日期: 2020-07-11 修回日期: 2020-08-31 网络出版时间: 2021-03-10 16:07

资助项目: 国家自然科学基金重大项目(No. 11991020; No. 11991022); 国家自然科学基金面上项目(No. 11971443); 重庆市研究生教育教育改革研究重点项目(No. yjg182019); 重庆师范大学研究生科研创新项目(No. YKC20040)

第一作者简介: 张新功, 男, 教授, 博士, 研究方向为组合最优化, E-mail: zxcg7980@163.com; 通信作者: 栗苹, 女, E-mail: 569526039@qq.com

网络出版地址: <https://kns.cnki.net/kcms/detail/50.1165.N.20210310.1129.006.html>

多项式动态规划算法,它的时间复杂度为 $O(n \sum p_j)$ 。Sterna 等人^[10]对在两个平行机上具有相同工期的最大化总提前损失问题进行了研究,提出了多项式时间近似方案,并探讨了误工损失和提前损失之间的关系。Chen 等人^[11]研究了在平行机最大化总提前损失的问题,理顺了该问题已获得的结果,并就固定数量的机器和相同工期提出了一种在拟多项式时间内运行的动态规划方法,以及基于该方法的完全多项式时间逼近方案。本文在上述文献的基础上研究了与总加权提前损失有关的两个代理单机排序问题:代理 A 中工件的工期相同,它的目标函数为总加权提前损失;代理 B 的目标函数是最大化正则函数,它的特殊情形为最大完工时间。利用背包问题证明了当代理 B 的目标函数是最大完工时间时,该问题是一般意义下 NP 难的,最后就该问题给出了一个拟多项式时间可解的算法。

1 预备知识

存在竞争代理 A 和 B,两个代理中的工件需在同一台机器上进行加工,但在机器上相同时间点有且只有 1 个工件在进行加工。它们分别有各自的工件集 $\{J_1^X, J_2^X, \dots, J_{n_X}^X\}$, 其中 $X = \{A, B\}$, 代理 X 中有 n_X 个工件,它们从 $t=0$ 时刻开始加工。 p_j^X, w_j^X, d_j^X 分别表示工件 J_j^X 的加工时间、权重以及工期。设 π 是该问题的可行排序, $S_j^X(\pi), C_j^X(\pi)$ 分别表示在可行排序 π 下工件 J_j^X 的开始加工时间和完工时间。根据 Ben-Yehoshua^[9]对工件 J_j^X 在可行排序 π 下的提前损失 $E_j^X(\pi)$ 定义为:若 $C_j^X(\pi) \leq d_j^X$, 则 $E_j^X(\pi) = p_j^X$, 称工件 J_j^X 为提前损失工件。若 $S_j^X(\pi) \leq d_j^X \leq C_j^X(\pi)$, 在情形 1 即工件 J_j^X 被中断、 p_{j1}^X 为工件 J_j^X 在 d_j^X 之前加工的部分且 p_{j2}^X 为工件 J_j^X 在 d_j^X 之后加工的部分时,则 $E_j^X(\pi) = p_{j1}^X$; 在情形 2 即工件 J_j^X 未被中断时,则 $E_j^X(\pi) = d_j^X - S_j^X(\pi)$, 称工件 J_j^X 为部分提前损失工件。若 $d_j^X \leq S_j^X(\pi)$, 则 $E_j^X(\pi) = 0$, 称工件 J_j^X 为误工工件。

考虑的目标函数有:1) 总加权提前损失,即 $\sum w_j^X E_j^X(\pi) = \sum_{j=1}^{n_X} w_j^X E_j^X(\pi)$ 。2) 最大正则函数,即 $f_{\max}^X(\pi) = \max_{i=1,2,\dots,n_X} \{f_i^X(C_j^X(\pi))\}$, $f_j^X(\cdot)$ 是关于工件 J_j^X 完工时间的一个非减函数。该函数的特殊情形最大完工时间为 $C_{\max}^B(\pi)$ 。

本文的目标是寻找一种排序,使得在满足代理 B 的目标函数不大于给定的上界的情况下,使得代理 A 中工件的工期相同,总加权提前损失最小。利用 Graham 等人^[12]提出的三参数法,将该文研究的总加权提前损失问题表示为: $1 | d_j^A = d^A | \sum w_j^A E_j^A : f_{\max}^B$ 。

2 问题分析

本节中,考虑代理 A 中的工件具有相同的工期,代理 B 只需要在排序中满足条件 $\max_j \{f_j^B(C_j^B)\} \leq U$ (U 为给定的整数),定义截止日期 D_j^B :若 $C_j^B \leq D_j^B$, 则 $f_j^B(C_j^B) \leq U$; 若 $C_j^B > D_j^B$, 则 $f_j^B(C_j^B) > U$ 。本文称工件 D_j^B 的最新开始时间(K)为它在可行排序中所获得的最大开始时间,并且对于所有 $J_j^B \in J^B$, 满足条件 $C_j^B \leq D_j^B$ 。

K 的值计算如下:代理 B 中的工件按 D_j^B 非减的排列,从最后 1 个工件 $J_{n_B}^B$ 开始排,它的开始时间为 $D_{n_B}^B - P_{n_B}^B$,继续向前排列,设 $K = \min\{D_j^B, K+1\} - p_j^B, j = n_B - 1, n_B - 2, \dots, 1$ 。容易得出:如果工件 J_j^B 在时刻 K 后开始加工,则代理 B 中至少有 1 个工件满足 $f_j^B(C_j^B) > U$ 。

现考虑代理 B 中的工件 J_j^B , 它的最新加工区间为 $[K, D_j^B]$ 。设 I 由 $\delta(\leq n_B)$ 个区间 $I_{1,h_1}, I_{h_1,h_2}, \dots, I_{h_{\delta-1},n_B}$ 构成,称为预留区间。每个预留区间 $I_{u,v}$ 的范围为 $[K_u, D_v^B]$, 且 $\|I_{u,v}\| = D_v^B - K_u = \sum_{j=u}^v p_j^B$, 换言之,工件 $J_u^B, \dots, J_v^B$ 与 $I_{u,v}$ 紧密相关。

引理 1 对于问题 $1 | d_j^A = d^A, \text{pmtn} | \sum w_j^A E_j^A : f_{\max}^B$ 存在一个最优解,其中代理 B 中工件为非中断的,这与它的预留区间相关。

证明 在问题 $1 | d_j^A = d^A, \text{pmtn} | \sum w_j^A E_j^A : f_{\max}^B$ 的最优解中,代理 B 中所有与 $I_{u,v}$ 相关的工件都要在 D_v^B 之前完成加工,当移动代理 B 中每个工件的所有部分到确定的区间 $I_{u,v}$ 时:情形 1, $d^A \in I_{u,v}$ 时,将工件 J_j^A 在 d^A 之前的加工部分 p_{j1}^A 向前移,在 d^A 之后的加工部分 p_{j2}^A 向后移;情形 2, $d^A < K_u$ 时,将工件 J_j^A 向后移;情形 3, $d^A > D_v^B$ 时,将工件 J_j^A 也向后移。故 3 种情形下代理 A 的总加权提前损失都没有增加。证毕

引理 2 给定 $1 \mid d_j^A = d^A, \text{pmtn} \mid \sum w_j^A E_j^A : f_{\max}^B$ 的一个最优解, 则存在 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : f_{\max}^B$ 的一个最优解, 其中它们的总加权提前损失工件是相等的。

证明 对于 $1 \mid d_j^A = d^A, \text{pmtn} \mid \sum w_j^A E_j^A : f_{\max}^B$ 的最优解问题, 若存在一个工件 J_j^A (属于任意代理), 至少可中断 1 次, 将工件的 p_{j1}^A 部分向后移靠近工期, 工件的 p_{j2}^A 部分向前移靠近工期, 这样总加权提前损失不变。对每个中断的工件重复以上操作, 则得到一个非中断的解。证毕

由于问题 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : C_{\max}^B \leq U$ 是 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : f_{\max}^B$ 的特殊情形, 因此讨论问题 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : f_{\max}^B$ 的时间复杂性, 可考虑问题 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : C_{\max}^B \leq U$ 的时间复杂性。

3 问题的算法

首先给出背包问题^[1], 即给定两个非负整数集 $\{u_1, u_2, \dots, u_n\}, \{w_1, w_2, \dots, w_n\}$ 和两个整数 b, W , 是否存在集合 $s \subset \{1, 2, \dots, n\}$, 使得 $\sum_{j \in s} u_j \leq b$ 且 $\sum_{j \in s} w_j \geq W$ 。

引理 3 问题 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : C_{\max}^B \leq U$ 是一般意义下 NP 难的。

证明 利用背包问题。

构造一个排序问题(用 EC_1 表示): 代理 A 有 $n_A = n$ 个工件, 代理 B 有 1 个工件, 需要在同一台机器上加工, 令: $\hat{u} = \sum_{j=1}^n u_j, \hat{w} = \sum_{j=1}^n w_j; p_j^A = u_j, d_j^A = b + 1 = d^A, w_j^A = w_j, j = 1, 2, \dots, n; p_1^B = \hat{u}\hat{w}, d_1^B = b; y = \hat{u}\hat{w} + (\hat{w} - W + w_1^B)p_1^B$; 考虑 $\sum w_j^A E_j^A \leq y$, 且 $C_{\max}^B \leq b + p_1^B = U$ 。

1) 背包 $\Rightarrow EC_1$ 。定义一个排序, 将工件 $\{J_j^A \mid j \in s\}$ 排在 J_1^B 的前面, 最后排工件 $\{J_j^A \mid j \notin s\}$ 。有 $\sum_{j \in s} p_j^A = \sum_{j \in s} u_j \leq b$, 则 $C_1^B = C_{\max}^B = \sum_{j \in s} p_j^A + p_1^B \leq b + p_1^B = U$, $\sum w_j^A E_j^A = \sum w_j^A \min\{\max\{d_j^A - S_j^A, 0\}, p_j^A\} \leq \sum_{j=1}^n w_j^A p_j^A + w_1^B p_1^B < \sum_{j=1}^n w_j^A C_j^A + w_1^B p_1^B = \sum_{j \in s} w_j^A C_j^A + \sum_{j \notin s} w_j^A C_j^A + w_1^B p_1^B = \sum_{j \in s} w_j^A C_j^A + \sum_{j \notin s} w_j^A (C_j^A - p_1^B + p_1^B) + w_1^B p_1^B = \sum_{j=1}^n w_j^A C_j^A + \sum_{j \notin s} w_j^A p_1^B + w_1^B p_1^B \leq \hat{u}\hat{w} + (\hat{w} - W + w_1^B)p_1^B = y$ 。

2) $EC_1 \Rightarrow$ 背包。 $\sum w_j^A E_j^A \leq y$, 且 $C_{\max}^B \leq b + p_1^B = U$ 成立, 将工件 $\{J_j^A \mid j \in s\}$ 排 J_1^B 的前面, 则 $y = \hat{u}\hat{w} + (\hat{w} - W + w_1^B)p_1^B = (1 + \hat{w} - W)p_1^B + w_1^B p_1^B > \sum_{j=1}^n w_j^A C_j^A + w_1^B p_1^B > \sum_{j \notin s} w_j^A C_j^A + w_1^B p_1^B$,

故有 $(1 + \hat{w} - W)p_1^B > \sum_{j \in s} w_j^A C_j^A = \sum_{j \in s} w_j C_j^A > \sum_{j \in s} w_j p_1^B$, 因此 $1 + \hat{w} - W > \sum_{j \in s} w_j, \hat{w} - W \geq \sum_{j \notin s} w_j$, 从而得到 $\sum_{j \in s} w_j \geq W$; 又有 $\sum_{j \in s} p_j^A = C_1^B - p_1^B \leq b$, 得到 $\sum_{j \in s} u_j \leq b$ 。证毕

对于特殊情形 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : C_{\max}^B \leq U$, 提出在 $O(n \sum p_j^A)$ 时间内更加有效的算法。假设代理 A 中的 n_A 个工件按照最小权重优先的规则排列, 即 $w_1^A \leq w_2^A \leq \dots \leq w_{n_A}^A$, 该方法的时间复杂度为 $O(n_A \log n_A)$, 支配于 $O(n \sum p_j^A)$ 。

在排序过程中, 只需要考虑所有工件从 0 时刻开始连续加工(没有空闲时间)。通过工件交换的原理, 可以证明在最优排序中代理 B 中的工件是连续加工的, 运行时间为 $O(n_B)$ 。因此, 可以将代理 B 中的所有工件看成一个加工时间为 $\sum p_j^B$ 的工件, 记作 $J_1^B, n_B = 1$, 且 $\sum p_j^A + p_1^B = T$ 。

引理 4 对于问题 $1 \mid d_j^A = d^A \mid \sum w_j^A E_j^A : C_{\max}^B \leq U$, 若存在最优排序 π 满足 $S_1^B(\pi) \geq d^A$, 那么 $\pi_0 = (J_1^A, J_2^A, \dots, J_{n_A}^A, J_1^B)$ 是一个最优排序。

证明 设 π 为最优排序, 但代理 A 中工件不按最小权重优先排, 在 π 中工件 J_j^A 排在 J_i^A 前面, 且工件 J_j^A 在 t 时刻开始加工, 其中 $d_j^A = d_i^A = d^A, w_j^A > w_i^A$ 。

情形 1, 工件 J_i^A 误工, 工件 J_j^A 部分提前。

1) 当 $E_j^A(\pi) \geq p_i^A$ 时,总加权提前损失为 $w_j^A E_j^A(\pi)$,现交换工件 J_j^A 与工件 J_i^A ,得到序列 π' ,总加权提前损失为 $w_i^A p_i^A + w_j^A (E_j^A(\pi) - p_i^A)$,由于 $w_j^A > w_i^A$,可知 π' 更优,这与 π 最优矛盾。

2) 当 $0 \leq E_j^A(\pi) < p_i^A$ 时,总加权提前损失为 $w_j^A E_j^A(\pi)$,现交换工件 J_j^A 与工件 J_i^A ,得到序列 π' ,总加权提前损失为 $w_i^A E_i^A(\pi')$,由于 $E_i^A(\pi') = E_j^A(\pi)$, $w_j^A > w_i^A$,可知 π' 更优,这与 π 最优矛盾。

情形 2,工件 J_i^A 部分提前,工件 J_j^A 提前。

1) 当 $d^A - t > p_i^A$ 时,总加权提前损失为 $w_j^A p_j^A + w_i^A (d^A - t - p_j^A)$;现交换工件 J_j^A 与工件 J_i^A ,得到序列 π' ,总加权提前损失为 $w_i^A p_i^A + w_j^A (d^A - t - p_i^A)$;由于 $w_j^A > w_i^A$,将前一情况下的总加权提前损失减去后一情况下的总加权提前损失,得 $(w_j^A - w_i^A)(p_j^A + p_i^A + t - d^A) > 0$,可知 π' 更优,这与 π 最优矛盾。

2) 当 $d^A - t \leq p_i^A$ 时,总加权提前损失为 $w_j^A p_j^A + w_i^A (d^A - t - p_j^A)$;现交换工件 J_j^A 与工件 J_i^A ,得到序列 π' ,总加权提前损失为 $w_i^A (d^A - t)$;由于 $w_j^A > w_i^A$,将前一情况下的总加权提前损失减去后一情况下的总加权提前损失,得 $(w_j^A - w_i^A)p_j^A > 0$,可知 π' 更优,这与 π 最优矛盾。证毕

接下来用 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 表示 $1 | d_j^A = d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 在约束条件为 $S_1^B(\pi) < d^A$ 的问题,其中 π 为该问题的可行排序。作为结论,在可行排序 π 中排在 J_1^B 之前的每个代理 A 的工件都是提前的。根据引理 4,找出问题 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 的最优排序 π ,再从 π 和 π_0 中选择最优的序列,进而解决问题 $1 | d_j^A = d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 。

对于问题 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$,通过建立动态规划算法来解决。其中: $P(j, t)$ 表示问题 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 在工件 $\{J_1^A, J_1^A, \dots, J_j^A\} \cup \{J_1^B\}$ 中讨论的子问题,并且最后 1 个工件在每个可行排序中的完工时间为 t ; $Z(j, t)$ 表示问题 $P(j, t)$ 的最优值;当情况不可行时,定义 $Z(j, t) = +\infty$ 。根据代理 A 中工件按最小权重优先排列,下面的引理很容易被证明。

引理 5 假设对于某些 $j \in \{1, 2, \dots, n_A\}$, $t \geq 0$ 的情形, $Z(j, t) < +\infty$, 那么问题 $P(j, t)$ 的最优排序满足工件 J_j^A 要么在 0 时刻开始加工,要么在时刻 t 完工。

根据引理 5 的描述,设问题 $P(j, t)$ 的一个最优排序为 π 。

工件 J_j^A 在 0 时刻开始加工,通过从序列中 π 删除工件 J_j^A 得到一个子问题 $P(j-1, t)$,最优值表示为: $Z(j-1, t) = Z(j, t) - w_j^A E_j^A(\pi) = Z(j, t) - w_j^A p_j^A$; 工件 J_j^A 在 t 时刻完工,通过从序列 π 中删除工件 J_j^A 得到一个子问题 $P(j-1, t-p_j^A)$,最优值表示为:

$$Z(j-1, t-p_j^A) = Z(j, t) - w_j^A E_j^A(\pi) = Z(j, t) - w_j^A \min\{\max\{d^A - (t-p_j^A), 0\}, p_j^A\}。$$

对于以上问题 $P(j, t)$ 的所有值 $Z(j, t)$ 进行分析,其中: $j \in \{1, 2, \dots, n_A\}$, 且 $t \geq 0$, 得出相应的动态规划算法 (DP1) 为: 1) 初始条件: 若 $d^A < p_1^B < U$, $Z(0, t) = 0$; 否则 $Z(0, t) = +\infty$ 。2) 递归公式: 对于每对 (j, t) , 其中: $j \in \{1, 2, \dots, n_A\}$, $t \geq 0$, 则有:

$$Z(j, t) = \min\{Z(j-1, t) + w_j^A p_j^A, Z(j-1, t-p_j^A) + w_j^A \min\{\max\{d^A - (t-p_j^A), 0\}, p_j^A\}\}。$$

问题 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 的最优值由 $Z(n_A, T)$ 得到,对应着最优排序 $P(n_A, T)$ 。

定理 1 问题 $1 | d_j^A = d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 的时间复杂度为 $O(n_B + n_A \sum p_j^A)$ 。

证明 因为在最优排序 π 中代理 B 中的工件是连续加工的,运行时间为 $O(n_B)$ 。因此,可以将代理 B 中的所有工件看成一个加工时间为 $\sum p_j^B$ 的工件,记作 J_1^B , $n_B = 1$ 。对于问题 $1 | d_j^A = d^A; S_1^B < d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$, 根据算法 DP1 只考虑状态 (j, t) , 其中: $j \in \{1, 2, \dots, n_A\}$, $t \in \{0, 1, \dots, \sum p_j^A\}$, 运行时间为 $O(n_A \sum p_j^A)$ 。又由引理 4 可知代理 A 中的 n_A 个工件按照最小权重优先的规则排列,即 $w_1^A \leq w_2^A \leq \dots \leq w_{n_A}^A$, 该方法的时间复杂度为 $O(n_A \log n_A)$, 支配于 $O(n_A \sum p_j^A)$ 。故时间复杂度为 $O(n_B + n_A \sum p_j^A)$ 。证毕

4 结论

本文主要研究问题 $1 | d_j^A = d^A | \sum w_j^A E_j^A; f_{\max}^B$ 的特殊情形 $1 | d_j^A = d^A | \sum w_j^A E_j^A; C_{\max}^B \leq U$ 的最小化问题,即在满足代理 B 的最大正则函数不大于给定的上界的条件下,代理 A 中工件的工期相同的情况下,它的总加权

提前损失最小。利用背包问题证明了该问题是一般意义下 NP 难的,并提出了一个拟多项式时间的算法,它的时间复杂度为 $O(n_B + n_A \sum p_j^A)$ 。后续研究可探讨在使代理 A 中工件的加工时间相同的情况下上述问题的最优排序,此外还可以考虑在平行机下的模型。

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 H. Competitive two-agent scheduling and its applications[J]. *Operations Research*, 2010, 58(2): 458-469.
- [4] PEREZ-GONZALEZ P, FRAMINAN J M. A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: multi-agent scheduling problems[J]. *European Journal of Operational Research*, 2014, 235(1): 1-16.
- [5] 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: 945-955.
- [6] ZHANG Y, YUAN J J. A note on a two-agent scheduling problem related to the total weighted late work[J]. *Journal of Combinatorial Optimization*, 2019, 37: 989-999.
- [7] 马露, 张新功. 关于总误工损失的两个代理单机排序问题[J]. *运筹学学报*, 2017, 21(1): 13-22.
MA L, ZHANG X G. Two-agent scheduling problem about total late work on a single machine[J]. *Operations Research Transactions*, 2017, 21(1): 13-22.
- [8] 张新功, 陈秋宏, 王祥兵. 关于误工的两个代理单机排序问题[J]. *重庆师范大学学报(自然科学版)*, 2018, 35(4): 1-6.
ZHANG X G, CHEN Q H, WANG X B. Two-agent scheduling problem about tardiness on a single machine[J]. *Journal of Chongqing Normal University (Natural Science)*, 2018, 35(4): 1-6.
- [9] BEN-YEHOSHUA Y, MOSHEIOV G. A single machine scheduling problem to minimize total early work[J]. *Computer and Operations Research*, 2016, 73: 115-118.
- [10] STERNA M, CZERNIACHOWSKA K. Polynomial time approximation scheme for two parallel machines scheduling with a common due date to maximize early work[J]. *Journal of Optimization Theory and Applications*, 2017, 174(3): 927-944.
- [11] CHEN X, LIANG Y G, STERNA M. Fully polynomial time approximation scheme to maximize early on parallel machines with common due date[J]. *European Journal of Operational Research*, 2020, 284(1): 67-74.
- [12] GRAHAM P L, LAWLER E L, LENSTRA J K. Optimization and approximation in deterministic machine scheduling: a survey [J]. *Annals of Discrete Mathematics*, 1979, 5(1): 287-326.

Operations Research and Cybernetics

Two-Agent Single Machine Scheduling Problem with the Total Weighted Early Work

ZHANG Xingong, LI Ping

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

Abstract: [Purposes] Two-agent single machine scheduling problem with the total weighted early work was studied. [Methods] The objective function is to minimize the total weighted early work under the same due date. The objective function of the second agent is the maximum regular function, in special cases, the maximum completion time. The goal is to find a scheduling so that the second agent objective is feasible, the objective function value of the first agent is minimized. [Findings] The proposed problem is NP-hard in the general sense. [Conclusions] An optimal algorithm for the scheduling problem with the total weighted early work is given. A quasi-polynomial time is presented to solve it.

Keywords: scheduling; two-agent; the total weighted early work

(责任编辑 方 兴)