运筹学与控制论

DOI:10.11721/cqnuj20230113

带有固定区间的双代理排序问题*

李　露，张新功
(重庆师范大学 数学科学学院，重庆 401331)

摘要:【目的】研究带有固定区间的双代理排序问题。【方法】第一个代理的工件加工过程可以中断，考虑两种机器类型：单台机器时考虑的目标函数为总权误工损失或总权提前损失；两台平行机时考虑的目标函数为总完工时间，同时必须在规定的固定区间加工第二个代理的工件，目标是在满足第二个代理目标的可行性前提下寻找一个使第一个代理的目标函数值更小的排序方案。【结果】设计了单台机器固定区间工件损失问题的排序算法，也为两台平行机总完工时间问题设计了相应算法。【结论】设计的算法可在多项式时间内得到解决，且证明了算法的最优性，并用数值实验说明了算法的可行性。

关键词:排序；双代理；固定区间；工件损失；总完工时间；时间复杂度

中图分类号:O223　　**文献标志码**:A　　**文章编号**:1672-6693(2023)01-0021-07

在现实生产环境中，工厂会接受不同的订单，待加工工件会来自不同的代理。面对如此复杂的加工条件，生产者想满足不同代理的需求，有必要合理地安排工件的加工顺序，让加工资源得到最优的利用。来源于工业生产加工的多代理排序问题最早由 Agnetis 等人[1]提出，他们研究了双代理排序的约束排序模型，以便找到一个排序能够最小化第一个代理的目标函数且第二个代理的目标函数值不超过给定值。Leung 等人[2]考虑了工件有到达时间且允许中断的双代理排序问题，将机器环境拓展到平行机上进行研究，考虑的目标函数有最大费用函数、总误工、总权完工时间等。近些年来，双代理排序的研究内容向各个领域扩充，学者们在双代理的基础上增加约束条件进行了深入研究。

固定区间问题可被理解为带有可用性约束的排序问题。可用性约束也被称为定期维护，这类问题最先由 Leung 等人[2]提出。陈秋宏等人[3]研究了与总误工相关的固定区间模型，提出了一个伪多项式时间可解的动态规划算法。误工损失问题最早由 Blazewicz[4]提出，并认为这是一种"信息流失"。之后误工损失一词的准确定义则由 Potts 等人[5]确定。戴秦等人[6]在 Potts 等人[5]的研究基础上研究误工损失的两个竞争代理问题。Zhang 等人[7]利用二分法给出了双代理误工损失问题的 NP 难证明，并提出了总权误工损失问题的伪多项式时间算法。提前损失问题首先由 Ben-Yehoshua 等人[8]提出，惩罚工件工期前完成加工的部分，且利用二划分问题给出了单机总提前损失问题的 NP 难证明，并提出了伪多项式时间动态规划算法。Sterna 等人[9]研究了在两台平行机上工件具有共同工期的总提前损失问题，并讨论了误工损失与提前损失的关系。张新功等人[10]研究了单台机器上与总加权提前损失有关的双代理排序问题，利用背包问题证明了该问题是一般意义下 NP 难的。Sterna[11]对近 20 年来误工损失及提前损失问题进行了完整的综述，包括复杂度分析以及针对单机、并行和专用机器的各种模型的精确、启发式和近似算法的建议，还扩展了其他参数和约束，如学习效果、机器不可用、机器合格性和设置时间。

本文在上述文献的基础上研究了带有固定区间的双代理排序问题，即：第一个代理的工件加工过程可中断，第二个代理的工件在固定时间窗口内加工；单台机器时，第一个代理的目标函数为总权误工损失与总权提前损失；两台平行机时，第一个代理的目标函数为总完工时间，在保证第二个代理的工件加工可行性的前提下，使得第一个代理的目标函数值更小。

---

* **收稿日期**:2022-03-25　　**修回日期**:2022-08-30　　**网络出版时间**:2023-02-23　09:03

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)；重庆市教育委员会科学技术研究计划重点项目(No. KJZD-K202000501)；重庆市科学技术局研究项目(No. cstc2021jcyj-msxmX0229)；重庆市教育委员会研究生教育教学改革研究重点项目(No. YJG182019)

第一作者简介:李露，女，研究方向为组合最优化，E-mail:2657993952@qq.com；**通信作者**:张新功，男，教授，博士，E-mail:zxg7980@163.com

网络出版地址:https://kns.cnki.net/kcms/detail/50.1165.N.20230222.1158.006.html

1 问题描述与一般性质

m 台机器上的两个竞争代理 A 和 B 的排序问题表述如下：设 $M_i(i=1,2,\cdots,m)$ 表示第 i 台机器，则 $\{M_1,M_2,\cdots,M_i\}$ 表示机器集合。在上下文明确的时候，直接称机器 i 指代 M_i；两个竞争代理 A 和 B 的工件集 J^{A} 和 J^{B} 在机器 i 上进行加工，代理 A 有 n_{A} 个工件，代理 B 有 n_{B} 个工件，用 $J^{\mathrm{A}}=\{J_1^{\mathrm{A}},J_2^{\mathrm{A}},\cdots,J_{n_{\mathrm{A}}}^{\mathrm{A}}\}$，$J^{\mathrm{B}}=\{J_1^{\mathrm{B}},J_2^{\mathrm{B}},\cdots,J_{n_{\mathrm{B}}}^{\mathrm{B}}\}$ 表示工件集合。令：$X=\{\mathrm{A},\mathrm{B}\}$，$n$ 表示所有的工件个数，有 $n=n_{\mathrm{A}}+n_{\mathrm{B}}$。在指定加工顺序 σ 下，每个工件有一个开始加工时间 $S_j^X(\sigma)$，对应一个完工时间 $C_j^X(\sigma)$，工件 J_j^X 的加工时间为 p_j^X、到达时间为 r_j^X、工期为 d_j^X、权重为 w_j^X。不失一般性，p_j^X 为非负整数。

目前国际上通用的三参数表示法为 $\alpha|\beta|\gamma$[12]，本文用到的目标函数有总权误工损失 $\sum w_jV_j$ 、总权提前损失 $\sum w_jE_j$ 、完工时间和 $\sum C_j$ 以及可行性 fsbl，其中：

$$V_j^X(\sigma)=\begin{cases}0,\text{若 } C_j^X(\sigma)\leqslant d_j^X\\ C_j^X(\sigma)-d_j^X,\text{若 } d_j^X<C_j^X(\sigma)<d_j^X+p_j^X,\\ p_j^X,\text{若 } d_j^X+p_j^X\leqslant C_j^X(\sigma)\end{cases}$$

$$E_j^X(\sigma)=\begin{cases}p_j^X,\text{若 } C_j^X(\sigma)\leqslant d_j^X\\ d_j^X-S_j^X(\sigma),\text{若 } S_j^X(\sigma)\leqslant d_j^X\leqslant C_j^X(\sigma)。\\ 0,\text{若 } d_j^X\leqslant S_j^X(\sigma)\end{cases}$$

在双代理排序中三参数表示法为 $\alpha|\beta^{\mathrm{A}};\beta^{\mathrm{B}}|\gamma^{\mathrm{A}}:\gamma^{\mathrm{B}}$，其中：$\beta^{\mathrm{A}}$、$\beta^{\mathrm{B}}$ 分别表示代理 A、B 的工件特征参数，γ^{A}、γ^{B} 分别表示代理 A、B 的目标函数。目标是在可行排序中找到在代理 B 工件的目标函数值不超过给定值的前提下，代理 A 的目标函数值最小的排序方案。本文研究的问题用三参数表示法表示为：

$$1|\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}|\sum w_i^{\mathrm{A}}V_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}},$$

$$1|\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}|\sum w_i^{\mathrm{A}}E_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}},$$

$$P2|\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}|\sum C_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}}。$$

在实际生产加工中，以上 3 个目标函数分别表示在保证代理 B 的要求得到满足的情况下，使得代理 A 的总权误工损失、总权提前损失、完工时间和更小。

定义 1[13] EDD 规则：按照工期 d_j 非减的顺序来排列工件 $J_1,J_2,\cdots,J_n$ 的排序规则。

定义 2[13] MW 规则：按照工期 w_j 非增的顺序来排列工件 $J_1,J_2,\cdots,J_n$ 的排序规则。

定义 3[13] SPT 规则：按照加工时间 p_j 非减的顺序来排列工件 $J_1,J_2,\cdots,J_n$ 的排序规则。

2 总权误工损失

Hariri 等人[14]研究了问题 $1\|\sum w_jV_j$，并且对问题 $1|\mathrm{pmtn}|\sum w_jV_j$ 给出了时间复杂度为 $O(n\log n)$ 的多项式时间算法。本节对固定区间的总权误工损失问题 $1|\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}|\sum w_i^{\mathrm{A}}V_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}}$ 提出了多项式时间算法。因为代理 B 工件必须在固定区间内加工，因此代理 A 的工件须中断，则将所有工件按 EDD 规则重新编号。令 t 表示当前最大工期的值，接着在工期大于等于 t 的工件集中，找出权重最大的工件，让它在 t 时刻完成加工；从后往前排列，如果遇到固定区间，则中断加工，在固定区间内加工代理 B 的工件；更新 t，直到所有工件加工完成。

算法 1 步骤 1，按 EDD 规则将代理 A 工件重新编号，有 $S=\{J_1^{\mathrm{A}},J_2^{\mathrm{A}},\cdots,J_{n_{\mathrm{A}}}^{\mathrm{A}}\}$，令 $t=d_{n_{\mathrm{A}}}^{\mathrm{A}}$。

步骤 2，将代理 B 中的工件 J_k^{B} 排在$[r_k^{\mathrm{B}},d_k^{\mathrm{B}}]$进行加工。

步骤 3，$A_j=\{J_j^{\mathrm{A}}|J_j^{\mathrm{A}}\in S;d_j^{\mathrm{A}}\geqslant t\}$，同时选择其中权重最大的一个工件加工，不妨设为 J_i^{A}，完工时间为 t。

步骤 3.1，如果代理 A 中有工件 J_i^{A} 满足 $t-p_i^{\mathrm{A}}<d_l^{\mathrm{A}}<t$。令 $s=d_l^{\mathrm{A}}$，否则 $s=\max\{t-p_i^{\mathrm{A}},0\}$。之后令 $t=s$。

步骤 3.2，如果代理 B 中有工件 J_k^B 满足 $r_k^B < t - p_i^A < d_k^B$。令 $s = d_k^B$，否则 $s = \max\{t - p_i^A, 0\}$。之后令 $t = s - p_k^B$。

步骤 4，对工件 J_i^A 在 $[s, t]$ 内令加工时间为 $t - s$，令 $p_i^A = p_i^A - (t - s)$，如果 $p_i^A = 0$，则 $S = S - \{J_i^A\}$。如果 $S = \varnothing$ 或 $t = 0$，那么便可计算出它的总权误工损失为 $\sum w_i^A V_i^A$，并终止。否则，对所有的工件 $J_i^A \in S$，有 $d_i^A < t$，令 $t = \max\{d_i^A\}$。并转到步骤 3。

定理 1　算法 1 是问题 $1 | \mathrm{pmtn}^A; r_j^B, d_j^B, p_j^B = d_j^B - r_j^B | \sum w_i^A V_i^A : \mathrm{fsbl}^B$ 的最优算法。

证明　考虑最优排序 σ，将它与算法 1 产生的排序 σ' 作比较。

代理 B 的工件只能安排在固定区间，因此排序 σ 与 σ' 中代理 B 工件的排序是相同的，下面考虑代理 A 工件的排法。

假设在 σ 中，在 $d_{n_A}^A$ 之前加工的工件是提前完工或部分提前完工工件，否则该工件将在不影响总权误工损失的情况下，在 $d_{n_A}^A$ 之后重新排序。下面需证在不增加总权误工损失的情况下，σ 经过有限次转换可得到 σ'。

假设 σ 与 σ' 在区间 $[0, d_{n_A}^A]$ 中工件排序不同，选择具有尽可能小的工期 h $(0 < h < d_{n_A}^A)$ 的工件，满足在区间 $[h, d_{n_A}^A]$ 的排序相同，在 σ' 中，假设工件 J_i^A 被安排在区间 $[l, h]$ $(0 < l < h)$ 而不是安排在 l 之前，因为在 σ' 中，机器在 h 前不能有空闲，如果存在，那该算法的机制保证了没有提前工件或部分提前工件可用，因此在 σ 中，机器在 h 前也有空闲，与 h 的定义相矛盾。

考虑在 σ 中，假设在区间 $[m, h]$ $(0 < m < h)$ 是空闲的但在 m 之前不是空闲的。令 $q = \min\{l, m\}$，改变 σ，通过将工件 J_i^A 的 $h - q$ 加工部分移动到区间 $[q, h]$，其中该部分原先被安排在 q 之前或 $d_{n_A}^A$ 之后，由于移动到区间 $[q, h]$ 部分的完工时间为 d_i^A（否则该算法不会将工件 J_i^A 安排在区间 $[l, h]$），因此该算法不会增加总权误工损失。

考虑排序 σ 中，工件 J_i^A 与工件 J_j^A 之间的转换，工件 J_i^A 被安排在区间 $[m, h]$ 中，而不是安排在 m 之前，现在将工件 J_i^A 在区间 $[q, h]$ 加工 $h - q$ 部分，与工件 J_i^A 原先被安排在 q 之前或 $d_{n_A}^A$ 之后的 $h - q$ 部分进行交换，在该算法下，工件 J_i^A 与工件 J_j^A 都是应该被安排在 h 之前的候选工件，但因为 $w_i \leqslant w_j$，故选择工件 J_i^A。其中工件 J_i^A 与工件 J_j^A 相应的误工损失部分交换，只会让总权误工损失减少。因此这样的交换不会增加工件的总权误工损失。

通过上述变换，得到最优排序，并在区间 $[q, d_{n_A}^A]$ 的排序与 σ' 相同，由于 σ 与 σ' 都有有限次中断，故 σ 经有限次转换得到最优排序 σ^*。　证毕

定理 2　问题 $1 | \mathrm{pmtn}^A; r_j^B, d_j^B, p_j^B = d_j^B - r_j^B | \sum w_i^A V_i^A : \mathrm{fsbl}^B$ 能在 $O(n_A \log n_A + n_B)$ 内得到最优排序。

证明　由于问题 $1 | \mathrm{pmtn} | \sum w_i V_i$ 能在多项式时间 $O(n_A \log n_A)$ 内得到最优排序[14]。此外安排代理 B 工件需要 $O(n_B)$。因此该问题最优排序能在 $O(n_A \log n_A + n_B)$ 内得到。　证毕

下面是算法 1 的应用。

算例 1　根据算法 1，对问题 $1 | \mathrm{pmtn}^A; r_j^B, d_j^B, p_j^B = d_j^B - r_j^B | \sum w_i^A V_i^A : \mathrm{fsbl}^B$ 考虑下列工件（表 1），给出最小化代理 A 的总权误工损失的最优排序。

表 1　算例 1 工件集基本信息

Tab. 1　The jobs information of example 1

工件信息	J_i^X				
	J_1^A	J_2^A	J_3^A	J_1^B	J_2^B
r_i^X	0	0	0	3	7
p_i^X	2	4	3	1	2
d_i^X	5	5	10	4	9
w_i^X	3	2	1		

根据算法 1,先将代理 B 工件安排在固定区间进行加工,即 J_1^B 安排在区间[3,4],J_2^B 安排在区间[7,9],然后再将代理 A 工件中工期最大的一个工件自该工期从后往前排,首先 J_3^A 的工期最大,于是 J_3^A 被安排在$[5,7]\cup[9,10]$,此时决策点 t 更新为 5,未排工件为 J_1^A,J_2^A,由于 $w_1^A>w_2^A$,因此优先排 J_1^A,J_1^A 被安排在$[2,3]\cup[4,5]$,剩余[0,2]加工 J_2^A,得出最优加工顺序为 $J_2^A\to J_1^A\to J_1^B\to J_1^A\to J_3^A\to J_2^B\to J_3^A\to J_2^A$,可见代理 A 工件的总权误工损失主要是 J_2^A 的总权误工损失,最终计算可得总权误工损失的最优值为 4。

3 总权提前损失

本节考虑的是固定区间的总权提前损失问题 $1|\mathrm{pmtn}^A;r_j^B,d_j^B,p_j^B=d_j^B-r_j^B|\sum w_i^A E_i^A:\mathrm{fsbl}^B$,由于代理 B 的工件在固定区间内加工,因此代理 A 的工件需中断才能使目标函数达到最优,运用前文所述的总权误工损失类似的思路,先将代理 B 的工件安排在固定区间内加工,再根据工件的工期大小,选择权重最大的工件从该工期开始加工。接下来给出该问题的多项式时间算法。

首先,令 S 表示未加工完成的工件集合,p_j 表示工件 J_j 未加工部分的剩余加工时间,a 表示决策点。

算法 2 步骤 1,按 EDD 规则把代理 A 中工件重新进行编号,$S=\{J_1^A,J_2^A,\cdots,J_{n_A}^A\}$,令 $u=d_1^A$,$\forall i=1,2,\cdots,n_A$,$j=1,2,\cdots,n$ 有 $d_i^A<\sum_n p_j$ 成立。

步骤 2,将代理 B 中的工件 J_k^B 排在$[r_k^B,d_k^B]$进行加工。

步骤 3,找出满足集合 $A_j=\{J_j^A|J_j^A\in S;d_j^A\leqslant u\}$,同时选择权重最大的一个工件加工,不妨设为 J_j^A。

步骤 3.1,如果代理 A 中有工件 J_l^A 满足 $u<d_l^A<u+p_i^A$。令 $v=d_l^A$,否则 $v=\min\left\{u+p_i^A,\sum_n p_j\right\}$。之后令 $u=v$。

步骤 3.2,如果代理 B 中有工件 J_k^B 满足 $r_k^B<a+p_i^A<d_k^B$。令 $v=r_k^B$,否则 $v=\max\left\{u+p_i^A,\sum_n p_j\right\}$。之后令 $u=v+p_k^B$。

步骤 4,对工件 J_i^A 在$[u,v]$内令加工时间为 $v-u$,令 $p_i^A=p_i^A-(v-u)$,如果 $p_i^A=0$,则 $S=S-\{J_i^A\}$。如果 $S=\varnothing$或 $u=\sum_n p_j$,那么便可计算出它的总权提前损失为 $\sum w_i^A E_i^A$,并终止。否则,若对所有的工件 $J_i^A\in S$,便有 $d_i^A\geqslant u$,令 $u=\min\{d_i^A\}$。并且转到步骤 3。

定理 3 算法 2 是问题 $1|\mathrm{pmtn}^A;r_j^B,d_j^B,p_j^B=d_j^B-r_j^B|\sum w_i^A E_i^A:\mathrm{fsbl}^B$ 的最优算法。

定理 3 的证明同定理 1 的证明类似。

定理 4 问题 $1|\mathrm{pmtn}^A;r_j^B,d_j^B,p_j^B=d_j^B-r_j^B|\sum w_i^A E_i^A:\mathrm{fsbl}^B$ 能在 $O(n_A\log n_A+n_B)$内得到最优排序。

证明 由于问题 $1|\mathrm{pmtn}|\sum w_i^A E_i^A$ 的最优排序能在多项式时间 $O(n_A\log n_A)$内得到。代理 B 的工件排序需要 $O(n_B)$时间。故此问题能在 $O(n_A\log n_A+n_B)$时间内得到最优排序。 证毕

4 平行机总完工时间问题

本节研究的问题是 $P2|\mathrm{pmtn}^A;r_j^B,d_j^B,p_j^B=d_j^B-r_j^B|\sum C_i^A:\mathrm{fsbl}^B$,其中的固定时间窗口是不重叠的情况,且这种固定时间窗口是只与时刻点有关而与机器无关的,即在固定时间窗口可以在任意一台机器上加工。设有两台平行机,即机器 1 和机器 2。

算法 3[15] 1) 将全部工件按 SPT 规则重新编号;2) 取前 m 个工件任意分给 m 个机器;3) 重复 2),直到全部工件排完。

引理 1[15] 对于平行机总完工时间问题 $Pm\|\sum C_j$,采用算法 3 可以得到最优排序。

引理 2 代理 B 的工件的加工有以下特点:1) 代理 B 的工件在机器 1 与机器 2 上加工是等价的;2) 代理 B 的工件可以在同一台机器上连续加工。

证明　1）由于机器 1 与机器 2 是两台平行机，工件在任何一台机器上都可以进行加工，因此代理 B 的工件在机器 1 与机器 2 上加工是等价的。

2）不妨假设代理 B 的工件 J_j^{B} 需在固定区间$[a,c]$进行加工，若工件 J_j^{B} 在机器 1 的区间$[a,b]$上加工，在机器 2 的区间$[b,c]$上加工，另外对应的机器 2 的区间$[a,b]$与机器 1 的区间$[b,c]$有代理 A 的工件加工，可以交换机器 2 的区间$[b,c]$上的工件与机器 1 的区间$[b,c]$上的工件，得到代理 B 的工件 J_j^{B} 在机器 1 的区间$[a,c]$上连续加工。同理可以交换机器 2 的区间$[a,b]$上的工件与机器 1 的区间$[a,b]$上的工件，得到代理 B 的工件 J_j^{B} 在机器 2 的区间$[a,c]$上连续加工。因此代理 B 的工件可在同一台机器上连续加工。　证毕

根据上述引理，设计问题 $P2\,|\,\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}\,\Big|\,\sum C_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}}$ 最优算法如下。

算法 4　步骤 1，将代理 B 的工件安排在机器 1 的固定区间进行加工。

步骤 2，对代理 A 工件按 SPT 规则进行重新编号。当机器一有空闲，便挑选出加工时间最短的代理 A 工件安排到空闲机器上进行加工。若两台机器都有空闲，不妨将加工时间最短的工件安排到机器 1 上，第二短的安排到机器 2 上。

步骤 3，机器 1 上的代理 A 工件 J_i^{A} 正在加工遇代理 B 工件 J_k^{B} 的固定区间，则中断机器 1 上代理 A 工件 J_i^{A} 以及机器 2 上代理 A 工件 J_{i+1}^{A} 的加工，并将未完成加工的部分看成新工件，此时空闲的机器为机器 2，转至步骤 2：挑选代理 A 加工时间最短工件在机器 2 上加工。

定理 5　算法 4 对问题 $P2\,|\,\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}\,|\,\sum C_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}}$ 产生最优排序。

证明　代理 B 的工件只能安排在固定区间内进行加工，无需讨论。接下来讨论代理 A 工件的最优排法：在有 2 台处理机的情况下，有 $2n_{\mathrm{A}}$ 个位置可供工件选择，因此有 $2n_{\mathrm{A}}$ 个系数供加工时间选择，这些系数是 2 个 n_{A}，2 个$(n_{\mathrm{A}}-1)$，…，2 个 1，加工时间分配给上述系数的一个子集，使乘积之和为最小。

为叙述方便，设$\dfrac{n}{m}$是整数，否则可增加一个加工时间为 0 的工件，使$\dfrac{n}{m}$为整数，增加加工时间为 0 的工件并不使问题有任何改变。易见为使乘积之和为最小，2 个最大的加工时间应分配给 2 个 1，2 个第二大的加工时间应分配给 2 个 2，等等。这使得 2 个最大的工件在不同的处理机上加工，2 个第二大的工件在不同的处理机上加工，等等。

上述分配与 SPT 规则产生的排序是一致的。根据 SPT 规则，最小的工件在机器 1 上加工，第二小的工件在机器 2 上加工，第三小的工件在机器 1 上加工，以此类推。

由于两者分配一致，SPT 规则产生最优排序。故算法 4 是该问题的最优算法。　证毕

定理 6　算法 4 的复杂度为 $O(n_{\mathrm{A}}\log n_{\mathrm{A}}+n_{\mathrm{B}})$。

证明　代理 A 中的工件通过 SPT 规则排序需 $O(n_{\mathrm{A}}\log n_{\mathrm{A}})$，代理 B 工件的安排需花费 $O(n_{\mathrm{B}})$时间，工件中断之后再进行重新排序的过程中最多需 $O(n_{\mathrm{A}}\log n_{\mathrm{A}})$，因此算法的复杂度为 $O(n_{\mathrm{A}}\log n_{\mathrm{A}}+n_{\mathrm{B}})$。　证毕

接下来，将进行一次数据实验，对算法 4 进行进一步的说明。

算例 2　对于问题 $P2\,|\,\mathrm{pmtn}^{\mathrm{A}};r_j^{\mathrm{B}},d_j^{\mathrm{B}},p_j^{\mathrm{B}}=d_j^{\mathrm{B}}-r_j^{\mathrm{B}}\,\Big|\,\sum C_i^{\mathrm{A}}:\mathrm{fsbl}^{\mathrm{B}}$，假设现有两个代理 A 和 B，已知工件 $J^{\mathrm{A}}=\{J_1^{\mathrm{A}},J_2^{\mathrm{A}},J_3^{\mathrm{A}},J_4^{\mathrm{A}},J_5^{\mathrm{A}}\}$，$J^{\mathrm{B}}=\{J_1^{\mathrm{B}},J_2^{\mathrm{B}},J_3^{\mathrm{B}}\}$，代理 A 和 B 中的工件在机器上加工的信息表如下列表 2 所示，求该问题的最优值。

表 2　算例 2 工件集基本信息

Tab. 2　The jobs information of example 2

工件信息	J_i^X							
	J_1^{A}	J_2^{A}	J_3^{A}	J_4^{A}	J_5^{A}	J_1^{B}	J_2^{B}	J_3^{B}
r_i^X	0	0	0	0	0	2	7	9
p_i^X	2	3	3	4	6	2	2	3
d_i^X						4	9	12

根据算法4,将代理B的工件安排在机器1的固定区间进行加工,即J_1^B,J_2^B,J_3^B的加工区间分别为[2,4]∪[7,9]∪[9,12],代理A工件按SPT规则在空闲的机器上加工,首先在0时刻,两台机器都空闲,按照算法,J_1^A在机器1的区间[0,2]上加工,J_2^A在机器2上加工,在时刻2遇J_1^B的固定区间中断,中断后新工件$J_{2'}^A$为加工时间最短的工件,被安排在空闲的机器上加工,即机器2。工件$J_{2'}^A$加工完成时刻为3,此时空闲机器为机器2,选择未排工件中加工时间最短的工件,即工件J_3^A。加工至时刻点4,J_1^B加工完成,机器1空闲,挑选出加工时间最短的工件在机器1上加工,以此类推,重复上述步骤,上述算例的最优排序为:机器1的加工顺序为$J_1^A \to J_1^B \to J_4^A \to J_2^B \to J_3^B$,机器2的加工顺序为$J_2^A \to J_3^A \to J_5^A \to J_4^A \to J_5^A$,代理A的总完工时间为32。

5 结束语

本文考虑了带有固定区间的双代理排序问题即单台机器固定区间工件损失问题与两台平行机固定区间总完工时间问题,其中:第二个代理(代理B)的工件在固定区间内加工,并且工件的加工时间p_j^B正好等于相应的固定区间长度,平行机的固定区间与时间段有关而与机器无关。本文给出了上述问题的算法和时间复杂度分析。在后续研究中,可考虑其他目标函数,也可以考虑更为复杂的机器环境,如多台平行机或流水车间等。

参考文献:

[1] AGNETIS A, MIRCHANDANI P B, PACCIARELLI D. Scheduling problems with two competing agents[J]. Operations Research,2004,52(2):229-242.

[2] LEUNG J Y T,PINEDO M,WAN G H. Competitive two-agent scheduling and its applications[J]. Operations Research,2010,58(2):458-469.

[3] 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 Transactions,2019,23(1):61-71.

[4] BLAZEWICZ J. Scheduling preemptive tasks on parallel processors with information loss[J]. Technique et Science Informatiques,1984,3(6):415-420.

[5] POTTS C N,Van WASSENHOVE L N. Single machine scheduling to minimize total late work[J]. Operations Research,1992,40(3):586-595.

[6] 戴秦,郑兴山,张新功,等.总迟后相关的两个工况代理的单机排序问题[J].工业工程与管理,2014,19(6):78-82.
DAI Q,ZHENG X S,ZHANG X G,et al. Two-agent scheduling problem of working condition with total late work on a single machine[J]. Industrial Engineering and Management,2014,19(6):78-82.

[7] 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(3):989-999.

[8] BEN-YEHOSHUA Y,MOSHEIOV G. A single machine scheduling problem to minimize total early work[J]. Computer and Operations Research,2016,73:115-118.

[9] STERNA M,CZERNIACHOWSKA K. Polynomial time approximation scheme for two parallel machines scheduling with acommon due date to maximize early work[J]. Journal of Optimization Theory and Applications,2017,174(3):927-944.

[10] 张新功,栗苹.关于总加权提前损失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),2021,38(1):64-68.
ZHANG X G,LI P. Two-agent single machine scheduling problem with the total weighted early work[J]. Journal of Chongqing Normal University (Natural Science),2021,38(1):64-68.

[11] STERNA M. Late and early work scheduling:a survey[J]. Omega,2021,104(15/16):102453.

[12] GRAHAM R L,LAWLER E L,LENSTRA J K,et al. Optimization and approximation in determ-inistic machine scheduling:a survey[J]. Annals of Discrete Mathematics,1979,5:287-326.

[13] PETER B. Scheduling algorithms[M]. New York:Springer-Verlag,2007.

[14] HARIRIA M A,POTTS C N,Van WASSENHOVE L N. Single machine scheduling to minimize total weighted late work[J]. ORSA Journal on Computing,1995,7(2):232-242.

[15] 万国华.排序与调度的理论、模型和算法[M].北京:清华大学出版社,2019.
WAN G H. Theory,model and algorithm of scheduling[M]. Beijing:Tsinghua University Press,2019.

Operations Research and Cybernetics

Two-Agent Preemptive Scheduling of Jobs with Fixed Time Windows Problem

LI Lu, ZHANG Xingong

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

Abstract: [**Purposes**]Two-agent scheduling problem with fixed time windows was studied. [**Methods**]The jobs of the first agent might be preemptive. The objective function is the total weighted late work or the total weighted early work on a single machine, and the objective function is the total completion time on two parallel machines. The second agent is arranged to be processed in a fixed time window. The objective is to find a scheduling scheme to make the objective function value of the first agent as small as possible when the second agent objective is feasible. [**Findings**]The scheduling algorithm of fixed time windows job loss problem on single machine was designed, and the corresponding algorithm was also designed for the total completion time of two parallel machines. [**Conclusions**]It is proved that the designed algorithms are polynomial time solvable, and the optimality of the algorithms is proved.

Keywords: scheduling; two agent; fixed time windows; work loss; total completion time; time complexity

（责任编辑　方　兴）