

带有拒绝工件和学习效应的资源约束排序问题研究^{*}

国峰, 王吉波

(沈阳航空航天大学 理学院, 沈阳 110136)

摘要:【目的】研究工件加工时间具有学习效应以及工件可拒绝的单机排序问题。在线性和凸资源分配函数的两种模型下, 为求得可接受加工的工件集合、可拒绝工件的集合以及确定可接受工件集合中的最优工件排序, 使工件的时间表长、总完工时间、资源耗费费用和工件拒绝费用的加权和最小。【方法】对于线性资源分配函数问题, 在拒绝工件数给定的情况下, 此问题可转化为指派问题。对于凸资源分配函数问题, 此问题可以用动态规划算法进行求解。【结果】对于带有拒绝工件和学习效应的资源约束排序问题给出了最优求解算法, 且它们的时间复杂度分别为 $O(n^4)$ 和 $O(n^3)$, 其中 n 为工件的个数。【结论】算法分析和 CPU 运算时间表明给出的求解算法非常有效。

关键词: 排序; 拒绝工件; 资源分配; 学习效应

中图分类号: O223; C934

文献标志码: A

文章编号: 1672-6693(2021)01-0114-07

在经典的排序模型中, 所有的工件是必须被加工的; 但在实际的生产中, 由于厂商加工能力的限制以及为寻求更高的价值利润, 厂商需要拒绝加工时间较长或利润较少的工件; 与此同时, 厂商也需要赔付一定的拒绝费用。此为可拒绝工件的排序问题。Barta 等人^[1]最先考虑可拒绝工件的排序模型, 对于平行机极小化可接受加工工件的时间表长和拒绝费用之和的问题提出了近似算法。Engels 等人^[2]提出了多种算法解决加权总完工时间和拒绝费用之和的极小化问题。Zhang 等人^[3]证明了在有准备时间的情况下, 时间表长和拒绝费用之和的问题是 NP-难的。Sengupta^[4]基于拒绝工件的前提对极小化工件最大延误和拒绝费用的问题给出了动态规划算法以及多项式时间近似算法。Li 等人^[5]利用动态规划算法解决了带拒绝和恶化工件的平行机排序问题。Cheng 等人^[6]证明了带有线性恶化和拒绝工件的单机排序问题中的极小化加权总完工时间、时间表长、最大延误与拒绝惩罚之和的问题都是 NP-难的。慕迪等人^[7]研究了目标函数是最小化接受工件的最大完工时间与所有被拒绝工件的拒绝费用之和极小化单机问题, 对此问题, 他们给出了分支定界算法。刘晓霞等人^[8]考虑了工件有到达时间且拒绝工件总个数不超过某个给定值的单机平行分批排序问题, 目标为如何安排被接受工件的加工批次和加工次序使得该问题最大完工时间与被拒绝工件的总拒绝费用之和最小, 并对此问题给出了伪多项式时间动态规划精确算法、2-近似算法和完全多项式时间近似方案。Shabtay 等人^[9]对于离线状态下的带可拒绝工件的排序问题进行了综述。

另一方面, 经典排序中通常假定工件的加工时间为给定常数^[10-12], 但在实际生产过程中, 工件的加工时间可能与它所分配的资源量有关。1980 年 Vickson^[13]最早提出加工时间具有资源可控的排序问题, Shabtay 等人^[14]对具有资源约束的排序问题进行总结, 给出了资源分配的两种主要模型: 线性资源分配和凸资源分配。在此后, Leyvand 等人^[15]提出并证明了带有凸资源分配排序问题的统一方法。与此同时, 工人具有一定的学习能力, 因此工件的加工时间可能与它所排位置有关, 此称为学习效应^[16-18]。Azzouz 等人^[19]对于工件具有学习效应的排序问题进行了综述。Wang 等人^[20]研究了工件同时具有学习效应和资源分配的单机排序问题, 当工件的加工时间分别是资源的线性函数和凸函数时, 他们对于两类正则目标函数(包括最大完工时间、总完工(等待)时间、完工(等待)时间的总绝对差)分别提出了多项式时间算法。Wang 等人^[21]研究了带有学习效应和资源分配的单机排序问题, 在凸资源分配模型下, 他们提出了两种优化模型能够在多项式时间内解决。与此同时, 这些求解模型

* 收稿日期: 2020-02-25 修回日期: 2020-10-08 网络出版时间: 2021-03-10 17:19

资助项目: 国家自然科学基金(No. 71471120); 辽宁省自然科学基金(No. 2020-MS-233)

第一作者简介: 国峰, 男, 研究方向为运筹学与组合优化, E-mail: 1666454012@qq.com; 通信作者: 王吉波, 男, 教授, 博士生导师, E-mail: wangjibo75@163.com

网络出版地址: <https://kns.cnki.net/kcms/detail/50.1165.n.20210310.1126.004.html>

的算法也可应用于老化效应的情况。对于工件具有学习效应和凸资源分配的双机流水作业排序问题, Liu 等人^[22]在工件具有不等待约束下,对两类正则目标函数分别给出了最优算法,并证明了这些算法是多项式时间的; Gao 等人^[23]则在工件具有不等待约束和共同工期的约束下,证明了当目标函数是提前或延误时,共同工期费用和资源消耗费用和的问题是多项式时间算法的。姜昆^[24]研究了带凸资源和恶化效应的单机松弛窗口排序问题,当总资源消耗费用有上界,且目标函数为极小化与窗口有关的排序问题是具有多项式时间算法的。杨盛武等人^[25]研究了在资源消耗费用有限情况下极小化最大完工时间问题,在最大完工时间有限制的情况下极小化资源消耗费用问题。他们证明了这两个问题均是多项式时间可解的。

本文考虑工件同时具有学习效应和工件可拒绝的排序问题。在线性和凸资源的两种资源分配函数模型下,目标是确定可接受加工的集合、拒绝工件的集合以及确定接受集合中工件的最优排序,对一类正则目标函数进行分析,证明了这些问题都是多项式时间可解的。

1 问题描述

本文的排序问题可以描述为:给定一台机器和 n 个工件 $J = \{J_1, J_2, \dots, J_n\}$, 假设 n 个工件都在 0 时刻到达, 而同一时刻机器最多加工一个工件, 且不允许中断。首先对工件 J_j 要判断它是否被加工, 则把工件分为可接受集合 A 和可拒绝集合 $\bar{A}$ 。对于可接受集合中的工件通过设计算法得到最优排列 π^* ; 对于可拒绝集合中的工件, 需要赔付一定的拒绝费用 e_j 。令 a_j 代表工件 J_j 的基础加工时间, 工件具有一定的学习效应, 同时工件的实际加工时间可以被表示为资源的函数, 即工件的实际加工时间可以表示为资源的线性函数:

$$p_j = a_j r^\alpha - b_j u_j, j = 1, 2, \dots, n_a. \quad (1)$$

其中: n_a 代表可接受集合中的工件个数, r 表示工件所处加工的位置, α 为学习因子, u_j 代表排序中已经排序的工件 J_j 所消耗的资源量, 且 $0 \leq u_j \leq \bar{u}_j < \frac{a_j n_a}{b_j}$, $\bar{u}_j$ 是可分配给工件 J_j 的资源数量的上限, b_j 是工件 J_j 的正压缩率。

与此同时, 工件的加工时间同样可以表示为资源的凸函数, 即工件的实际加工时间可以被表示为:

$$p_j = \left(\frac{a_j r^\alpha}{u_j} \right)^k, j = 1, 2, \dots, n_a. \quad (2)$$

其中: $k > 0$ 为给定常数。

令 $[j]$ 表示在排序中的第 j 个位置, 目标是考虑在有拒绝工件的条件下, 找到可接受工件集合 A 及可拒绝工件集合 $\bar{A}$, 并在 A 中找到一个最优排序 π 及对应的资源分配 u , 使目标函数

$$f(\pi, u) = \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} \quad (3)$$

最小, 即使得可接受集合中工件的时间表长、总完工时间、工件拒绝费用和所消耗资源费用的线性加权和最小, 其中 $\delta_1 \geq 0, \delta_2 \geq 0, \delta_3 \geq 0, \delta_4 \geq 0$ 均为给定常数, G_j 是工件 J_j 消耗资源的单位费用。本文所研究的问题, 用三参数表示法^[26]可表示为:

$$\begin{aligned} & 1 \mid p_j = a_j r^\alpha - b_j u_j, \text{rej} \mid \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}, \\ & 1 \mid p_j = \left(\frac{a_j r^\alpha}{u_j} \right)^k, \text{rej} \mid \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}. \end{aligned}$$

2 主要结论

2.1 线性资源分配问题

本节讨论加工时间具有线性资源函数关系的可拒绝工件排序问题。

通过 $C_{\max} = \sum_{j=1}^{n_a} p_{[j]}$ 和 $\sum_{j=1}^{n_a} C_{[j]} = \sum_{j=1}^{n_a} (n_a - j + 1) p_{[j]}$, 可得:

$$f(\pi, u) = \delta_1 \sum_{j=1}^{n_a} p_{[j]} + \delta_2 \sum_{j=1}^{n_a} (n_a - j + 1) p_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} =$$

$$\sum_{j=1}^{n_a} [\delta_1 + \delta_2(n_a - j + 1)] p_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} = \sum_{j=1}^{n_a} \omega_j (a_{[j]} j^a - b_{[j]} u_{[j]}) + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} = \sum_{j=1}^{n_a} \omega_j a_{[j]} j^a + \delta_3 \sum_{j \in \bar{A}} e_j + \sum_{j=1}^{n_a} (\delta_4 G_{[j]} - \omega_j b_{[j]}) u_{[j]}. \quad (4)$$

其中: $\omega_j = \delta_1 + \delta_2(n_a - j + 1)$ 。

定理 1 给定任一排序 π , 最优的资源分配可表示为:

$$u_{[j]}^* = \begin{cases} \overline{u_{[j]}}, & \text{当 } \delta_4 G_{[j]} - \omega_j b_{[j]} < 0 \\ u_j \in [0, \overline{u_{[j]}}], & \text{当 } \delta_4 G_{[j]} - \omega_j b_{[j]} = 0 \\ 0, & \text{当 } \delta_4 G_{[j]} - \omega_j b_{[j]} > 0 \end{cases} \quad (5)$$

证明 对于给定排序 π , 由(4)式, 对目标函数求偏导数得 $\frac{\partial f(\pi, u)}{\partial u_{[j]}} = \delta_4 G_{[j]} - \omega_j b_{[j]}, j=1, 2, \dots, n_a$ 。当 $\delta_4 G_{[j]} - \omega_j b_{[j]} < 0$ 时, 最优的资源分配为 $u_{[j]}^* = \overline{u_{[j]}}$; 当 $\delta_4 G_{[j]} - \omega_j b_{[j]} = 0$ 时, 最优的资源分配为 $u_{[j]}^* = u_j \in [0, \overline{u_{[j]}}]$; 当 $\delta_4 G_{[j]} - \omega_j b_{[j]} > 0$ 时, 最优的资源分配为 $u_{[j]}^* = 0$ 。证毕

通过已求得的资源分配, 代入到目标函数中, 为了得到可接受集合中的最优排序 π^* , 令:

$$x_{ij} = \begin{cases} 1, & \text{工件 } J_i \text{ 排在位置 } j \text{ 处} \\ 0, & \text{否则} \end{cases}$$

以及

$$\lambda_{ij} = \begin{cases} \omega_j a_i j^a, & \text{当 } \delta_4 G_i - \omega_j b_i \geq 0 \text{ 且 } 1 \leq j \leq n_a \\ \omega_j a_i j^a + (\delta_4 G_i - \omega_j b_i) \overline{u_i}, & \text{当 } \delta_4 G_i - \omega_j b_i < 0 \text{ 且 } 1 \leq j \leq n_a \\ \delta_3 e_j, & \text{当 } n_a + 1 \leq j \leq n \end{cases} \quad (6)$$

则上述排序问题可以转化为如下指派问题:

$$\begin{aligned} \min z &= \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij} x_{ij}, \\ \text{s. t. } &\begin{cases} \sum_{i=1}^n x_{ij} = 1, j=1, 2, \dots, n \\ \sum_{j=1}^n x_{ij} = 1, i=1, 2, \dots, n \\ x_{ij} = 0 \text{ 或 } 1, i, j=1, 2, \dots, n \end{cases} \end{aligned} \quad (7)$$

由以上分析, 可以给出问题 $1 | p_j = a_j r^a - b_j u_j, \text{rej} | \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}$ 的求解算法如下。

算法 1 步骤 1, 当 $n_a = 0$ 时, 计算 $f(\pi, u, 0) = \sum_{j=1}^n e_j$;

步骤 2, 当 $1 \leq n_a \leq n$ 时, 根据(6)式计算权重 λ_{jr} , 并根据指派问题(7)确定工件的最优排序 π_i^* , 并确定此时的 $f(\pi, u, n_a)$;

步骤 3, 最优目标函数值为 $\min\{f(\pi, u, n_a) | 0 \leq n_a \leq n\}$;

步骤 4, 根据(5)式计算出最优的目标函数值所对应的资源分配 u_j^* 。

定理 2 算法 1 可以解决问题 $1 | p_j = a_j r^a - b_j u_j, \text{rej} | \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in \bar{A}} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}$, 且算法 1 的时间复杂度为 $O(n^4)$ 。

证明 算法 1 中步骤 1、3 和 4 的时间复杂度分别为 $O(n)$, 步骤 2 需要解决 $n-1$ 个工件的指派问题, 指派问题的算法复杂度为 $O(n^3)$, 因此算法 1 总的时间复杂度为 $O(n^4)$ 。证毕

通过上述的分析和讨论, 算法 1 能够在多项式时间算法解决问题 $1 | p_j = a_j r^a - b_j u_j, \text{rej} | \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} +$

$\delta_3 \sum_{j \in A} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}$, 其中由 $0 \leq n_a \leq n$ 可知, 通过遍历全部接受工件数量并利用指派问题计算出相应的目标函数值 $f(\pi, u, n_a)$, 并在 $f(\pi, u, n_a)$ 值中可以得到最优的目标函数值和相应的最优资源分配, 由此算法 1 能够正确的解决此问题。

2.2 凸资源分配问题

本节讨论加工时间具有凸资源函数关系的排序问题:

$$1 \left| p_j = \left(\frac{a_j r^a}{u_j} \right)^k, \text{rej} \right| \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in A} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}.$$

类似于本文 2.1 节, 有:

$$f(\pi, u) = \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j \in A} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} = \sum_{j=1}^{n_a} \omega_j \left(\frac{a_{[j]} j^a}{u_{[j]}} \right)^k + \delta_3 \sum_{j \in A} e_j + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}, \quad (8)$$

其中: $\omega_j = \delta_1 + \delta_2 (n_a - j + 1)$ 。

定理 3 给定任一排序 π , 最优的资源分配可表示为:

$$u_{[j]}^* = \left(\frac{k \omega_j}{\delta_4 G_{[j]}} \right)^{\frac{1}{k+1}} \times (a_{[j]} j^a)^{\frac{k}{k+1}}, j = 1, 2, \dots, n_a. \quad (9)$$

证明 对于给定排序, 由(8)式, 对目标函数求偏导数, 并令 $\frac{\partial f(\pi, u)}{\partial u_{[j]}} = 0, j = 1, 2, \dots, n_a$, 可得结论。证毕

通过已求得的资源分配(9)式, 代入到目标函数(8)式中, 得到目标函数新的表达方式:

$$f(\pi, u^*(\pi)) = (k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}) \times (\delta_4)^{\frac{k}{k+1}} \times \sum_{j=1}^{n_a} \theta_{[j]} \varphi_j + \delta_3 \sum_{j=n_a+1}^n e_{[j]}. \quad (10)$$

其中: $\theta_{[j]} = (G_{[j]} a_{[j]})^{\frac{k}{k+1}}, \varphi_j = (\omega_j j^a)^{\frac{1}{k+1}}$ 。

引理 1^[27] 如果存在两个序列 $\{x_i\}, \{y_i\}, i = 1, 2, \dots, n$, 序列 $\{x_i\}$ 按非减顺序排列, 而序列 $\{y_i\}$ 按非增顺序排列, 则 $\sum_{i=1}^n x_i y_i$ 最小。

不难看出, φ_j 随着 j 的增大是逐渐递减的。由引理 1 知接受的工件按照 θ_j 的 SPT 规则不减排序, 因此在接下来的讨论中, 将工件按照 θ_j 的 SPT 规则不减排序, 即 $\theta_1 \leq \theta_2 \leq \dots \leq \theta_n$ 。本文提出一种基于动态规划算法用于

求解问题 $1 \left| p_j = \left(\frac{a_j r^a}{u_j} \right)^k, \text{rej} \right| \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j=n_a+1}^n e_{[j]} + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}$ 。

现定义一种状态变量 $(i, j)_{m_a}$, 其中 m_a 表示接受工件的数量, i 表示工件 $\{J_1, J_2, \dots, J_i\}$ 需要被考虑, $j (0 \leq j \leq i)$ 表示工件接受集中部分工件是已经被排序的。令 $F_{m_a}(i, j)$ 代表在 $(i, j)_{m_a}$ 情况下的最优排序下的最优目标函数值, $S_{m_a}(i, j)$ 代表在 $(i, j)_{m_a}$ 情况下任意排序的解, 对于给定接受集数量值 n_a , 根据(10)式分别计算 $\varphi_j, j = 1, 2, \dots, n_a$ 且有 $\varphi_1 > \varphi_2 > \dots > \varphi_{n_a}$ 。由引理 1 可将第 l 个位置的接受工件匹配相对应的 φ_l , 其中 $J_l \in \{J_1, J_2, \dots, J_i\}$, 因此 $S_{m_a}(i, j)$ 必须是在以前的状态下通过以下两个决策之一来构建的:

1) 若 J_i 是接受工件并进行排序, 则有:

$$F_{m_a}(j, r) = F_{m_a}(j-1, r-1) + (k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}) \times (\delta_4)^{\frac{k}{k+1}} \times \varphi_j \times \theta_i;$$

2) 若 J_i 是拒绝工件并支付拒绝费用, 则有 $F_{m_a}(j, r) = F_{m_a}(j-1, r) + \delta_3 e_i$, 边界条件需满足 $F_{m_a}(0, 0) = 0$, $F_{m_a}(i, j) = +\infty$ 。

通过上述的分析, 问题 $1 \left| p_j = \left(\frac{a_j r^a}{u_j} \right)^k, \text{rej} \right| \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j=n_a+1}^n e_{[j]} + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]}$ 的求解算法如下。

算法 2 步骤 1, 将工件按照 θ_j 按照 SPT 规则不减排序;

步骤 2, 当 $0 \leq n_a \leq n$ 时, 通过(10)式计算 φ_j 的值, 并初始化 $F_{m_a}(i, j) = \begin{cases} 0, & \text{如果 } i=j=0 \\ +\infty, & \text{其他} \end{cases}$;

步骤 3, 对于 $j = 0, 1, \dots, \min\{m_a, i\}$, 计算:

$$F_{m_a}(i, j) = \begin{cases} F_{m_a}(i-1, j-1) + (k^{\frac{-k}{k+1}} + k^{\frac{1}{k+1}}) \times (\delta_4)^{\frac{k}{k+1}} \times \varphi_j \times \theta_i \\ F_{m_a}(i-1, j) + \delta_3 e_i \end{cases}; \quad (11)$$

步骤 4, 最优目标函数值为 $f^*(\pi, u, n_a) = \min\{F_{m_a}(n, m_a) | 0 \leq m_a \leq n\}$, 并通过回溯可以得到最优的接受工件排序序列 π^* 。

步骤 5, 根据(9)式计算出最优的目标函数值所对应的资源分配 u_j^* 。

定理 4 问题 1 $\left| p_j = \left(\frac{a_j r^a}{u_j} \right)^k, \text{rej} \left| \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j=n_a+1}^n e_{[j]} + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} \right. \right.$ 时间复杂度为 $O(n^3)$ 。

证明 对于每一个给定的 n_a 值, 算法需要递归 i 从 1 到 n 和 j 从 0 到 i , 共需要 $O(n^2)$ 时间, 与此同时计算 $F_{m_a}(i, j)$ 需要 $O(n)$ 时间, 因此算法 2 总的复杂度为 $O(n^3)$, 定理得证。证毕

通过上述的分析和讨论, 问题 1 $\left| p_j = \left(\frac{a_j r^a}{u_j} \right)^k, \text{rej} \left| \delta_1 C_{\max} + \delta_2 \sum_{j=1}^{n_a} C_{[j]} + \delta_3 \sum_{j=n_a+1}^n e_{[j]} + \delta_4 \sum_{j=1}^{n_a} G_{[j]} u_{[j]} \right. \right.$ 能够在多项式时间算法解决, 通过遍历全部的 $i(1 \leq i \leq n)$ 和 $j(0 \leq j \leq i)$ 值, 并利用算法 2 可得最优的接受工件数量值 n_a 、工件排序 π^* 和目标函数值 $f^*(\pi, u, n_a)$, 进而得到最优的资源分配 u_j^* 。由此, 此动态规划算法能够在多项式时间 $O(n^3)$ 内正确的解决上述问题。

3 数值算例

本文将通过数值实验验证算法 1 和算法 2 的有效性。该程序在 CPU Inter @ Core i7 2.4 GHz, 8 GB RAM 和 Window 7 系统的个人计算机上通过 Matlab 进行运行, 其中: 1) 加工的个数为 $n \in \{50, 100, 150, 200, 250\}$; 2) 工件的单位资源权重 $G_j \in (0, 1)$; 3) 常数 $k=2, \delta_1, \delta_2, \delta_3, \delta_4$ 为 $[0.5, 1.5]$ 上的随机数; 4) 学习因子 $\alpha = -0.35$ 。

表格 1、2 分别表示算法 1、2 在下述 3 种情况下的结果值和 CPU 运算时间, 其中: 情况 A, 工件的初始数据中 a_j, b_j, e_j 为 $[1, 100]$ 上满足均匀分布的随机数; 情况 B, a_j, b_j 为 $[50, 100]$, e_j 为 $[1, 50]$ 上满足均匀分布的随机数; 情况 C, a_j, b_j 为 $[1, 50]$, e_j 为 $[50, 100]$ 上满足均匀分布的随机数。

表 1 算法 1 在不同情况下的 CPU 运行时间

Tab. 1 CPU times of Algorithm 1 under different conditions

s

情况分类	CPU 运行时间取值	n				
		50	100	150	200	250
情况 A	最大值	1.741 1	13.840 3	66.186 7	217.693 7	552.643 2
	均值	1.563 2	13.500 1	66.175 7	210.198 0	541.171 5
情况 B	最大值	1.518 8	13.005 8	65.330 6	210.606 4	511.606 7
	均值	1.454 1	12.661 8	63.660 8	208.322 5	502.813 3
情况 C	最大值	1.528 2	12.196 6	60.984 3	196.813 1	496.310 5
	均值	1.446 5	11.935 1	60.642 2	196.329 7	494.778 3

表 2 算法 2 在不同情况下的 CPU 运行时间

Tab. 2 CPU times of Algorithm 2 under different conditions

s

情况分类	CPU 运行时间取值	n				
		50	100	150	200	250
情况 A	最大值	0.473 6	3.677 6	12.542 1	29.365 0	59.028 5
	均值	0.459 4	3.431 4	11.941 6	28.094 4	58.924 7
情况 B	最大值	0.432 4	3.749 3	11.529 7	28.568 1	55.619 5
	均值	0.416 9	3.394 7	11.450 3	27.579 5	55.234 1
情况 C	最大值	0.451 1	3.618 5	12.449 7	29.516 3	58.959 5
	均值	0.433 8	3.561 1	11.568 2	27.785 4	55.203 4

从表 1、表 2 可以看出,情况 A 的 CPU 运行时间均大于情况 B 的 CPU 运行时间,当最大完工时间费用参数小于拒绝费用时(情况 C),CPU 运行时间最短。且当计算工件数量较少的情况下,算法 1、2 均可在较短的时间内得到最优排序并得出最优值。但在工件数量较多时,算法 1 会比算法 2 花费更多的时间,当工件数量为 250 时,算法 1 的 CPU 运算时间可达 552.643 2 s。

4 结论

本文主要考虑了存在学习效应和拒绝工件的资源分配单机排序问题。目标是在考虑在线性资源和凸资源分配加工时间的基础上求得可接受工件集合 A ,可拒绝工件集合 $\bar{A}$ 以及确定接受集中的最优排序 π^* 和所对应的资源分配,使工件的时间表长,总完工时间,资源耗费费用和工件拒绝总费用的线性加权和最小。对此两类问题提出相应的最优求解算法与时间复杂度分析,并在 3 种不同参数下给出了相应的数值模拟。在以后的研究中,可以考虑工件的加工时间有可能存在着其他的影响(比如恶化效应),或者考虑多机的情况(比如流水作业)。

参考文献:

- [1] BARTAL Y, LEONARDI S, SPACEAMELA A M, et al. Multiprocessor scheduling with rejection[J]. SIAM Journal Discrete Mathematics, 2000, 5: 64-78.
- [2] ENGELS D W, KARGER D R, KOLLIPOULOS S G. Techniques for scheduling with rejection[J]. Journal of Algorithms, 2003, 49: 175-191.
- [3] ZHANG L Q, LU L F, YUAN J J. Single machine scheduling with release dates and rejection[J]. European Journal of Operational Research, 2009, 198: 975-978.
- [4] SENGUPTA S. Algorithms and approximation schemes for minimum lateness/tardiness scheduling with rejection[J]. Lecture Notes in Computer Science, 2003, 2748: 79-90.
- [5] LI S S, YUAN J J. Parallel-machine scheduling with deteriorating jobs and rejection[J]. Theoretical Computer Science, 2010, 411: 3642-3650.
- [6] CHENG Y S, SUN S J. Scheduling linear deteriorating jobs with rejection on a single machine[J]. European Journal of Operational Research, 2009, 194(1): 18-27.
- [7] 慕迪, 刘培海. 带拒绝和到达时间的单机排序问题[J]. 华东理工大学学报(自然科学版), 2017, 43(6): 890-894.
MU D, LIU P H. Single machine scheduling with job rejection and release dates[J]. Journal of East China University of Science and Technology (Natural Science Edition), 2017, 43(6): 890-894.
- [8] 刘晓霞, 余山杉, 罗文昌. 工件有到达时间且拒绝工件总个数受限的单机平行分批排序问题的近似算法[J]. 运筹学学报, 2020, 24(1): 131-139.
LIU X X, YU S S, LUO W C. Approximation algorithms for single machine parallel-batch scheduling with release dates subject to the number of rejected jobs not exceeding a given threshold[J]. Operations Research Transactions, 2020, 24(1): 131-139.
- [9] SHABTAY D, GASPARD N, KASPI M. A survey on off-line scheduling with rejection[J]. Journal of Scheduling, 2013, 16: 3-28.
- [10] 唐恒永, 赵传立. 排序引论[M]. 北京: 科学出版社, 2002.
TANG H Y, ZHAO C L. Scheduling introduction[M]. Beijing: Science Press, 2002.
- [11] 唐国春, 张峰, 罗守成, 等. 现代排序论[M]. 上海: 上海科学普及出版社, 2003.
TANG G C, ZHANG F, LUO S C, et al. Theory of modern scheduling[M]. Shanghai: Shanghai Science Popularization Press, 2003.
- [12] 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.
- [13] VICKSON R G. Two single machine sequencing problems involving controllable job processing times[J]. AIIE Transactions, 1980, 12(3): 258-262.
- [14] SHABTAY D, STEINER G. A survey of scheduling with controllable processing times[J]. Discrete Apply Math, 2007, 155(13): 1643-1666.
- [15] LEYVAND Y, SHABTAY D, STEINER G. A unified approach for scheduling with convex resource consumption functions using positional penalties[J]. European Journal of Operational Research, 2010, 206(2): 301-312.
- [16] MOSHEIOV G, SIDNEY J B. Scheduling with general job-dependent learning curves[J]. European Journal of Operational

Research, 2003, 147: 665-670.

- [17] WU C C, YIN Y Q, CHENG S R. Single-machine and two-machine flow shop scheduling problems with truncated position-based learning functions[J]. Journal of the Operational Research Society, 2013, 64: 147-156.
- [18] 王雪茹, 白雪莲, 王吉波, 等. 基于截断学习效应的加权总完工时间流水作业排序问题研究[J]. 重庆师范大学学报(自然科学版), 2017, 34(5): 12-17.
WANG X R, BAI X L, WANG J B, et al. Flow shop scheduling problem with truncated learning effects[J]. Journal of Chongqing Normal University (Natural Science), 2017, 34(5): 12-17.
- [19] AZZOUZ A, ENNIGROU M, SAID L B. Scheduling problems under learning effects: Classification and cartography[J]. International Journal of Production Research, 2018, 56: 1642-1661.
- [20] WANG D, WANG M Z, WANG J B. Single-machine scheduling with learning effect and resource-dependent processing times[J]. Computers & Industrial Engineering, 2010, 59(3): 458-462.
- [21] WANG J B, WANG J J. Research on scheduling with job-dependent learning effect and convex resource dependent processing times[J]. International Journal of Production Research, 2015, 53: 5826-5836.
- [22] LIU Y, FENG Z. Two-machine no-wait flowshop scheduling with learning effect and convex resource-dependent processing times[J]. Computers & Industrial Engineering, 2014, 75: 170-175.
- [23] GAO F, LIU M, WANG J J, et al. No-wait two-machine permutation flow shop scheduling problem with learning effect, common due date and controllable job processing times[J]. International Journal of Production Research, 2018, 56: 2361-2369.
- [24] 姜昆. 带凸资源和恶化效应的单机松弛窗口排序[J]. 运筹与管理, 2020, 29(7): 105-109.
JIANG K. Single machine slack due window scheduling with convex resource and deterioration effect[J]. Operations Research and Management Science, 2020, 29(7): 105-109.
- [25] 杨盛武, 刘恒, 耿新娜, 等. 具有凸资源约束的单机指数学习效应问题[J]. 数学的实践与认识, 2020, 50(10): 15-20.
YANG S W, LIU H, GENG X N, et al. Single machine exponential learning effect problems with convex resource constraint[J]. Mathematics In Practice and Theory, 2020, 50(10): 15-20.
- [26] GRAHAM R L, LAWLER E L, LENSTRA J K, et al. Optimization and approximation in deterministic sequencing and scheduling: a survey[J]. Annals of Discrete Mathematics, 1979, 5: 287-326.
- [27] HARDY G H, LITTLEWOOD J E, POLYA G. Inequalities[M]. Cambridge: Cambridge University Press, 1967.

Operations Research and Cybernetics

Research on Resource Constraint Scheduling with Job Rejection and Learning Effect

GUO Feng, WANG Jibo

(School of Science, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: [Purposes] Single machine scheduling with learning effect and job rejection is considered here. Under the constraint of linear and convex resource allocation functions, the goal is to obtain the acceptance job set, the rejection job set and the optimal sequence of acceptance job set, such that the weighted sum of makespan, total completion time, resource consumption cost and job rejection cost is minimized. [Methods] For the linear resource allocation function, if the number of rejection jobs is given, the problem can be transformed into the assignment problem. For the convex resource allocation function, the problem can be solved by using the dynamic programming. [Findings] Polynomial time algorithms for these two problems is presented, and the time complexity is $O(n^4)$ and $O(n^2 \log n)$, respectively, where n is the number of jobs. [Conclusions] The algorithms analysis and CPU time show that the proposed algorithms are very effective.

Keywords: scheduling; job rejection; resource allocation; learning effect

(责任编辑 黄颖)