



同类平行机下工件有任意到达时间的在线排序

献给韩继业教授 90 寿辰

成夏炎¹, 赵聪聪², 马丽娜³, 李荣珩^{4,5*}

1. 湖南第一师范学院数学与统计学院, 长沙 410205;

2. 湖南师范大学数学与统计学院, 长沙 410081;

3. 湖南省沅陵县第一中学, 怀化 419600;

4. 云南财经大学统计与数学学院, 昆明 650221;

5. 湖南师范大学数学与统计学院计算与随机数学教育部重点实验室, 复杂系统的控制与优化湖南省高校重点实验室, 长沙 410081

E-mail: cxy175@163.com, 786670120@qq.com, 75042481@qq.com, lirongheng@hunnu.edu.cn

收稿日期: 2024-03-31; 接受日期: 2024-10-08; 网络出版日期: 2024-12-09; * 通信作者

国家自然科学基金 (批准号: 12471303) 和湖南省教育厅科学研究 (批准号: 16A126) 资助项目

摘要 本文研究同类平行机环境下的在线排序问题, 其中工件具有任意到达时间, 目标为最小化最大完工时间. 所讨论机器的速度, 除了最后一台为 s ($s > 1$) 外, 其余 $m - 1$ 台机器的速度均为 1. 本文分析了列表 (list scheduling, LS) 算法的性能, 得到了机器数 $m = 2$ 及 $m \geq 3$ 时 LS 算法的竞争比分别为 $\frac{3+\sqrt{5}}{2}$ 和 $4 - \frac{4}{m+1}$, 并在一般情形下设计了一个竞争比不超过 3.8626 的更好的算法—改良列表 (modified list scheduling, MLS) 算法.

关键词 在线排序 同类平行机 到达时间 算法 竞争比

MSC (2020) 主题分类 90B35, 90C27

1 引言

排序问题中, 给定 n 个工件的集合 $L = \{J_1, J_2, \dots, J_n\}$, 任意工件 J_j 的长度用正数 p_j 表示, m 台机器记为 $M_1, M_2, \dots, M_m$, 每台机器 M_i 在任一时刻只能加工一个工件. 在线排序是指工件序列一个一个地在线出现, 在不知道后面工件任何信息的情形下, 当一个工件到达就必须立即为其安排一台机器加工, 只有当前工件被安排后才能获得后面是否还有工件到来的信息. 最后使所有工件完工的目标函数达到最优. 本文所讨论的目标是在这个排序下使该工件序列 L 加工完成的最后完工时间最小, 即最大完工时间最小. 通常使用学界公认的三参数表示法 $\alpha|\beta|\gamma$ 表示排序问题, 其中 α 为机器环境, β 为工件特征, γ 为目标函数; 在线排序问题的三参数表示法为 $\alpha|\text{online}|\gamma$. 机器环境中的同型平行机 (也称为相同平行机) 用 P 表示, 同类平行机用 Q 表示. 求解在线排序问题的方案被称为在线排序算法. 一般用竞争比衡量一个算法性能的好坏.

英文引用格式: Cheng X Y, Zhao C C, Ma L N, et al. Online scheduling for jobs with arbitrary release times on uniform machine (in Chinese). Sci Sin Math, 2025, 55: 221–236, doi: 10.1360/SSM-2024-0095

对于 m 台同型平行机环境下的在线排序问题 $P_m|\text{online}|C_{\max}$, 最简单的在线排序算法是由 Graham^[14] 给出的列表 (list scheduling, LS) 算法. LS 算法是经常被用到的非中断在线算法的简单例子, 它总是将当前工件安排到能将其最早加工完成的机器上. Graham 已证明了 m 台同型平行机下 LS 算法的竞争比为 $2 - \frac{1}{m}$, 且当 $m = 2, 3$ 时 LS 算法是最好的. 当机器数 $m \geq 4$ 时, Galambos 和 Woeginger^[13] 将这个界进行了改进. 第一个成功将竞争比值从 2 变小的是 Bartal 等^[3], 后来 Chen 等^[6]、Karger 等^[17]、Albers^[1] 给出了一些比 LS 算法要好的算法. 据我们所知, 对于机器数 m 为任意值时当前最好的竞争比为 1.9201, 这是由 Fleischer 和 Wahl^[12] 给出的. 基于目前已知的 3 个最优值下界 $\sum p_j/m$, $p(1)$ 及 $\max_{1 \leq k \leq \lfloor \frac{n-1}{m} \rfloor} \sum_{j=km-k+1}^{km+1} p(j)$, Tan 和 Li^[22] 给出了同型平行机器数 $m \geq 4$ 时较好的竞争比伪下界, 其中 $p(1)$ 和 $p(j)$ 分别表示工件集中加工时间第 1 大和第 j 大的工件长度.

目标为最小化最大完工时间的 m 台同类平行机环境下的在线排序问题记为 $Q_m|\text{online}|C_{\max}$, 其中 $m \geq 2$. 用 s_i (一般情形下假设 $s_i \geq 1$) 表示机器 M_i 的加工速度, 则工件 J_j 在机器 M_i 上加工所用的时间为 $\frac{p_j}{s_i}$, 其中 $j = 1, 2, \dots, n, i = 1, 2, \dots, m$. 不失一般性, 假定 $s_1 \leq s_2 \leq \dots \leq s_m$, 工件、机器在零时刻是可用的, 同时, 当一个工件在加工时不允许将其中断加工, 而让另一个没有加工的工件抢先加工. Cho 和 Sahni^[8] 最先研究了 m 台同类平行机下的在线排序问题. 他们证明了 $Q_m|\text{online}|C_{\max}$ 的 LS 算法的竞争比不超过 $\frac{1+\sqrt{m-1}}{\sqrt{2}}$; 当 $s_i = 1$ ($i = 1, 2, \dots, m-1$) 且 $s_m = s > 1$ 时, LS 算法的竞争比为 $1 + \frac{m-1}{m+s-1} \min\{s, 2\} \leq 3 - \frac{4}{m+1}$, 且当 $s = 2$ ($m \geq 3$) 时, 界可达到 $3 - \frac{4}{m+1}$. 对于 $Q_2|\text{online}|C_{\max}$, 即 2 台同类平行机下的在线排序问题, Epstein 等^[11] 证明了 LS 算法的竞争比为 $\min\{\frac{2s+1}{s+1}, \frac{s+1}{s}\}$, 且此时 LS 算法是最优的在线排序算法, 其中 $s = s_2/s_1$ 表示机器速度之比. Cai 和 Yang^[5] 考虑了 $Q_3|\text{online}|C_{\max}$. 他们令 $s = s_2/s_1$ 和 $t = s_3/s_2$ 为机器速度之比, 证明了当机器速度比值 $(s, t) \in G_1 \cup G_2$ 时, 贪婪算法—LS 算法是一个最优的在线算法, 其中 $G_1 = \{(s, t) \mid 1 \leq t < \frac{1+\sqrt{31}}{6}, s \geq \frac{3t}{5+2t-6t^2}\}$, $G_2 = \{(s, t) \mid t \geq 1+s, s \geq 1, t \geq 1\}$. 此外, 对于更一般的速度比, 他们给出了竞争比的一个上界. Aspnes 等^[2] 最先考虑了为 $Q_m|\text{online}|C_{\max}$ 设计一个比 LS 算法更好的算法. 他们给出了一个新算法, 在确定情形下的竞争比为 8, 在随机情形下的竞争比为 $2e \approx 5.436$. 后来, Berman 等^[4] 将相应比值分别改进至 $3 + \sqrt{8} \approx 5.828$ 和 $\frac{3.31}{\ln 2.155} \approx 4.311$. 当机器数 $m \geq 4$ 时, Li 和 Shi^[20] 给出了一个竞争比为 2.8795 的新算法, 而 LS 算法相应的竞争比却为 3, 并证明了当 $m \leq 3$ 且 $s_i = 1$ ($i = 1, 2, \dots, m-1$), $s_m = 2$ 时 LS 算法是最优的. 此外, 当 $s_i = 1$ ($i = 1, 2, \dots, m-1$) 及 $s_m = s > 1$ 时, 可将界 $3 - \frac{4}{m+1}$ 进行改进. 当 $m \geq 4$ 及 $1 \leq s \leq 2$ 时, Cheng 等^[7] 给出了一个竞争比不超过 2.45 的算法. Dolgui 等^[9] 将问题推广为 k 台机器速度是 s ($1 < s \leq 2$)、其他 $m-k$ 台机器速度是 1 的情形, 针对该问题设计的在线算法的竞争比不超过 2.618. 另外, Ebenlendr 和 Sgall^[10] 改进机器数 m 为任意值时的下界至 2.564. Jež 等^[16] 给出了机器数 $m \leq 11$ 时的下界. 关于同类平行机下工件不含到达时间的在线排序相关算法竞争比如表 1 所示.

Li 和 Huang^[18, 19] 在机场取货物流的应用中将 m 台同型平行机下的 Graham 在线排序问题进行了推广. 他们依据在线订单模式描述推广的在线排序模型. 一个订单包含的信息是一个工件的到达时间和它的长度, 工件订单在开始时一个接一个在线出现, 订单一到达, 排序者就需要给相应工件安排一台机器和一个时间窗口加工该工件, 只有当前工件被安排好以后才能知道是否有下一个工件订单. 此模型中后到的订单工件的到达时间可能小于先到的订单工件的到达时间, 故称此模型为工件有任意到达时间的在线排序问题. 这个问题可以按如下的方式定义: 给定工件序列 $\{J_1, J_2, \dots, J_n\}$, 工件 J_j 的到达时间和长度分别为 r_j 和 p_j . 设当前要安排的是到达时间和长度分别为 r_j 和 p_j 的工件 J_j , 排序者只有在工件 J_j 被分配了一台机器和一个加工时间窗口后才能知道后面是否还有工件订单的到来. 这里工件达到时间序列 $\{r_1, r_2, \dots, r_n\}$ 是任意的. 如果所有工件的到达时间是 0, 则此问题就变为

表 1 同类平行机下工件不含到达时间的在线排序问题相关算法竞争比

算法	竞争比	机器数	机器速度
LS	$\frac{1+\sqrt{m-1}}{\sqrt{2}}$ [8]	m 任意	—
LS	$3 - \frac{4}{m+1}$ [8]	m 任意	$s_i = 1 (i = 1, 2, \dots, m-1), s_m = s > 1$
LS	$\min\{\frac{2m-3+s}{m-1}, \frac{2m-2+s}{m-1+s}\}$ [21]	m 任意	$s_i = 1 (i = 1, 2, \dots, m-1), s_m = s \leq 1$
LS	$\min\{\frac{2s+1}{s+1}, \frac{s+1}{s}\}$ [11]	$m = 2$	$s = \frac{s_2}{s_1}$
LS	$\frac{1+s+2st}{s+st}$ [5]	$m = 3$	$s_1 = 1, s_2 = s, s_3 = st, s, t \geq 1, (s, t) \in G_1,$ $G_1 = \{(s, t) \mid 1 \leq t \leq \frac{1+\sqrt{31}}{6}, s \geq \frac{3t}{5+2t+6t^2}\}$
LS	$\frac{1+s}{st}$ [5]	$m = 3$	$s_1 = 1, s_2 = s, s_3 = st, s, t \geq 1, (s, t) \in G_2,$ $G_2 = \{(s, t) \mid s(t-1)t \geq 1+s, s \geq 1, t \geq 1\}$
LS	$\min\{\frac{1+s+2st}{s+st}, \frac{1+s}{st}, \frac{1+s+3st}{1+s+st}\}$ [5]	$m = 3$	$s_1 = 1, s_2 = s, s_3 = st, s, t \geq 1$
LS	$\min\{\frac{1+s_2+s_3}{s_3}, \frac{1+s_2+2s_3}{s_2+s_3}, \frac{1+s_2+3s_3}{1+s_2+s_3}\}$ [15]	$m = 3$	—
Assign-R	8 [2]	m 任意	—
Assign-R	$3 + \sqrt{8}$ [4]	m 任意	—
算法 A	2.8795 [20]	$m \rightarrow \infty$	$s_i = 1 (i = 1, 2, \dots, m-1), s_m = 2$
算法 S	2.45 [7]	$m \geq 4$	$s_i = 1 (i = 1, 2, \dots, m-1), s_m = s (1 \leq s \leq 2)$
R 竞争算法 A	2.564 [10]	m 任意	$s_i = \alpha^{-i} (i = 1, 2, \dots, m), \alpha > 1$
参数算法	2.618 [9]	m 任意	$s_i = s (1 < s \leq 2, i = 1, 2, \dots, k),$ $s_t = 1 (t = k+1, k+2, \dots, m)$
TwoGroups	$\frac{2(s+2)}{3}$ [21]	$m = 3$	$s_1 = s_2 = 1, s_3 = s, 0 \leq s \leq -1 + \frac{\sqrt{10}}{2}$
TwoGroups	$\frac{2(s+3)}{2s+3}$ [21]	$m = 3$	$s_1 = s_2 = 1, s_3 = s, -1 + \frac{\sqrt{10}}{2} < s \leq 1$

Graham 的经典排序问题.

本文考虑速度分别为 $s_1 = s_2 = \dots = s_{m-1} = 1, s_m = s > 1$ 的 m 台同类平行机下的工件有任意到达时间的在线排序问题, 这在之前的研究中是没有出现过的. 本文证明了当 $m = 2$ 时 LS 算法的竞争比为 $\frac{3+\sqrt{5}}{2}$, 当 $m \geq 3$ 时 LS 算法的竞争比为 $4 - \frac{4}{m+1}$, 并给出了一个竞争比不超过 3.8626 的更好的算法—改良列表 (modified list scheduling, MLS) 算法.

本文余下内容分 4 节进行论述. 第 2 节主要介绍需要用到的几个定义以及工件有任意到达时间的 LS 算法. 第 3 节讨论给定速度的 m 台同类平行机下工件有任意到达时间的 LS 算法的竞争比. 第 4 节主要介绍 MLS 算法, 并对其性能进行详细分析. 第 5 节对全文进行总结.

2 一些定义及 LS 算法

首先给出衡量算法性能好坏的竞争比的概念.

定义 2.1 给定速度为 $s_1, s_2, \dots, s_m$ 的 m 台同类平行机器集 $M = \{M_1, M_2, \dots, M_m\}$ 及工件集 $L = \{J_1, J_2, \dots, J_n\}$. 不失一般性, 假设 $s_1 \leq s_2 \leq \dots \leq s_m$. 工件 J_j 的到达时间与工件长度分别用 r_j 和 p_j 表示, 算法 A 为一启发式算法, $C_{\max}^A(L)$ 和 $C_{\max}^{\text{OPT}}(L)$ 分别表示算法 A 的最后完工时间和最优离线算法的最后完工时间. 下式表示 $Q_m | \text{online}, r_j | C_{\max}$ 问题算法 A 的竞争比:

$$R(m, A) = \sup_L \frac{C_{\max}^A(L)}{C_{\max}^{\text{OPT}}(L)}.$$

接下来的论述中, 若没有特别说明, 均认为 $s_i = 1, i = 1, 2, \dots, m-1, s_m = s > 1$.

在工件有任意到达时间的在线排序中, 空闲时间间隔是非常重要的概念.

定义 2.2 假设 J_j 是当前要安排加工的工件, 其到达时间和长度分别为 r_j 和 p_j . 称机器 M_i 对工件 J_j 有空闲时间间隔, 若存在满足下面两个条件的时间间隔 $[T_1, T_2]$.

(1) 机器 M_i 在区间 $[T_1, T_2]$ 是空闲的, 且一个到达时间为 T_2 的工件已被安排在 M_i 上的 T_2 时刻开始加工.

(2) 当 $1 \leq i \leq m-1$ 时, $T_2 - \max\{T_1, r_j\} \geq p_j$; 当 $i = m$ 时, $T_2 - \max\{T_1, r_j\} \geq \frac{p_j}{s}$.

显然, 若机器 M_i 对工件 J_j 有空闲时间间隔, 则工件 J_j 在空闲时间间隔能被安排在机器 M_i 上加工.

定义 2.3 对于一个在线算法 A , H_i^j 表示机器 M_i ($1 \leq i \leq m$) 加工工件 J_j 前夕的最后完工时间, 简称机器 M_i 的高度. H_i 表示算法 A 下机器 M_i 加工最后一个工件 J_n 前夕的高度.

下面对工件有任意到达时间的 LS 算法进行介绍.

LS 算法

- 1°. 设 H_i^j 是机器 M_i ($i = 1, 2, \dots, m$) 的当前完工时间. 工件 J_j 是算法下的到达时间和工件长度分别为 r_j 和 p_j 的新工件.
- 2°. 假设存在一些对工件 J_j 有空闲时间间隔的机器. 选择对工件 J_j 有空闲时间间隔 $[T_1, T_2]$ 的最小的 T_1 的机器 M_i , 再将工件 J_j 安排在机器 M_i 上, 在空闲时间间隔 $[T_1, T_2]$ 中的 $\max\{T_1, r_j\}$ 时刻开始加工. 否则转入 3°.
- 3°. 令 $k := \max\{i \mid M_i \text{ 能最早加工完工件 } J_j\}$, 将工件 J_j 安排在机器 M_k 上加工.

3 LS 算法的性能分析

本节主要讨论机器环境为同类平行机、工件具有任意到达时间的 LS 算法的竞争比, 其中机器速度分别为 $s_1 = s_2 = \dots = s_{m-1} = 1, s_m = s > 1$. 本节用到的方法为最小反例法.

定理 3.1 对于任意工件集 $L = \{J_1, J_2, \dots, J_n\}$, 有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \begin{cases} 2 + \min\left\{\frac{1}{s}, \frac{s}{s+1}\right\} \leq \frac{3+\sqrt{5}}{2}, & m=2, \\ 2 + \frac{m-1}{s+m-1} \min\{2, s\} \leq 4 - \frac{4}{m+1}, & m \geq 3. \end{cases}$$

证明 设 $L = \{J_1, J_2, \dots, J_n\}$ 是一个最小反例, 即 L 是不满足定理 3.1 且含最少工件的工件集. 显而易见, 对于最小反例 L , 最后的工件 J_n 完工时间是 $C_{\max}^{\text{LS}}(L)$.

首先证明当 $m \geq 2$ 时, 有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \frac{s(m-1)}{s+m-1}. \quad (3.1)$$

由 LS 算法有 $sH_m + p_n \geq sC_{\max}^{\text{LS}}(L)$ 及 $H_i + p_n \geq C_{\max}^{\text{LS}}(L), i \in \{1, 2, \dots, m-1\}$. 故

$$C_{\max}^{\text{LS}}(L) \leq \frac{\sum_{i=1}^{m-1} (H_i + p_n) + (sH_m + p_n)}{s+m-1}. \quad (3.2)$$

令 u_i 为 LS 算法下机器 M_i ($i = 1, 2, \dots, m$) 上的总空闲时间, 则易得下面的不等式:

$$u_i \leq C_{\max}^{\text{OPT}}(L), \quad i = 1, 2, \dots, m,$$

$$C_{\max}^{\text{OPT}}(L) \geq \frac{\sum_{j=1}^n p_j}{s+m-1},$$

$$C_{\max}^{\text{OPT}}(L) \geq r_j + \frac{p_j}{s}, \quad j = 1, 2, \dots, n.$$

由 (3.2) 及上面不等式有

$$\begin{aligned} \frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} &\leq \frac{\sum_{i=1}^{m-1} (H_i + p_n) + (sH_m + p_n)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= \frac{\sum_{j=1}^n p_j + \sum_{i=1}^{m-1} u_i + su_m + (m-1)p_n}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq \frac{(s+m-1)C_{\max}^{\text{OPT}}(L) + \sum_{i=1}^{m-1} C_{\max}^{\text{OPT}}(L) + sC_{\max}^{\text{OPT}}(L) + s(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= 2 + \frac{s(m-1)}{s+m-1}. \end{aligned}$$

接下来证明当 $m \geq 3$ 及 $s \geq 2$ 时有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \frac{2(m-1)}{s+m-1}. \quad (3.3)$$

若不存在 LS 算法下在机器 M_m 上加工但在最优算法下不在机器 M_m 上加工的工件, 则 $H_m \leq C_{\max}^{\text{OPT}}(L)$ 且有

$$C_{\max}^{\text{LS}}(L) \leq H_m + \frac{p_n}{s} \leq 2C_{\max}^{\text{OPT}}(L) \leq \left(2 + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L).$$

若存在这样的工件, LS 算法下在机器 M_m 上加工但在最优算法下不在机器 M_m 上加工, 则令其中最后一个工件 J' 的长度为 t' . 令 $\text{SUC}(J')$ 为在 LS 算法下 J' 之后的安排在机器 M_m 上的所有工件组成的集合, 也表示该集合中所有工件的总长. 故 $\text{SUC}(J')$ 中的所有工件在最优算法下安排在机器 M_m 上. 显然有

$$\text{SUC}(J') \leq sC_{\max}^{\text{OPT}}(L) \quad \text{及} \quad t' \leq C_{\max}^{\text{OPT}}(L).$$

我们分两种情形讨论:

情形 1 LS 算法下 J' 之后机器 M_m 上有空闲时间间隔.

此时, 记 $[a, b]$ 是 LS 算法下机器 M_m 上最后一个空闲时间间隔. 若 $b > \frac{H_m + \frac{p_n}{s}}{2}$, 则有

$$C_{\max}^{\text{LS}}(L) \leq H_m + \frac{p_n}{s} < 2b < 2C_{\max}^{\text{OPT}}(L) \leq \left(2 + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L).$$

若 $b \leq \frac{H_m + \frac{p_n}{s}}{2}$, 则

$$\text{SUC}(J') \geq sH_m - sb \geq \frac{sH_m - p_n}{2}.$$

所以

$$C_{\max}^{\text{LS}}(L) \leq H_m + \frac{p_n}{s} \leq \frac{2\text{SUC}(J') + p_n}{s} + \frac{p_n}{s} = \frac{2(\text{SUC}(J') + p_n)}{s}. \quad (3.4)$$

在最优算法下, 若 J_n 安排在机器 M_m 上加工, 则有 $\text{SUC}(J') + p_n \leq sC_{\max}^{\text{OPT}}(L)$. 故由 (3.4) 有

$$C_{\max}^{\text{LS}}(L) \leq \frac{2(\text{SUC}(J') + p_n)}{s} \leq 2C_{\max}^{\text{OPT}}(L) \leq \left(2 + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L).$$

在最优算法下, 若 J_n 不是安排在机器 M_m 上加工, 则有 $\text{SUC}(J') \leq sC_{\max}^{\text{OPT}}(L)$ 及 $p_n \leq C_{\max}^{\text{OPT}}(L)$. 由 (3.4) 也有

$$\begin{aligned} C_{\max}^{\text{LS}}(L) &\leq \frac{2(\text{SUC}(J') + p_n)}{s} \leq \frac{2(sC_{\max}^{\text{OPT}}(L) + C_{\max}^{\text{OPT}}(L))}{s} \\ &\leq \left(2 + \frac{2}{s}\right)C_{\max}^{\text{OPT}}(L) \leq \left(2 + \frac{2(m-1)}{s+m-1}\right)C_{\max}^{\text{OPT}}(L). \end{aligned}$$

所以, 在 LS 算法下只要机器 M_m 上 J' 后存在空闲时间间隔, (3.3) 就成立.

情形 2 LS 算法下 J' 之后机器 M_m 上没有空闲时间间隔.

由 LS 算法有

$$H_i + t' \geq H_m - \frac{\text{SUC}(J')}{s}, \quad i = 1, 2, \dots, m-1.$$

所以, 若在最优算法下工件 J_n 安排在机器 M_m 上加工, 则有 $\text{SUC}(J') + p_n \leq sC_{\max}^{\text{OPT}}(L)$, 且对于 $i = 1, 2, \dots, m-1$ 有

$$\begin{aligned} H_i + C_{\max}^{\text{OPT}}(L) &\geq H_i + t' \geq H_m - \frac{\text{SUC}(J')}{s} = H_m + \frac{p_n}{s} - \frac{\text{SUC}(J') + p_n}{s} \\ &\geq C_{\max}^{\text{LS}}(L) - C_{\max}^{\text{OPT}}(L). \end{aligned} \quad (3.5)$$

若在最优算法下工件 J_n 不是安排在机器 M_m 上加工, 则有 $p_n \leq C_{\max}^{\text{OPT}}(L)$ 及

$$C_{\max}^{\text{LS}}(L) \leq H_i + p_n \leq H_i + C_{\max}^{\text{OPT}}(L), \quad i = 1, 2, \dots, m-1. \quad (3.6)$$

由 (3.5) 及 (3.6), 总有

$$C_{\max}^{\text{LS}}(L) \leq H_i + 2C_{\max}^{\text{OPT}}(L), \quad i = 1, 2, \dots, m-1. \quad (3.7)$$

对于机器 M_m , 有

$$sC_{\max}^{\text{LS}}(L) \leq sH_m + p_n.$$

故结合 (3.7), 有

$$\begin{aligned} \frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} &= \frac{sC_{\max}^{\text{LS}}(L) + (m-1)C_{\max}^{\text{LS}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq \frac{sH_m + p_n + \sum_{i=1}^{m-1}(H_i + 2C_{\max}^{\text{OPT}}(L))}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= \frac{sH_m + p_n + \sum_{i=1}^{m-1}H_i + 2(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq \frac{\sum_{j=1}^n p_j + \sum_{i=1}^{m-1}u_i + su_m + 2(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq \frac{(s+m-1)C_{\max}^{\text{OPT}}(L) + \sum_{i=1}^{m-1}C_{\max}^{\text{OPT}}(L) + sC_{\max}^{\text{OPT}}(L) + 2(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= 2 + \frac{2(m-1)}{s+m-1}. \end{aligned}$$

故 (3.3) 总是成立的. 结合 (3.1), 对于任意的 $m \geq 3$, 都有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \frac{(m-1)}{s+m-1} \min\{2, s\} \leq 4 - \frac{4}{m+1}.$$

当 $m = 2$ 时, 下面的不等式显然成立:

$$sC_{\max}^{\text{LS}}(L) \leq sH_2 + p_n \leq su_2 + \sum_{j=1}^n p_j \leq sC_{\max}^{\text{OPT}}(L) + (1+s)C_{\max}^{\text{OPT}}(L).$$

所以有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \frac{1}{s}.$$

结合 (3.1), 当 $m = 2$ 时有

$$\frac{C_{\max}^{\text{LS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \min \left\{ \frac{1}{s}, \frac{s}{s+1} \right\} \leq \frac{3+\sqrt{5}}{2}.$$

至此, 定理 3.1 证明完毕. □

定理 3.2 对于任意的 $m \geq 2$, 都有

$$R(m, \text{LS}) = \begin{cases} \frac{3+\sqrt{5}}{2}, & m = 2, \\ 4 - \frac{4}{m+1}, & m \geq 3. \end{cases}$$

证明 由定理 3.1 知, 只需要证明当 $m \geq 3$ 时存在一个工件序列 L_0 使得

$$\frac{C_{\max}^{\text{LS}}(L_0)}{C_{\max}^{\text{OPT}}(L_0)} \rightarrow 4 - \frac{4}{m+1}$$

以及当 $m = 2$ 时存在一个工件序列 L_1 使得

$$\frac{C_{\max}^{\text{LS}}(L_1)}{C_{\max}^{\text{OPT}}(L_1)} \rightarrow \frac{3+\sqrt{5}}{2}.$$

对于任意给定的 $m \geq 3$, 令 $s = 2$ 及 $n = (2m-2)(m+1) + 3$. 构造下列工件序列 $L_0 = \{J_1, J_2, \dots, J_n\}$:

- $r_i = k + \frac{1}{2} - \varepsilon$, $p_i = 2\varepsilon$, $0 \leq k \leq m$, $i = k(m+1) + 1$;
- $r_i = k + 1 - \varepsilon$, $p_i = 2\varepsilon$, $0 \leq k \leq m$, $i = k(m+1) + 2$;
- $r_i = k + 1 - \varepsilon$, $p_i = \varepsilon$, $0 \leq k \leq m$, $k(m+1) + 2 < i \leq (k+1)(m+1)$;
- $r_i = 0$, $(m+1)^2 + 1 \leq i \leq (2m-2)(m+1) + 3$;
- $p_i = 1$, $(m+1)^2 + 1 \leq i \leq (2m-2)(m+1)$;
- $p_{(2m-2)(m+1)+1} = p_{(2m-2)(m+1)+2} = m+1$;
- $p_{(2m-2)(m+1)+3} = 2(m+1)$,

其中 ε 为充分小的正数.

除了最后 3 个工件, 我们可将前面的 $(2m-2)(m+1)$ 个工件划分成 $2m-2$ 组 $G_1, G_2, \dots, G_{2m-2}$, 每一组 G_k ($k = 1, 2, \dots, 2m-2$) 包含 $m+1$ 个工件, 即

$$G_k = \{J_{(k-1)(m+1)+1}, J_{(k-1)(m+1)+2}, \dots, J_{(k-1)(m+1)+m+1}\}.$$

为了简单起见, 令 $g(k) = (k-1)(m+1)$ ($k = 1, 2, \dots, 2m-2$), 则工件 $J_{g(k)+j}$ ($k = 1, 2, \dots, 2m-2$, $j = 1, 2, \dots, m+1$) 表示 G_k 组中第 j 个工件.

首先考虑 LS 算法. 每组 G_k ($k = 1, 2, \dots, 2m-2$) 中前两个工件 $J_{g(k)+1}$ 及 $J_{g(k)+2}$ 安排在机器 M_m 上加工, 第 $i+2$ 个工件 $J_{g(k)+i+2}$ 安排在机器 M_{m-i} ($i = 1, 2, \dots, m-1$) 上加工, 最后三个工件均安排在机器 M_m 上加工. 用 $M_i^{\text{LS}}(L_0)$ 表示在 LS 算法下工件序列 L_0 在机器 M_i ($i = 1, 2, \dots, m$) 上的工件集, 则 LS 算法下的工件序列 L_0 表示如下 (如图 1 所示):

$$M_m^{\text{LS}}(L_0) = \{J_{g(1)+1}, J_{g(1)+2}, \dots, J_{g(k)+1}, J_{g(k)+2}, \dots, J_{g(2m-3)+1}, J_{g(2m-3)+2}, J_{n-2}, J_{n-1}, J_n\},$$

$$M_{m-i}^{\text{LS}}(L_0) = \{J_{g(1)+i+2}, \dots, J_{g(k)+i+2}, \dots, J_{g(2m-3)+i+2}\},$$

其中 $1 \leq i \leq m-1$, 所以有 $C_{\max}^{\text{LS}}(L_0) = 4m$.

接下来, 考虑最优算法. 此时, 工件 J_n, J_{n-1}, J_{n-2} 及每组 G_k ($k = m+2, \dots, 2m-2$) 中的工件将先于每组 G_k ($k = 1, 2, \dots, m, m+1$) 中的工件安排. 工件 J_n 安排到机器 M_m 上, 工件 J_{n-1} 及 J_{n-2} 分别安排到机器 M_{m-1} 及 M_{m-2} 上, 每组 G_{m+1+i} ($i = 1, 2, \dots, m-3$) 中的总长为 $m+1$ 的 $m+1$ 个工件均安排在相应机器 M_i 上. 至于前 $m+1$ 组工件, 每组 G_k ($k = 1, 2, \dots, m+1$) 长度为 ε , 共有 $m-1$ 个, 它们是 $J_{g(k)+i+2}$ ($i = 1, 2, \dots, m-1$), 因 LS 算法中的安排我们将之相应安排在机器 M_{m-i} 上. 每组 G_k ($k = 1, 2, \dots, m+1$) 中剩下的两个长度为 2ε 的工件为 $J_{g(k)+1}$ 及 $J_{g(k)+2}$. 故将 $J_{g(i)+2}$ ($i = 1, 2, \dots, m-1$) 安排在机器 M_i 上, 其余的 $m+3$ 个工件 $J_{g(1)+1}, J_{g(2)+1}, \dots, J_{g(m-1)+1}, J_{g(m)+1}, J_{g(m)+2}, J_{g(m+1)+1}, J_{g(m+1)+2}$ 均安排在机器 M_m 上. 用 $M_i^{\text{OPT}}(L_0)$ 表示最优算法下工件序列 L_0 在机器 M_i ($i = 1, 2, \dots, m$) 上的工件集, 则最优算法下工件序列 L_0 的排序结果如下 (图 2 所示):

$$M_m^{\text{OPT}}(L_0) = \{J_n, J_{g(1)+1}, J_{g(2)+1}, \dots, J_{g(m-1)+1}, J_{g(m)+1}, J_{g(m)+2}, J_{g(m+1)+1}, J_{g(m+1)+2}\},$$

$$M_{m-1}^{\text{OPT}}(L_0) = \{J_{n-1}, J_{g(1)+m+1}, J_{g(2)+m+1}, \dots, J_{g(m+1)+m+1}, J_{g(m-1)+2}\},$$

$$M_{m-2}^{\text{OPT}}(L_0) = \{J_{n-2}, J_{g(1)+m}, J_{g(2)+m}, \dots, J_{g(m+1)+m}, J_{g(m-2)+2}\},$$

$$M_{m-i}^{\text{OPT}}(L_0) = G_{m+1+i} \cup \{J_{g(1)+i+2}, J_{g(2)+i+2}, \dots, J_{g(m+1)+i+2}, J_{g(m-i)+2}\},$$

其中 $3 \leq i \leq m-1$.

所以有 $C_{\max}^{\text{OPT}}(L_0) = m+1 + (m+3)\varepsilon$. 因此

$$\frac{C_{\max}^{\text{LS}}(L_0)}{C_{\max}^{\text{OPT}}(L_0)} = \frac{4m}{m+1 + (m+3)\varepsilon} \rightarrow 4 - \frac{4}{m+1} \quad (\varepsilon \rightarrow 0).$$

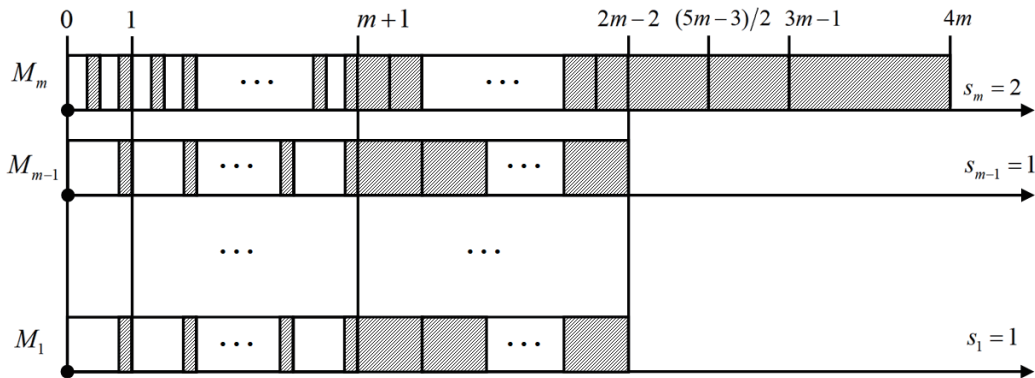
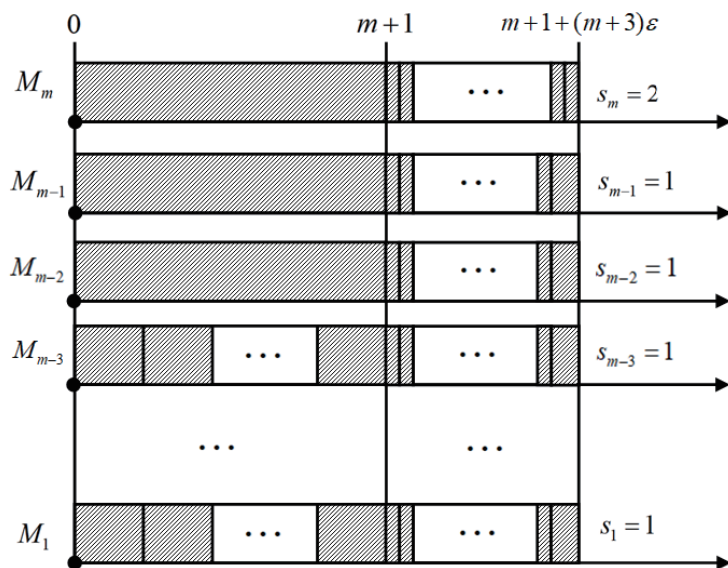


图 1 LS 算法下工件序列 L_0 的排序

图 2 最优算法下工件序列 L_0 的排序

当 $m = 2$ 时, 令 $s = \frac{1+\sqrt{5}}{2}$ 及 $n = 4$. 构造按如下方式工件序列 $L_1 = \{J_1, J_2, J_3, J_4\}$:

- $r_1 = r_2 = 1 - \varepsilon, r_3 = r_4 = 0$;
- $p_1 = \frac{(1+\sqrt{5})\varepsilon}{2}, p_2 = \varepsilon, p_3 = 1, p_4 = \frac{1+\sqrt{5}}{2}$,

其中 ε 是充分小的正数. 则 LS 算法及最优算法下的排序结果如下:

- $M_1^{\text{LS}}(L_1) = \{J_2, J_4\}, M_2^{\text{LS}}(L_1) = \{J_1, J_3\}$;
- $M_1^{\text{OPT}}(L_1) = \{J_3, J_2\}, M_2^{\text{OPT}}(L_1) = \{J_4, J_1\}$.

因此有

$$C_{\max}^{\text{OPT}}(L_1) = 1 + \varepsilon, \quad C_{\max}^{\text{LS}}(L_1) = r_2 + p_2 + p_4 = 1 - \varepsilon + \varepsilon + \frac{1 + \sqrt{5}}{2} = \frac{3 + \sqrt{5}}{2}.$$

所以

$$\frac{C_{\max}^{\text{LS}}(L_1)}{C_{\max}^{\text{OPT}}(L_1)} = \frac{\frac{3+\sqrt{5}}{2}}{1+\varepsilon} \rightarrow \frac{3+\sqrt{5}}{2} \quad (\varepsilon \rightarrow 0).$$

至此, 定理 3.2 证明完毕. $\square$

4 MLS 算法及其性能分析

本节给出一个比 LS 算法更好的新在线算法—MLS 算法. 首先定义一个符号: 给定一批工件 $\{J_1, J_2, \dots, J_n\}$, 对于 $j = 1, 2, \dots, n$, 令

$$R_j := \max \left\{ \frac{p_1 + p_2 + \dots + p_j}{s + m - 1}, r_k + \frac{\sum_{r_{k'} \geq r_k, k' \leq j} p_{k'}}{s + m - 1}, r_k + \frac{p_k}{s}, r_k + p_{\max\{r_k, \lceil s \rceil\}}, k = 1, 2, \dots, j \right\},$$

其中 $p_{\max\{r_k, \lceil s \rceil\}}$ 表示到达时间不小于 r_k 的前 j 个工件中第 $\lceil s \rceil$ 个大工件的长度, 若不存在这一工件, 则令 $p_{\max\{r_k, \lceil s \rceil\}} = 0$. 不难看出, R_j 是最优算法下工件序列 $L = \{J_1, J_2, \dots, J_j\}$ 最后完工时间的一个下界. 为了简便起见, 在不引起混淆的情形下将 R_n 写成 R .

命题 4.1 容易看出 $R_1 \leq R_2 \leq \dots \leq R_n$.

命题 4.2 $\alpha = 3.8626$ 及 $\beta = 1.7012589$ 满足下面两个方程:

$$\frac{1}{\beta^2} + \frac{17(\alpha - \beta)}{33\beta} = 1, \quad \frac{1}{3\beta} + \frac{11}{3} = \alpha.$$

由定理 3.1 可知, LS 算法的最坏性能比是关于 m 的增函数, 且当 $1 \leq s < 2$ 时该最坏性能比是 s 的增函数, $s \geq 2$ 时却是 s 的减函数. 因此, 针对 $m \geq 29$ 以及在 $s = 2$ 的邻域内达到最坏性能比的工件特征将 LS 算法进行了改进, 设计了 MLS 算法. 具体如下:

MLS 算法

0°. 若有 $\lceil s \rceil > \frac{m+1}{10}$ 或 $m < 29$, 则应用 LS 算法, 否则转入 1°.

1°. 在现算法下, 假设 J_j 是当前需要加工的新工件, 且在安排 J_j 之前机器 M_i 的当前完工时间为 H_i . 将相应机器按 $H_1 \leq H_2 \leq \dots \leq H_{m-1}$ 重新排序.

2°. 假设存在对工件 J_j 有空闲时间间隔的机器. 选择对工件 J_j 有空闲时间间隔 $[T_1, T_2]$ 的最小的 T_1 机器 M_i , 再将工件 J_j 安排在机器 M_i 上, 在空闲时间间隔 $[T_1, T_2]$ 中的 $\max\{T_1, r_j\}$ 时刻开始加工. 否则转入 3°.

3°. 令 $k := \max\{i \mid M_i \text{ 能最早加工完工件 } J_j\}$.

4°. 若满足下面的条件:

- (a) $k \leq m - 2$;
- (b) $H_k \leq r_j < H_{k+1}$;
- (c) $H_{k+1} + p_j \leq \alpha R$;
- (d) $r_j \geq \frac{R}{\beta}$,

则将工件 J_j 安排在机器 M_{k+1} 上加工, 否则转入 5°.

5°. 将工件 J_j 安排在机器 M_k 上加工.

下面分析 MLS 算法的最坏性能比. 先给出如下引理.

引理 4.1 若 $\lceil s \rceil > \frac{m+1}{10}$ 或 $m < 29$, 则有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \alpha.$$

证明 若 $\lceil s \rceil > \frac{m+1}{10}$ 或 $m < 29$, 则 MLS 算法就是 LS 算法. 考虑以下 3 种情形.

情形 1 $m < 29$. 此时由定理 3.1 有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 4 - \frac{4}{29} < \alpha.$$

情形 2 $\lceil s \rceil > \frac{m+1}{10}$ 及 $m \geq 39$. 此时有

$$\begin{aligned} \frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} &\leq 2 + \frac{m-1}{s+m-1} \min\{2, s\} = 2 + \frac{2(m-1)}{s+m-1} \\ &\leq 2 + \frac{2(m-1)}{\frac{m+1}{10} - 1 + m - 1} = \frac{42}{11} + \frac{160}{11(11m-19)} \\ &\leq \frac{42}{11} + \frac{160}{11(11 \times 39 - 19)} \\ &< \alpha. \end{aligned}$$

情形 3 $\lceil s \rceil > \frac{m+1}{10}$ 及 $29 \leq m < 39$. 此时有 $\lceil s \rceil > \frac{m+1}{10} \geq 3$, 因此 $s > 3$. 进而有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 2 + \frac{m-1}{s+m-1} \min\{2, s\} = 2 + \frac{2(m-1)}{s+m-1}$$

$$\begin{aligned} &\leq 2 + \frac{2(m-1)}{3+m-1} \leq 2 + \frac{2(39-1)}{3+39-1} \\ &< \alpha. \end{aligned}$$

证毕. $\square$

接下来对于任意工件集 $L = \{J_1, J_2, \dots, J_n\}$, 我们将证明当 $\lceil s \rceil \leq \frac{m+1}{10}$ 及 $m \geq 29$ 时下面的 (4.1) 成立:

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \alpha. \quad (4.1)$$

因此由引理 4.1 及 (4.1) 有如下定理.

定理 4.1 对于任意工件集 $L = \{J_1, J_2, \dots, J_n\}$, 有 $\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \alpha$.

现在开始证明 (4.1). 接下来的证明若没有特别说明, 假设 $\lceil s \rceil \leq \frac{m+1}{10}$ 及 $m \geq 30$ 总是成立的. 另设 $L = \{J_1, J_2, \dots, J_n\}$ 是 (4.1) 的最小反例, 即使得 (4.1) 不成立的最少工件的集合.

定义 4.1 在 MLS 算法下, 若工件 J_j 安排在机器 M_i 加工且 $H_i^j \leq r_j$, 则称 J_j 是 M_i 上的空闲工件; 称时间间隔 $[H_i^j, r_j]$ 是由工件 J_j 产生的空闲时间间隔; 若机器 M_i ($1 \leq i \leq m$) 上在加工完空闲工件 J_j 之后再无空闲工件, 则称空闲工件 J_j 为 M_i 上最后空闲工件.

由空闲工件的定义可知, 在 MLS 算法下任意机器上第一个工件是空闲工件且任意空闲工件一定是在步骤 5° 被安排加工的. 令 (a, b) 是满足下面两条的空闲区间:

- (i) $\frac{R}{\beta} \leq a < b \leq R$;
- (ii) MLS 算法下在区间 (a, b) 中同时空闲的所有机器数是最大的.

在 MLS 算法下, 令机器 M_i 上产生空闲时间间隔 (a, b) 的空闲工件为 $J_{j(i)}$. 现假设

- $A = \{i \mid 1 \leq i \leq m-1, \text{在 MLS 算法下 } M_i \text{ 在区间 } (a, b) \text{ 是空闲的}\}$;
- $A' = \{i \mid i \in A, p_{j(i)} \leq \frac{(\beta-1)R}{\beta}\}$;
- $j(i_0) = \max\{j(i) \mid i \in A'\}$;
- $B = \{i \mid i \in A', j(i) < j(i_0)\} = A' \setminus \{j(i_0)\}$.

引理 4.2 集合 B 满足 $|B| \leq \frac{17(m+1)}{30}$.

证明 首先证明, 若 $B \neq \emptyset$, 则工件 $J_{j(i_0)}$ 是在步骤 5° 被安排加工的, 原因是步骤 4° 的条件 (c) 不满足. 工件 $J_{j(i_0)}$ 在步骤 5° 被安排加工且 $r_{j(i_0)} > a \geq H_{i_0}^{j(i_0)}$ 是显然的, 因为空闲时间间隔 (a, b) 是由工件 $J_{j(i_0)}$ 产生的. 对于任意的 $i \in B$, 由集合 B 及区间 (a, b) 的定义可知 $j(i_0) > j(i)$ 成立, 所以有

$$H_{i_0}^{j(i_0)} \leq a < b \leq r_{j(i)} < r_{j(i)} + p_{j(i)} = H_i^{j(i)+1} \leq H_i^{j(i_0)}.$$

也就是说, 由 MLS 算法步骤 4° 知 $H_i^{j(i_0)} > r_{j(i_0)}$ 一定成立. 因此工件 $J_{j(i_0)}$ 满足步骤 4° 的条件 (a) 和 (b). 至于条件 (d), 它直接来自于以下不等式:

$$r_{j(i_0)} \geq b > a \geq \frac{R}{\beta} \geq \frac{R_{j(i_0)}}{\beta}.$$

所以, 工件 $J_{j(i_0)}$ 是因为不满足步骤 4° 的条件 (c) 而在步骤 5° 被安排加工. 因此, 对于任意的 $i \in B$, 都有

$$p_{j(i_0)} + H_i^{j(i_0)} > \alpha R_{j(i_0)}.$$

对于任意的 $i \in B$, 令 $J_{\bar{j}(i)}$ 是机器 M_i 上最后空闲工件. 由上述讨论及 MLS 算法知

$$\begin{aligned} H_i^{j(i_0)} - r_{\bar{j}(i)} &> \alpha R^{j(i_0)} - p_{j(i_0)} - r_{\bar{j}(i)} \\ &> \alpha r_{\bar{j}(i)} - p_{j(i_0)} - r_{\bar{j}(i)} \\ &\geq \alpha r_{\bar{j}(i)} - \frac{(\beta-1)R}{\beta} - r_{\bar{j}(i)} \\ &\geq (\alpha-1)a - \frac{(\beta-1)R}{\beta} \\ &\geq \frac{(\alpha-\beta)R}{\beta}. \end{aligned}$$

令

$$\begin{aligned} S_i &= \{j \mid J_j \text{ 在机器 } M_i \text{ 上的时间间隔 } [r_{\bar{j}(i)}, H_i^{j(i_0)}] \text{ 内加工}\}, \quad i \in B, \\ S &= \bigcup_{i \in B} S_i, \end{aligned}$$

则有

$$\sum_{j \in S} p_j = \sum_{i \in B} \sum_{j \in S_i} p_j = \sum_{i \in B} (H_i^{j(i_0)} - r_{\bar{j}(i)}) > \frac{|B|(\alpha-\beta)R}{\beta}.$$

接下来进一步证明对于任意的 $j \in S$ 都有 $r_j \geq \frac{R}{\beta^2}$. 设工件 J_j 在机器 M_i 上加工. 若 $j = \bar{j}(i)$, 则 $r_j \geq a \geq \frac{R}{\beta^2}$. 否则, 工件 J_j 安排在工件 $J_{\bar{j}(i)}$ 之后且在工件 $J_{j(i_0)}$ 之前. 所以, 工件 J_j 在步骤 4° 必须被安排加工. 由 MLS 算法步骤 4° 的条件 (d), 有下面的不等式成立:

$$r_j \geq \frac{R_j}{\beta} \geq \frac{r_{j(i)}}{\beta} > \frac{a}{\beta} \geq \frac{R}{\beta^2}.$$

所以

$$R \geq \frac{R}{\beta^2} + \frac{\sum_{j \in S} p_j}{s+m-1} > \frac{R}{\beta^2} + \frac{|B|(\alpha-\beta)R}{(s+m-1)\beta} \geq \frac{R}{\beta^2} + \frac{10|B|(\alpha-\beta)R}{11(m+1)\beta},$$

其中最后的不等式由 $\lceil s \rceil \leq \frac{m+1}{10}$ 得到. 因此, 由上面的不等式及命题 4.2 的第一个方程 $\frac{1}{\beta^2} + \frac{17(\alpha-\beta)}{33\beta} = 1$ 有 $|B| \leq \frac{17(m+1)}{30}$. 引理证毕. $\square$

引理 4.3 集合 A 满足 $|A| \leq |B| + \lceil s \rceil \leq \frac{2(m+1)}{3}$.

证明 首先证明 $|A| \leq |B| + \lceil s \rceil$. 若结论不成立, 令 $C = \{i \mid i \in A, p_{j(i)} > \frac{(\beta-1)R}{\beta}\}$, 则由 $A = B \cup C \cup \{i_0\}$ 有 $|C| \geq \lceil s \rceil$. 因为对于任意的 $i \in A$ 都有 $r_{j(i)} > a$, 所以

$$R > a + p_{\max\{a, \lceil s \rceil\}} \geq a + \frac{(\beta-1)R}{\beta} \geq \frac{R}{\beta} + \frac{(\beta-1)R}{\beta} = R.$$

显然矛盾. 故第一个不等式已经证明. 至于第二个不等式, 可由引理 4.2 的结论及假设 $\lceil s \rceil \leq \frac{m+1}{10}$ 直接得到. $\square$

接下来, 用 u_i 表示 MLS 算法下机器 M_i ($i = 1, 2, \dots, m$) 上的总空闲时间. 由集合 A 的定义可知, 对于区间 $[\frac{R}{\beta}, R]$ 中的任意时间, 在机器 $M_1, M_2, \dots, M_{m-1}$ 中最多有 $|A|$ 台机器同时处于空闲状态. 故有下面的不等式成立:

$$\sum_{i=1}^{m-1} u_i \leq \frac{(m-1)R}{\beta} + |A| \left(R - \frac{R}{\beta} \right) \leq \frac{(m-1-|A|+|A|\beta)C_{\max}^{\text{OPT}}(L)}{\beta}. \quad (4.2)$$

引理 4.4 若工件 J_n 在步骤 5° 安排加工, 则下面的不等式成立:

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{s(m-1)}{s+m-1}.$$

证明 由 MLS 算法及 (4.2) 有

$$\begin{aligned} \frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} &\leq \frac{\sum_{i=1}^{m-1}(H_i + p_n) + (sH_m + p_n)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= \frac{\sum_{j=1}^n p_j + \sum_{i=1}^{m-1} u_i + su_m + (m-1)p_n}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{s(m-1)}{s+m-1}. \end{aligned}$$

证毕. □

引理 4.5 若 J_n 在步骤 5° 安排加工, 则下面的不等式成立:

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}.$$

证明 与定理 3.1 的证明类似. 若不存在 MLS 算法下在机器 M_m 上加工但最优算法下不在机器 M_m 上加工的工件, 即 MLS 算法下在机器 M_m 上加工的所有工件在最优算法下同样在机器 M_m 上加工, 则 $H_m \leq C_{\max}^{\text{OPT}}(L)$, 且有

$$\begin{aligned} C_{\max}^{\text{MLS}}(L) &\leq H_m + \frac{p_n}{s} \leq 2C_{\max}^{\text{OPT}}(L) \\ &\leq \left(1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L). \end{aligned}$$

若存在这样的工件, MLS 算法下在机器 M_m 上加工但最优算法下不在机器 M_m 上加工, 则令其中最后一个工件 J' 的长度为 t' . 用 $\text{SUC}(J')$ 表示 MLS 算法下机器 M_m 上在 J' 之后加工的所有工件, 这些工件的总长度也用它表示. 在最优算法下 $\text{SUC}(J')$ 中的工件也安排在机器 M_m 上加工. 显然, $\text{SUC}(J') \leq sC_{\max}^{\text{OPT}}(L)$ 及 $t' \leq C_{\max}^{\text{OPT}}(L)$. 现从以下两种情形进行讨论.

情形 1 MLS 算法下机器 M_m 上在 J' 之后有空闲时间间隔. 此时用 $[a, b]$ 表示 MLS 算法下机器 M_m 上的最后空闲时间.

若 $b > \frac{H_m + \frac{p_n}{s}}{2}$, 则

$$\begin{aligned} C_{\max}^{\text{MLS}}(L) &\leq H_m + \frac{p_n}{s} \leq 2b \leq 2C_{\max}^{\text{OPT}}(L) \\ &\leq \left(1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L). \end{aligned}$$

若 $b \leq \frac{H_m + \frac{p_n}{s}}{2}$, 则 $\text{SUC}(J') \geq sH_m - sb \geq \frac{sH_m - p_n}{2}$. 所以

$$C_{\max}^{\text{MLS}}(L) \leq H_m + \frac{p_n}{s} \leq \frac{2\text{SUC}(J') + p_n}{s} + \frac{p_n}{s} = \frac{2(\text{SUC}(J') + p_n)}{s}.$$

在最优算法下, 若 J_n 是安排在机器 M_m 上加工的, 则 $\text{SUC}(J') + p_n \leq sC_{\max}^{\text{OPT}}(L)$. 所以有

$$\begin{aligned} C_{\max}^{\text{MLS}}(L) &\leq \frac{2(\text{SUC}(J') + p_n)}{s} \leq 2C_{\max}^{\text{OPT}}(L) \\ &\leq \left(1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L). \end{aligned}$$

若 J_n 在最优算法下不是安排在机器 M_m 上加工的, 则 $\text{SUC}(J') \leq sC_{\max}^{\text{OPT}}(L)$ 且 $p_n \leq C_{\max}^{\text{OPT}}(L)$. 故有

$$\begin{aligned} C_{\max}^{\text{MLS}}(L) &\leq \frac{2(\text{SUC}(J') + p_n)}{s} \leq \left(2 + \frac{1}{s}\right) C_{\max}^{\text{OPT}}(L) \\ &\leq \left(1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}\right) C_{\max}^{\text{OPT}}(L). \end{aligned}$$

情形 2 MLS 算法下机器 M_m 上在 J' 之后没有空闲时间间隔.

与定理 3.1 的证明类似, 可以得到如下结论:

- $C_{\max}^{\text{MLS}}(L) \leq H_i + 2C_{\max}^{\text{OPT}}(L)$, $i = 1, 2, \dots, m-1$;
- 对于机器 M_m , $sC_{\max}^{\text{MLS}}(L) \leq sH_m + p_n$.

故有

$$\begin{aligned} \frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} &= \frac{sC_{\max}^{\text{MLS}}(L) + (m-1)C_{\max}^{\text{MLS}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq \frac{(sH_m + p_n) + \sum_{i=1}^{m-1} (H_i + 2C_{\max}^{\text{OPT}}(L))}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= \frac{(sH_m + p_n) + \sum_{i=1}^{m-1} H_i + 2(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &= \frac{\sum_{j=1}^n p_j + \sum_{i=1}^{m-1} u_i + su_m + 2(m-1)C_{\max}^{\text{OPT}}(L)}{(s+m-1)C_{\max}^{\text{OPT}}(L)} \\ &\leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}. \end{aligned}$$

证毕. □

引理 4.6 若工件 J_n 在步骤 5° 安排加工, 则有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 1 + \frac{m-1-|A|+|A|\beta}{(m+1)\beta} + \frac{2m}{m+1}.$$

证明 由引理 4.4 有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{s(m-1)}{s+m-1}.$$

易知上述不等式的右边部分是关于 s 的增函数. 由引理 4.5 有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq 1 + \frac{m-1-|A|+|A|\beta}{(s+m-1)\beta} + \frac{s}{s+m-1} + \frac{2(m-1)}{s+m-1}.$$

易知上述不等式的右边部分是关于 s 的减函数. 令 $s = 2$, 至此引理 4.6 得证. □

引理 4.7 若工件 J_n 在步骤 5° 安排加工, 则 $\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \alpha$.

证明 由引理 4.3–4.6, 有

$$\begin{aligned} \frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} &\leq 1 + \frac{m-1-|A|+|A|\beta}{(m+1)\beta} + \frac{2m}{m+1} \\ &\leq 3 + \frac{m+1-\frac{2(m+1)}{3} + \frac{2(m+1)}{3}\beta}{(m+1)\beta} \\ &= \frac{1}{3\beta} + \frac{11}{3} \\ &= \alpha. \end{aligned}$$

最后一个等式由命题 4.2 的第二个方程得到. □

引理 4.8 若工件 J_n 在步骤 4° 安排加工, 则 $\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \alpha$.

证明 由 MLS 算法, 有

$$\frac{C_{\max}^{\text{MLS}}(L)}{C_{\max}^{\text{OPT}}(L)} \leq \frac{H_{k+1} + p_n}{R} \leq \alpha.$$

证毕. □

由引理 4.7 和 4.8 的结论可直接得到定理 4.1.

至此, (4.1) 已被证明完毕.

5 结论

本文在综述了在线排序问题研究进展的基础上, 先讨论了速度分别为 $s_1 = s_2 = \cdots = s_{m-1} = 1$, $s_m = s > 1$ 的 m 台同类平行机下的工件有任意到达时间的 LS 算法的性能, 得到了机器数为 2 时的竞争比为 $\frac{3+\sqrt{5}}{2}$, 超过 2 台机器时的竞争比为 $4 - \frac{4}{m+1}$. 然后设计了一个比 LS 算法更好的在线算法—MLS 算法, 并证明了其竞争比不超过 3.8626.

致谢 由衷感谢审稿人对本文提出宝贵的修改意见.

参考文献

- 1 Albers S. Better bounds for online scheduling. *SIAM J Comput*, 1999, 29: 459–473
- 2 Aspnes J, Azar Y, Fiat A, et al. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *J ACM*, 1997, 44: 486–504
- 3 Bartal Y, Fiat A, Karloff H, et al. New algorithms for an ancient scheduling problem. *J Comput System Sci*, 1995, 51: 359–366
- 4 Berman P, Charikar M, Karpinski M. On-line load balancing for related machines. *J Algorithms*, 2000, 35: 108–121
- 5 Cai S Y, Yang Q F. Online scheduling on three uniform machines. *Discrete Appl Math*, 2012, 160: 291–302
- 6 Chen B, Van Vliet A, Woeginger G J. New lower and upper bounds for on-line scheduling. *Oper Res Lett*, 1994, 16: 221–230
- 7 Cheng T C E, Ng C T, Kotov V. A new algorithm for online uniform-machine scheduling to minimize the makespan. *Inform Process Lett*, 2006, 99: 102–105
- 8 Cho Y, Sahni S. Bounds for list schedules on uniform processors. *SIAM J Comput*, 1980, 9: 91–103
- 9 Dolgui A, Kotov V, Nekrashevich A, et al. General parametric scheme for the online uniform machine scheduling problem with two different speeds. *Inform Process Lett*, 2018, 134: 18–23
- 10 Ebenlendr T, Sgall J. A lower bound on deterministic online algorithms for scheduling on related machines without preemption. *Theory Comput Syst*, 2015, 56: 73–81
- 11 Epstein L, Noga J, Seiden S S, et al. Randomized on-line scheduling on two uniform machines. *J Sched*, 2001, 4: 71–92
- 12 Fleischer R, Wahl M. On-line scheduling revisited. *J Sched*, 2000, 3: 343–353
- 13 Galambos G, Woeginger G J. An on-line scheduling heuristic with better worst-case ratio than Graham's list scheduling. *SIAM J Comput*, 1993, 22: 349–355
- 14 Graham R L. Bounds on multiprocessing timing anomalies. *SIAM J Appl Math*, 1969, 17: 416–429
- 15 Han F, Tan Z, Yang Y. On the optimality of list scheduling for online uniform machines scheduling. *Optim Lett*, 2012, 6: 1551–1571
- 16 Jež L, Schwartz J, Sgall J, et al. Lower bounds for online makespan minimization on a small number of related machines. *J Sched*, 2013, 16: 539–547
- 17 Karger D R, Phillips S J, Torng E. A better algorithm for an ancient scheduling problem. *J Algorithms*, 1996, 20: 400–430
- 18 Li R, Huang H C. On-line scheduling for jobs with arbitrary release times. *Computing*, 2004, 73: 79–97
- 19 Li R, Huang H C. Improved algorithm for a generalized on-line scheduling problem on identical machines. *European J Oper Res*, 2007, 176: 643–652

- 20 Li R, Shi L. An on-line algorithm for some uniform processor scheduling. SIAM J Comput, 1998, 27: 414–422
- 21 Musitelli A, Nicoletti J M. Competitive ratio of list scheduling on uniform machines and randomized heuristics. J Sched, 2011, 14: 89–101
- 22 Tan Z, Li R. Pseudo lower bounds for online parallel machine scheduling. Oper Res Lett, 2015, 43: 489–494

Online scheduling for jobs with arbitrary release times on uniform machines

Xiayan Cheng, Congcong Zhao, Lina Ma & Rongheng Li

Abstract In this paper, we study the problem of online scheduling on uniform machines, in which the jobs have arbitrary release times and the objective function is to minimize the maximum completion time. The speeds of the machines are 1 for the first $m - 1$ machines, but the last one's is s ($s > 1$). The performance of the LS (list scheduling) algorithm is analyzed. When the numbers of machines are $m = 2$ and $m \geq 3$, it is presented that the competitive ratios of the LS algorithm are $\frac{3+\sqrt{5}}{2}$ and $4 - \frac{4}{m+1}$, respectively. In general, a better algorithm called MLS (modified list scheduling) is also designed for the problem and it is shown that the competitive ratio of the MLS is no more than 3.8626.

Keywords online scheduling, uniform machine, release time, algorithm, competitive ratio

MSC(2020) 90B35, 90C27

doi: 10.1360/SSM-2024-0095