

# 无延迟加工约束下在线LPT算法 关于平行机在线排序问题 的性能

李文杰

洛阳师范学院, 数学科学学院, 河南 洛阳

收稿日期: 2022年2月11日; 录用日期: 2022年3月7日; 发布日期: 2022年3月14日

---

## 摘 要

本文在无延迟加工约束下研究最小化最大完工时间 $m$ 台平行机在线排序问题。这里的“无延迟加工”是指当工件到达时, 如果有机器空闲则必须选择工件加工, 即工件不能被延迟加工。当 $m \geq 2$ 时, 证明无延迟加工约束下在线LPT算法是 $3/2$ -竞争的最好可能在线算法。如果所有工件都具有友好到达时间, 首先给出无延迟加工约束下排序问题的下界分别为 $5/4$  (当 $m = 2$ 时)和 $4/3$  (当 $m \geq 3$ 时), 其次证明在线LPT算法是 $5/4$ -竞争的最好可能在线算法对两台平行机情形。

## 关键词

在线排序, 在线算法, 无延迟加工, 最大完工时间

---

# LPT Online Strategy for Parallel-Machine Scheduling of Jobs with Non-Delayed Processing Constraint

Wenjie Li

School of Mathematical Sciences, Luoyang Normal University, Luoyang Henan

Received: Feb. 11<sup>th</sup>, 2022; accepted: Mar. 7<sup>th</sup>, 2022; published: Mar. 14<sup>th</sup>, 2022

---

## Abstract

This paper studies the  $m$  parallel-machines online schedule under the non-delayed processing

constraint to minimize the maximum completion times of jobs. Here “non-delayed processing” means that the available jobs cannot be delayed for processing when some machine is idle. When  $m \geq 2$ , this paper shows that the online LPT algorithm is a  $3/2$ -competitive best possible online algorithm. For the restricted version in which the jobs have kind release times, this paper first shows that the online LPT algorithm is a best possible online algorithm with a competitive ratio of  $5/4$  for the case  $m = 2$ , and then presents a lower bound of  $4/3$  for the case  $m \geq 3$ .

## Keywords

Online Scheduling, Online Algorithm, Non-Delayed Processing, Maximum Completion Time

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

本在线排序是现代排序中的核心研究方向,广泛地应用于生产管理、物流运输、商品销售、网络服务等很多领域。时间在线排序是三类在线排序模型(时间在线、列表在线、不可测在线)中的一类重要的排序模型,也是现代排序领域中发展最为迅速的研究方向之一。时间在线排序中工件的一切信息(例如,到达时间、加工时间等)只有在其到达时才能被知晓。

通常用竞争比来衡量最小化目标函数在线排序问题对应在线算法的性能。我们称算法  $A$  是  $\rho_A$ -竞争的,如果满足  $\rho_A = \sup_{I \in \mathcal{I}} \{A(I)/OPT(I) : OPT(I) > 0\}$ , 其中  $I$  是排序问题的任意一个实例,  $A(I)$  表示算法  $A$  关于  $I$  产生的目标函数值,  $OPT(I)$  表示离线最优算法关于  $I$  产生的目标函数值。特别的,如果一个排序问题在线算法的竞争比等于排序问题的竞争比下界,则称该算法是排序问题的一个最好可能在线算法。

工件无延迟加工在线排序是 Li 和 Yuan 在文献[1]中首先研究的一类新的在线排序模型。所谓无延迟加工是指已到达的工件不能在机器出现空闲的时刻被推迟加工。换言之,机器只有在无工件可选择加工的情形下产生空闲时间段。此约束在文献[1]中被简记为 NDP 约束。NDP 约束下的工件加工方式来源于实际中的生产加工模式。例如,在机械制造行业,很多大型生产加工设备每次启动都要消耗大量的资源(如人力资源、电力资源、油气资源等),为了节约生产成本加工设备一旦开工通常会采取无空闲的加工方式去处理已到达工件,则不会强制推迟工件加工。

本文在 NDP 约束下研究的具体排序问题描述如下: 设  $I$  是一个工件集合(又称实例)包含  $n$  个工件  $J_1, J_2, \dots, J_n$ 。对  $I$  中的每一个工件  $J_j$ , 它具有一个到达时间(也称释放时间)  $r_j \geq 0$ , 一个加工时间(也称工件长度)  $p_j > 0$ 。有  $m$  台平行机(即每台机器的加工速度相同)  $M_1, M_2, \dots, M_m$ ,  $m \geq 2$ 。排序目标: 确定一个工件不能被延迟加工的可行排序最小化所有工件的最大完工时间( $C_{\max}$ ), 其中  $C_{\max} = \max\{C_j : j=1, 2, \dots, n\}$ ,  $C_j$  为工件  $J_j$  的完工时间。工件具有友好到达时间在线排序是 Li 和 Yuan 在文献[2]中首先研究的一类半在线排序模型(文献中简记为 KRT 模型), 其中“KRT”是指在线环境中当所有机器都在忙碌时, 则不会有新工件达到。为了便于讨论, 现将本文在 NDP 约束下研究的两个原问题(最小化最大完工时间  $m$  台平行机在线排序问题和 KRT 模型中的最小化最大完工时间 2 台平行机在线排序问题)分别简记为(P1)和(P2), 将 NDP 约束下的排序问题 (P1)和(P2)分别简记为(Q1)和(Q2)。采用 Graham 等人在文献[3]中给出的三参数表示法, 这四个排序问题可分别表示为:

$$\begin{cases} (P1): Pm | \text{online}, r_j | C_{\max}, \\ (Q1): Pm | \text{online}, r_j, \text{NDP} | C_{\max}, \\ (P2): P2 | \text{online}, r_j, \text{KRT} | C_{\max}, \\ (Q2): P2 | \text{online}, r_j, \text{NDP-KRT} | C_{\max}, \end{cases}$$

其中问题(Q1)和(Q2)是本文研究的排序问题, 并且问题(Q2)中的“NDP-KRT”是指排序中每个工件都具有友好到达时间并且不能被延迟加工。关于原问题(P1), Chen和Vestjens在文献[4]给出该问题的下界是1.347并证明在线LPT算法的竞争比为1.5。对两台平行机情形, Noga和Seiden在文献[5]中设计出一个1.382-竞争的最好可能在线算法。关于原问题(P2), Li和Yuan在文献[2]中证明在线LPT算法是1.25-竞争的最好可能在线算法。

本文将在第二节给出问题(Q1)和(Q2)的下界, 在第三节中分析在线LPT算法的竞争比。

## 2. 问题的下界

本节首先给出问题(Q1)的下界为 $3/2$ , 其次给出问题  $Pm | \text{online}, r_j, \text{NDP-KRT} | C_{\max}$  的下界分别为 $5/4$  (当 $m=2$ 时)和 $4/3$  (当 $m \geq 3$ 时, 即为问题(Q2))。这里用  $C_{\text{on}}(I)$  和  $C_{\text{opt}}(I)$  分别表示任意一个在线算法和离线最优算法关于任意一个工件实例 $I$ 产生的目标函数值。

**定理2.1** 对于问题  $Pm | \text{online}, r_j, \text{NDP} | C_{\max}$ , 不存在竞争比小于 $3/2$ 的在线算法。

证明考察关于问题(Q1)的任意一个在线算法 $A$ 。设 $\varepsilon$ 是一个充分小的正实数并满足 $0 < \varepsilon < 0.1$ 。下面将采用对手法构造出至多包含 $2m+2$ 个工件的实例 $I$ 来完成定理证明。

在0时刻有 $m+1$ 个工件 $J_1, J_2, \dots, J_m, J_x$ 到达其加工时间分别为 $p_1 = p_2 = \dots = p_m = \varepsilon$ 和 $p_x = 2\varepsilon$ 。讨论以下两种情形:

情形1 算法 $A$ 在0时刻加工 $J_1, J_2, \dots, J_m$ , 则之后不会再有新工件到达。由NDP约束可知 $J_x$ 只能在 $J_1, J_2, \dots, J_m$ 完工后被立刻安排加工, 即 $S_x = \varepsilon$ , 进而可得 $C_{\text{on}}(I) = 3\varepsilon$ 。注意到, 离线最优算法则会将 $J_x$ 单独安排在一台机器上并且在0时刻开工, 于是可得 $C_{\text{opt}}(I) = 2\varepsilon$ 。因此, 可得

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{3}{2}.$$

情形2 算法 $A$ 在0时刻加工 $J_x$ , 即 $S_x = 0$ , 则将会有 $m$ 个加工时间均等于1的新工件 $J'_1, J'_2, \dots, J'_m$ 在 $2\varepsilon$ 时刻到达和一个加工时间等于2的新工件 $J'_x$ 在 $3\varepsilon$ 时刻到达。可知此情形下工件实例 $I = \{J_1, J_2, \dots, J_m, J_x, J'_1, J'_2, \dots, J'_m, J'_x\}$ 。由NDP约束可知在 $2\varepsilon$ 时刻工件 $J_1, J_2, \dots, J_m, J_x$ 均已完工,  $J'_1, J'_2, \dots, J'_m$ 必须在 $2\varepsilon$ 时刻被分别安排在 $m$ 台机器上开工。又由于 $r'_x > r'_j, j=1, 2, \dots, m$ , 工件 $J'_x$ 只能在 $C'_j = 1 + 2\varepsilon$ 时刻开工。于是可得 $C_{\text{on}}(I) = C'_j + p'_x = 3 + 2\varepsilon$ 。然而在离线最优排序中, 可以在0时刻安排 $J_1, J_2, \dots, J_m$ 在 $m$ 台机器上开工,  $\varepsilon$ 时刻安排 $J_x$ 在 $M_1$ 上开工,  $2\varepsilon$ 时刻安排 $J'_1, J'_2, \dots, J'_m$ 中的 $m-1$ 个工件分别在 $M_2, M_3, \dots, M_m$ 上开工,  $3\varepsilon$ 时刻安排 $J'_x$ 在 $M_1$ 上开工(注意到, 此时 $J_x$ 已完工且 $J'_x$ 已到达),  $1 + 2\varepsilon$ 时刻安排最后一个 $J'_1, J'_2, \dots, J'_m$ 中的工件在 $M_2, M_3, \dots, M_m$ 上的某一台机器上开工。所以可得 $C_{\text{opt}}(I) = p_j + p_x + p'_x = 2 + 3\varepsilon$ 。因此可推得

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{3 + 2\varepsilon}{2 + 3\varepsilon} \rightarrow \frac{3}{2}, \text{ 当 } \varepsilon \rightarrow 0 \text{ 时。}$$

综上, 定理2.1证毕。□

**定理2.2** 对于问题  $Pm | \text{online}, r_j, \text{NDP-KRT} | C_{\max}$ , 当 $m=2$ 时不存在竞争比小于 $5/4$ 的在线算法; 当

$m \geq 3$  时, 不存在竞争比小于  $4/3$  的在线算法。

证明考察关于该问题的任意一个在线算法  $A$ 。设  $\varepsilon$  是一个充分小的正实数并满足  $0 < \varepsilon < 0.1$ 。

当  $m = 2$  时, 用对手法构造的实例  $I$  至多包含 4 个工件。在 0 时刻, 一个长度为 1 的工件  $J_1$  到达。由 NDP 约束可知  $J_1$  在 0 时刻开工。不妨假设  $J_1$  被算法  $A$  安排在  $M_1$  上开工。在  $\varepsilon$  时刻(注意到, 此时刻  $M_2$  空闲符合 KRT 模型要求), 有两个工件  $J_2$  和  $J_3$  到达并满足  $p_2 = 2, p_3 = 3$ 。如果算法  $A$  在  $\varepsilon$  时刻选择在  $M_2$  加工  $J_2$ , 则之后不会再有新工件到达。由 NDP 约束可知  $J_3$  只能在  $J_1$  完工之后被安排在  $M_1$  上开工。于是可得  $C_{\text{on}}(I) = C_3 = p_1 + p_3 = 4$ 。而离线最优算法则会在  $\varepsilon$  时刻先安排  $J_3$  在  $M_2$  上加工, 后安排  $J_2$  在  $M_1$  上 1 时刻开工, 即有  $C_{\text{opt}}(I) = r_3 + p_3 = 3 + \varepsilon$ 。因此可得,

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{4}{3 + \varepsilon} > \frac{5}{4}.$$

如果算法  $A$  在  $\varepsilon$  时刻选择在  $M_2$  加工  $J_3$ , 则一个长度为 2 的新工件  $J_4$  将会在 1 时刻到达(此时刻  $J_1$  完工  $M_1$  空闲符合 KRT 模型要求)。此情形由 NDP 约束和  $3 < 3 + \varepsilon$  可知, 算法  $A$  只能安排  $J_2$  和  $J_4$  在  $M_1$  上加工。于是可得  $C_{\text{on}}(I) = p_1 + p_2 + p_4 = 5$ 。而在离线最优排序中则可以依次安排  $J_1$  和  $J_3$  在  $M_1$  上加工, 安排  $J_2$  和  $J_4$  在  $M_2$  上加工, 进而有  $C_{\text{opt}}(I) = r_2 + p_2 + p_4 = 4 + \varepsilon$ 。因此可得,

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{5}{4 + \varepsilon} \rightarrow \frac{5}{4}, \text{ 当 } \varepsilon \rightarrow 0 \text{ 时}.$$

当  $m \geq 3$  时, 用对手法构造的实例  $I$  至多包含  $2m$  个工件。在 0 时刻有  $m$  个长度为 1 工件  $J_1, J_2, \dots, J_m$  和一个长度为 2 的工件  $J_x$  到达。如果算法  $A$  在 0 时刻不选择加工  $J_x$ , 则之后不会再有新工件到达。此情形可得,

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{3}{2} > \frac{4}{3}.$$

如果算法  $A$  在 0 时刻选择加工  $J_x$  (不妨假设在  $M_1$  上加工  $J_x$ ), 则有  $m-1$  个长度为 2 的新工件  $J'_1, J'_2, \dots, J'_{m-1}$  将会在  $1 + \varepsilon$  时刻到达(注意到, 此时刻有  $m-1$  台机器空闲符合 KRT 模型要求)。此情形  $I = \{J_1, J_2, \dots, J_m, J_x, J'_1, J'_2, \dots, J'_{m-1}\}$ 。这里不妨假设在 0 时刻算法  $A$  依次安排  $J_1, J_2, \dots, J_{m-1}$  在  $M_2, M_3, \dots, M_m$  上开工并安排  $J_m$  在  $M_2$  上继  $J_1$  之后开工。由 NDP 约束可知  $J'_1, J'_2, \dots, J'_{m-1}$  中至少有一个工件被安排在  $M_1$  上继  $J_x$  之后或  $M_2$  上继  $J_m$  之后开工, 进而可得  $C_{\text{on}}(I) = 2 + 2 = 4$ 。离线最优算法则将会在 0 时刻加工  $J_1, J_2, \dots, J_m$ , 在 1 时刻和  $1 + \varepsilon$  时刻依次加工  $J_x, J'_1, J'_2, \dots, J'_{m-1}$ , 进而可得  $C_{\text{opt}}(I) = r'_j + p'_j = 3 + \varepsilon, j = 1, 2, \dots, m-1$ 。于是可得,

$$\frac{C_{\text{on}}(I)}{C_{\text{opt}}(I)} = \frac{4}{3 + \varepsilon} \rightarrow \frac{4}{3}, \text{ 当 } \varepsilon \rightarrow 0 \text{ 时}.$$

综上, 定理 2.2 证毕。 □

### 3. 论在线 LPT 算法及其竞争比分析

本节将证明在线 LPT 算法是问题 (Q1) 和 (Q2) 的最好可能在线算法。在线 LPT 算法是 Chen 和 Vestjens 在文献 [4] 中研究问题 (P1) 时给出的, 其执行策略可叙述为“在任何时刻当有机器出现空闲, 则从当前已经到达但还未被加工的工件中挑选加工时间最大的工件安排在该台空闲机器上加工”。显然, 在线 LPT 算法的执行策略满足 NDP 约束条件, 并且该算法也是文献 [2] 中解决问题 (P2) 的在线算法。因此, 本文也将此算法作为解决问题 (Q1) 和 (Q2) 的在线算法。文献 [4] 和 [2] 中的两个主要结论将作为本文的一个重要引理

并描述如下:

**引理3.1** 对于问题(P1)和(P2), 在线LPT算法的竞争比分别为是3/2和5/4。 □

下面分析在线LPT算法关于问题(Q1)和(Q2)的性能. 设 $I$ 是任意一个工件实例. 对于 $I$ , 令 $\sigma_1, \sigma'_1, \sigma_2, \sigma'_2$ 分别表示在线LPT算法关于问题(Q1), (P1), (Q2), (P2)生成的可行排序, 令 $\pi_1, \pi'_1, \pi_2, \pi'_2$ 分别表示离线最优算法关于问题(Q1), (P1), (Q2), (P2)生成的最优排序. 由LPT算法的执行策略和NDP约束的定义可知,  $\sigma_1 = \sigma'_1$  和  $\sigma_2 = \sigma'_2$ . 又知 $\pi_1$ 和 $\pi_2$ 可以分别看作问题(P1)和(P2)的一个可行排序(原因是 $\pi_1$ 和 $\pi_2$ 中的工件加工受NDP约束, 而问题(P1)和(P2)中的工件加工无限制). 令 $\sigma_1(I), \sigma'_1(I), \sigma_2(I), \sigma'_2(I)$ 分别表示在线LPT算法关于 $I$ 对应上述四个排序问题产生的目标函数值, 令 $\pi_1(I), \pi'_1(I), \pi_2(I), \pi'_2(I)$ 分别表示离线最优算法关于 $I$ 对应上述四个排序问题产生的目标函数值, 则有 $\sigma_1(I) = \sigma'_1(I)$ ,

$$\frac{\sigma_1(I)}{\pi_1(I)} = \frac{\sigma'_1(I)}{\pi_1(I)} \leq \frac{\sigma'_1(I)}{\pi'_1(I)} \text{ 和 } \frac{\sigma_2(I)}{\pi_2(I)} = \frac{\sigma'_2(I)}{\pi_2(I)} \leq \frac{\sigma'_2(I)}{\pi'_2(I)}.$$

由实例 $I$ 的任意性及引理3.1可以推得在线LPT算法关于问题(Q1)的竞争比是3/2, 关于问题(Q2)的竞争比是5/4. 结合定理2.1和2.2得到本文的主要结论.

**定理3.2** 对于问题  $Pm | \text{online}, r_j, \text{NDP} | C_{\max}$  和  $Pm | \text{online}, r_j, \text{NDP-KRT} | C_{\max}$ , 在线LPT算法是最好可能在线算法其竞争比分别为3/2和5/4。 □

## 4. 结论与展望

本文讨论了工件无延迟加工约束下在线 LPT 算法关于最小化最大完工时间平行机在线排序问题的性能, 得到了在线 LPT 算法就是问题(Q1)和(Q2)的最好可能在线算法这两个较完整的结论. 但对问题  $Pm | \text{online}, r_j, \text{NDP-KRT} | C_{\max}$ ,  $m \geq 3$  只给出了一个 4/3 的下界, 而关于在线 LPT 算法的性能有待进一步的研究.

## 基金项目

河南省自然科学基金项目(222300420503), 河南省高校重点科研项目(22A110015), 河南省高校青年骨干教师培养计划项目(2019GGJS202, 2018XJGGJS-10)。

## 参考文献

- [1] Li, W.J. and Yuan, J.J. (2021) Single-Machine Online Scheduling of Jobs with Non-Delayed Processing Constraint. *Journal of Combinatorial Optimization*, **41**, 830-843. <https://doi.org/10.1007/s10878-021-00722-4>
- [2] Li, W.J. and Yuan, J.J. (2016) LPT Online Strategy for Parallel-Machine Scheduling with Kind Release Times. *Optimization Letters*, **10**, 159-168. <https://doi.org/10.1007/s11590-015-0862-y>
- [3] Graham, R.L., Lawer, E.L. and Lenstra, J.K. (1979) Optimization and Approximation in Deterministic Sequencing and Scheduling. *Annals of Discrete Mathematics*, **5**, 287-326. [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- [4] Chen, B. and Vestjens, A.P.A. (1997) Scheduling on Identical Machines: How Good Is LPT in an On-Line Setting? *Operations Research Letters*, **21**, 165-169. [https://doi.org/10.1016/S0167-6377\(97\)00040-0](https://doi.org/10.1016/S0167-6377(97)00040-0)
- [5] Noga, J. and Seiden, S.S. (2002) An Optimal Online Algorithm for Scheduling Two Machines with Release Times. *Theoretical Computer Science*, **268**, 133-143. [https://doi.org/10.1016/S0304-3975\(00\)00264-4](https://doi.org/10.1016/S0304-3975(00)00264-4)