Pure Mathematics 理论数学, 2026, 16(5), 175-182
Published Online May 2026 in Hans. https://www.hanspub.org/journal/pm
https://doi.org/10.12677/pm.2026.165141

Hans汉斯

NDP加工模式下两个典型在线算法的性能分析

李文杰[1]，杜　强[2]，孙晓锋[3]

[1]洛阳师范学院数学科学学院，河南　洛阳
[2]洛阳地矿双语学校，河南　洛阳
[3]洛阳师范学院文学院，河南　洛阳

收稿日期：2026年4月7日；录用日期：2026年5月12日；发布日期：2026年5月26日

摘　要

本文首先在NDP加工模式(即只要机器空闲且有工件可选择，则必须立即选择工件加工)下分别研究了最小化总完工时间平行机在线排序问题，并证明SPT是一个无界的在线算法。其次研究了最小化最大加权流程时间在线分批排序问题，对工件加工时间无限制情形，证明不存在常数界的在线算法；对等长工件情形，分别证明LSB是1-竞争的最优在线算法(在NDP加工模式下)和2-竞争的最好可能在线算法(在一般加工模式下)。

关键词

在线排序，在线算法，NDP加工模式，加权流程时间

Performance Analysis of Two Typical Online Algorithms under the NDP Processing Mode

Wenjie Li[1], Qiang Du[2], Xiaofeng Sun[3]

[1]Faculty of Mathematical Sciences, Luoyang Normal University, Luoyang Henan
[2]Luoyang Dikuang Bilingual School, Luoyang Henan
[3]College of Chinese Language and Literature, Luoyang Normal University, Luoyang Henan

Received: April 7, 2026; accepted: May 12, 2026; published: May 26, 2026

Abstract

This paper first studies the online scheduling problem on parallel machines to minimize the total completion time of jobs under the NDP processing mode (*i.e.*, the available jobs cannot be delayed for processing when some machine is idle, it must be selected for processing immediately), and then prove that SPT is an unbounded online algorithm. Second, we studied the online batch- scheduling

文章引用: 李文杰, 杜强, 孙晓锋. NDP 加工模式下两个典型在线算法的性能分析[J]. 理论数学, 2026, 16(5): 175-182.
DOI: 10.12677/pm.2026.165141

problem to minimize the maximum weighted flow times of jobs. For the case that the job processing times are unrestricted, we show that there is no online algorithm with the constant competitive ratio. For the case of equal-length jobs, we prove that LSB is a 1-competitive optimal algorithm (under the NDP processing mode) and a 2-competitive best possible online algorithm (under the general processing mode), respectively.

Keywords

Online Scheduling, Online Algorithm, NDP Processing Mode, Weighted Flow Time

1. 引言

在线排序是算法与运筹优化领域中发展最为迅速的研究方向之一。它通常要求算法制定者在只掌握部分工件(已经到达的工件)信息或工件的部分信息(如到达时间已知，加工时间未知；加工时间已知，工件权重未知等)的情况下制定排序策略，使得排序目标(一个或多个)达到最优。由于在线排序模式更接近于现实环境，其被广泛地应用于产品加工、货物运输、交通调度、线上服务等众多领域[1]-[4]。

竞争比是衡量在线算法性能的重要手段。一般将其定义为 $\rho_A = \sup_{\forall I}\left\{A(I)/OPT(I): OPT(I)>0\right\}$，这里的 I 表示工件集合，$A(I)$ 表示算法 H 作用在 I 上产生的排序目标值，$OPT(I)$ 表示最优排序下的目标值。如果一个在线算法满足其竞争比刚好等于该排序问题竞争比的下界，则称该算法是解决该排序问题的最好可能在线算法。当 $\rho_A = 1$ 时，A 是一个最优算法。

一般假设工件集合 I 中含有 n 个工件 $J_1, J_2, \cdots, J_n$。每一个工件 J_j 的信息参数包括到达时间 $r_j \geq 0$，加工时间 $p_j \geq 0$，或者权重 $w_j \geq 0$。用 $C_1, C_2, \cdots, C_n$ 分别表示 n 个工件在一个可行排序下的完工时间。用 $\sum C_j$ 和 $WF_{\max} = \max\limits_{1\leq j\leq n}\left\{w_j F_j\right\}$ 分别表示 n 个工件的总完工时间和最大的加权流程时间，其中 J_j 的加权流程时间为 $w_j F_j = w_j\left(C_j - r_j\right)$。NDP 加工模式是 Li 和 Yuan [5]定义的一种机器选择工件加工的模式：如机器空闲且有工件到达，则机器必须立刻选择工件开工。NDP 加工模式源于现实环境中一些产品的生产加工方式。比如机械制造领域，一些超大型生产加工设备每次启动都要消耗大量的能源(如油气、网电等能源)。为了降低能耗或减少污染，这些超大型加工设备一旦启动就会采取连续无延迟的加工方式去处理已到达工件。有关 NDP 加工模式下的在线排序研究可参看近几年的文献[6]-[8]。一般加工模式是指在机器出现空闲的任何时刻，若有工件到达，则机器可以立刻选择工件开工，也可不选择工件开工(即可延迟已到达的工件开工)。本文在一般加工模式(即在工件开工无 NDP 限制)和 NDP 模式下研究的五个具体排序问题用三参数表示法[9]，分别记为：

Q(1,1): $P\left|online, r_j, NDP\right|\sum C_j$;

Q(2,1): $1\left|online, r_j, p-batch, b=\infty\right|WF_{\max}$;

Q(2,2): $1\left|online, r_j, p-batch, b=\infty, NDP\right|WF_{\max}$;

Q(3,1): $P_m\left|online, r_j, p_j=1, p-batch, b=\infty\right|WF_{\max}$;

Q(3,2): $P_m\left|online, r_j, p_j=1, p-batch, b=\infty, NDP\right|WF_{\max}$,

其中 P_m 表示有 m ($m\geq 1$)台平行机器，P 表示平行机的个数大于等于 2，“$p-batch,b$”表示 b 个工件可以放在一起形成一批工件在同一时刻开工且它们有相同的完工时间，“$b=\infty$”表示批容量无限制，“$p_j=1$”表示工件集合中的工件具有相同的加工时间(均为 1)。

最小化总完工时间在线排序问题，在一般加工模式下的两个重要研究结果分别为 Lu 等人[10]给出了一个 2-竞争的最好可能在线算法(D-SPT)对单机情形和 Liu 等人[11]证明算法的竞争比是 2 对多机情形。最小化最大加权流程时间在线排序问题，目前的研究结果较少。Chai 等人[12]研究了工件的权重受限制的单机在线分批排序问题，其研究结果包括：当工件的加工时间均为 1 且批容量有限制情形，分别给出了一个 $\frac{1+\sqrt{4w+1}}{2}$-竞争的最好可能在线算法和 w-竞争的在线算法，对应 $w\in[1,2]$ 情形和 $w\in(2,+\infty)$ 情形，这里的参数 w 等于工件集合里最大权值和最小权值之比。关于最小化最大流程时间在线排序问题相关研究结果可参见文献[13]-[16]。

本文将在第二节证明经典的 SPT 算法是关于问题 Q(1,1)的一个无界算法。在第三节证明问题 Q(2,1)和 Q(2,2)均不存在常数界的在线算法。在第四节证明经典的 LSB 算法是 1-竞争的最优在线算法(对问题 Q(3,2))和 2-竞争的最好可能在线算法(对问题 Q(3,1)的 $m=1$ 情形)。

2. SPT 算法关于问题 Q(1,1)的性能分析

Biskup 在 1999 年的研究[17]将 SPT (Shortest Processing Time)算法(也称 SPT 规则)列为“经典调度规则”。SPT 算法的执行策略：*若机器出现空闲时刻且有工件可选择，则立即选择加工时间最小的工件开工，否则机器保持等待*。这里分别用 $\sum_{spt}C_j$ 和 $\sum_{opt}C_j$ 表示 SPT 算法和离线最优算法 OPT 关于任意一个工件集合生成的排序所对应的目标函数值。

定理 2.1 对于排序问题 Q(1,1)：$P|online,r_j,NDP|\sum C_j$，SPT 算法的竞争比是 ∞。

证明不妨设有 m ($m\geq 2$)台机器，$n>100$。工件集合 I 包含 $2m+n^3+1$ 个工件，其中 m 个工件 $J_1,J_2,\cdots,J_m$ 的到达时间均为 0，加工时间均为 $1+\frac{1}{n^3}$，一个工件 J_0 的到达时间为 0，加工时间为 1；另 m 个大工件 $J_1',J_2',\cdots,J_m'$ 在 1 时刻到达其加工时间均为 M(M 是个非常大的正数)；n^3 个加工时间为 0 的工件 $J_1'',J_2'',\cdots,J_{n^3}''$ 在 $2+\frac{2}{n^3}$ 时刻到达。各类工件的详细数据如下表：

工件	J_0	$J_1,J_2,\cdots,J_m$	$J_1',J_2',\cdots,J_m'$	$J_1'',J_2'',\cdots,J_{n^3}''$
到达时间	0	0	1	$2+2/n^3$
加工时间	1	$1+1/n^3$	M	0

根据 SPT 算法的执行规则，可知在 0 时刻工件 $J_0,J_1,\cdots,J_{m-1}$ 被依次安排在 m 台机器上加工。注意到，在 1 时刻虽然有 m 个大工件到达，但根据 SPT 规则将安排工件 J_m 开工。而这 m 个大工件分别在 $1+\frac{1}{n^3}$ 时刻和 $2+\frac{1}{n^3}$ 时刻被依次安排在 m 台机器上加工。由于 n^3 个 0 工件的到达时间是 $2+\frac{2}{n^3}$，此时大工件正在被加工机器没有空闲，则这 n^3 个 0 工件只能在大工件的最早完工时刻 $M+1+\frac{1}{n^3}$ 之后开始加工。于是可得这 $2m+n^3+1$ 个工件在 SPT 加工顺序下的完工时间和为

$$\sum_{spt} C_j = 1+(m-1)\left(1+\frac{1}{n^3}\right)+2+\frac{1}{n^3}+(m-1)\left(M+1+\frac{1}{n^3}\right)+M+2+\frac{1}{n^3}+n^3\left(M+1+\frac{1}{n^3}\right)$$
$$= (M+1)n^3+m(M+2)+4+\frac{2m}{n^3}.$$

在离线最优序下工件集合 I 中的 $2m+n^3+1$ 个工件的加工顺序是工件 $J_0,J_1,\cdots,J_{m-1}$ 被依次安排在 m 台机器上在 0 时刻开工；在 1 时刻大工件已经到达，于是将一个大工件 J_1' 安排在此时刻开工，将 $J_m,J_2',\cdots,J_{m-1}'$ 安排在 $1+\frac{1}{n^3}$ 时刻开工；注意到 J_m 在 $2+\frac{2}{n^3}$ 时刻完工，而 n^3 个 0 工件此时刚好到达，于是将其安排在 $2+\frac{2}{n^3}$ 时刻开工，J_m' 安排在 0 工件完工之后开工。故而可得最优序下的总完工时间为

$$\sum_{opt} C_j = 3+(m-1)\left(1+\frac{1}{n^3}\right)+2+\frac{2}{n^3}+(m-2)\left(M+1+\frac{1}{n^3}\right)+2M+3+\frac{2}{n^3}+n^3\left(2+\frac{2}{n^3}\right)$$
$$= 2n^3+m(M+2)+5+\frac{2m+1}{n^3}.$$

进而可得

$$\frac{\sum_{spt} C_j}{\sum_{opt} C_j} = \frac{(M+1)n^3+m(M+2)+4+\frac{2m}{n^3}}{2n^3+m(M+2)+5+\frac{2m-1}{n^3}} \xrightarrow{n\to\infty} \frac{M+1}{2} \xrightarrow{M\to\infty} \infty。$$

综上，定理 2.1 得证。 □

3. 问题 Q(2,1)和 Q(2,2)的竞争比下界分析

本节将用对手法构造工件集合并证明问题 Q(2,1)和 Q(2,2)均不存在常数界的在线算法，问题 Q(3,1)不存竞争比小于 2 的在线算法。这里分别用 WF_{on} 和 WF_{opt} 表示在线算法和离线最优算法关于任意一个工件集合生成的排序所对应的目标函数值。

定理 3.1 对于排序问题 Q(2,1)：$1\left|online,r_j,p-batch,b=\infty\right|WF_{\max}$，不存在常数界的在线算法。

证明 不妨设 A 是关于问题 Q(2,1)的任一个在线算法。这里的工件集合至多包含两个工件，$n>100$。

工件 J_1 在 0 时刻到达，，其加工时间 $p_1=1$，权重 $w_1=1$。若 J_1 形成一批工件在算法 A 下的开工时间 $S_1\geq n$，则不再有新工件达到。此情形中，

$$WF_{on}=w_1F_1=w_1(C_1-r_1)=w_1(S_1+p_1-r_1)\geq n+1。$$

在最优序中 J_1 会被安排在 0 时刻开工，进而有 $WF_{opt}=1$。于是可得

$$\rho_A=\frac{WF_{on}}{WF_{on}}\geq n+1\xrightarrow{n\to\infty}\infty。$$

若 J_1 在算法 A 下的开工时间 $S_1<n$，则工件 J_2 在 $S_1+\frac{1}{n^2}$ 时刻到达，其加工时间 $p_2=\frac{1}{n^2}$，权重 $w_2=n^2$。由于此时第一批工件正在加工，J_2 只能单独形成一批在 J_1 完工之后开工，即 $S_2\geq C_1$。此情形中，

$$WF_{on}=\max\{w_1F_1,w_2F_2\}=w_2(C_2-r_2)=w_2(S_2+p_2-r_2)$$
$$\geq w_2(C_1+p_2-r_2)=w_2(S_1+p_1+p_2-r_2)=n^2.$$

而在最优序下，J_2 和 J_1 各形成一批分别在 $S_1+\frac{1}{n^2}$ 和 $S_1+\frac{2}{n^2}$ 时刻开工，进而有 $WF_{opt}=S_1+1+\frac{2}{n^2}$。

由于 $S_1 < n$，于是可得

$$\rho_A = \frac{WF_{on}}{WF_{on}} \geq \frac{n^2}{S_1 + 1 + \frac{1}{n^2}} > \frac{n^2}{n + 1 + \frac{1}{n^2}} \xrightarrow{n \to \infty} \infty \text{。}$$

综上，定理 3.1 得证。 □

定理 3.2 对于排序问题 Q(2,2)：$1\left|online, r_j, p-batch, b=\infty, NDP\right|WF_{\max}$，不存在常数界的在线算法。

证明 不妨设 A 是关于问题 Q(2,2)的任一个在线算法。这里的工件集合至多包含 3 个工件，$n > 100$。

工件 J_1 和 J_2 在 0 时刻到达，其加工时间和权重分别为 $p_1 = 1$，$p_2 = n$，$w_1 = 1$，$w_2 = 0$。由于工件加工受 NDP 约束，在 0 时刻算法 A 必须选择工件批开工。若 J_2 单独一批或 J_1 和 J_2 形成一批在 0 时刻开工，则不在有新工件到达。此情形有 $C_1 \geq n$。进而可得

$$WF_{on} = \max\{w_1 F_1, w_2 F_2\} = w_1 (C_1 - r_1) \geq n \text{。}$$

注意到，最优序下 J_1 单独一批在 0 时刻开工。故有 $WF_{opt} = 1$。于是可得

$$\rho_A = \frac{WF_{on}}{WF_{on}} \geq n \xrightarrow{n \to \infty} \infty \text{。}$$

若 J_1 单独一批在 0 时刻开工，则新工件 J_3 将在 n 时刻到达，其加工时间和权重分别为 $p_3 = \frac{1}{n}$，$w_3 = n^2$。由于 n 时刻 J_2 正在被加工，可知 J_3 只能被安排在 $n+1$ 时刻开工。此情形下有

$$WF_{on} = \max\{w_1 F_1, w_2 F_2, w_3 F_3\} = w_2 (C_3 - r_3) = n^2 \left(n + 1 + \frac{1}{n} - n \right) = n^2 + n \text{。}$$

而最优序下 J_1 和 J_2 成一批在 0 时刻开工，J_3 在 n 时刻开工，进而可得 $WF_{opt} = n$。于是可得

$$\rho_A = \frac{WF_{on}}{WF_{on}} = \frac{n^2 + n}{n} = n + 1 \xrightarrow{n \to \infty} \infty \text{。}$$

综上，定理 3.2 得证。 □

定理 3.3 对于排序问题 Q(3,1)：$P_m\left|online, r_j, p_j = 1, p-batch, b=\infty\right|WF_{\max}$，当 $m = 1$ 时，不存在竞争比小于 2 的在线算法。

证明 不妨设 A 是关于该问题的任一个在线算法。这里的工件集合至多包含 2 个工件，$n > 100$。

工件 J_1 在 0 时刻到达，其加工时间 $p_1 = 1$，权重 $w_1 = 1$。若 J_1 形成一批工件在算法 A 下的开工时间 $S_1 \geq 1$，则不在有新工件达到。此情形中，

$$WF_{on} = w_1 F_1 = w_1 (C_1 - r_1) = w_1 (S_1 + p_1 - r_1) \geq 2 \text{。}$$

在最优序中 J_1 会被安排在 0 时刻开工，进而有 $WF_{opt} = 1$。于是可得

$$\rho_A = \frac{WF_{on}}{WF_{on}} \geq 2 \text{。}$$

如算法 A 安排 J_1 的开工的时间小于 1，即 $S_1 < 1$，则新工件 J_2 将会在 $S_1 + \frac{1}{n}$ 时刻到达，其加工时间 $p_1 = 1$，权重 $w_1 = n$。由于此时 J_1 还未完工，J_2 只能单独一批在 J_1 完工之后开始加工。于是可得

$$WF_{on} = \max\{w_1 F_1, w_2 F_2\} = w_2 (C_2 - r_2) \geq w_2 (S_1 + 1 + 1 - r_2)$$

$$= n\left(S_1 + 1 + 1 - S_1 - \frac{1}{n} \right) = 2n - 1.$$

在最优序中 J_1 和 J_2 形成一批在 $S_1+\frac{1}{n}$ 时刻开工，于是可得 $WF_{opt}=n$。于是可得

$$\rho_A=\frac{WF_{on}}{WF_{opt}}\geq\frac{2n-1}{n}\xrightarrow{n\to\infty}2。$$

综上，定理 3.3 得证。 □

4. LSB 算法关于问题 Q(3,1)和 Q(3,2)的性能分析

Graham 在 1966 年的研究[18]中给出 LS (List Scheduling)算法(也称 LS 规则)的定义。在分批排序中 LS 算法可简记为 LSB (List Scheduling with Batching)算法其执行策略为：*若有机器出现空闲时刻且有工件可选择，则立即将所有未被加工的工件放在一起形成一批工件并安排在一台机器上开始加工，否则机器保持等待*。

对任意一个工件集合 I_i，用 $\sigma_1,\sigma_2,\pi_1,\pi_2$ 分别表示 LSB 算法和离线最优算法 OPT 关于 I_i 生成的可行排序和最优排序，$i=1,2$。用 $WF_{\max}(\sigma_i)$，$WF_{\max}(\pi_i)$ 分别表示排序 σ_i 和 π_i 下的目标值，$i=1,2$。下面用最小反例法证明 LSB 算法关于问题 Q(3,1)和 Q(3,2)的竞争比分别为 $\rho_1=2$ 和 $\rho_2=1$。

不妨设工件集合 I_i 满足 $WF_{\max}(\sigma_i)>\rho_i WF_{\max}(\pi_i)$ 且包含最小的工件数量 $i=1,2$。用 $B_1^i,B_2^i,\cdots,B_n^i$ 表示排序 σ_i 中依次开工的加工批其开工时间分别用 $S_1(\sigma_i),S_2(\sigma_i),\cdots,S_n(\sigma_i)$ 来表示。由 LSB 算法的执行策略和批容量无限制可知，$0=S_1(\sigma_i)<S_2(\sigma_i)<\cdots<S_n(\sigma_i)$。设第 k $(1\leq k\leq n)$批工件是第一个满足其加权流程时间等于目标值(最大的加权流程时间)的加工批。下面给出一个重要引理。

引理 4.1 (i) 每批工件中只有一个工件，即 $B_j^i=\{J_j^i\},i=1,2;j=1,2,\cdots,n$；

(ii) $k=n$，即 $WF_{\max}(\sigma_i)=w_nF_n(\sigma_i),i=1,2$；

(iii) 时间区间 $[0,S_n(\sigma_i)]$ 内机器无共同的空闲时间段。

证明若加工批 $B_l^i,1\leq l\leq n$ 中包含多个工件，则只保留加权流程时间最大的那个工件并删除其余工件，这样得到一个新的工件集合 I_i'。由 LSB 算法的执行策略可知，LSB 关于 I_i' 生成的排序中依然有 n 批工件并且每批工件的最大加权流程时间相比之前没有变化，即对目标值的贡献没有改变。而对最优排序目标值的贡献不会变大。因此构造出一个更小工件数量的反例，这显然与 I_i 是最小反例相矛盾。结论(i)得证。

若 $k<n$。令 $I_i^*=I_i\setminus(B_{k+1}^i\cup\cdots\cup B_n^i)$，则 LSB 算法关于新工件集合 I_i^* 生成的加工批与 $B_1^i,B_2^i,\cdots,B_k^i$ 完全一样。因此对在线排序目标值的贡献一样且对最优排序目标值的贡献不会变大。这与 I_i 是最小反例相矛盾。结论(ii)得证。

若在区间 $[0,S_n(\sigma_i)]$ 内机器有若干个共同的空闲时间段，则将 σ_i 中安排在最后一个公共空闲时间段之前加工的工件删除，可得到一个更小的反例，矛盾。结论(iii)得证。 □

由引理 4.1(i)以及批容量无限制可知，工件 J_n^i 在工件 J_{n-1}^i 开工之后到达，即 $r_n>S_{n-1}(\sigma_i)$。于是在最优序中 $C_n(\pi_i)\geq r_j+p_j=r_j+1$。对问题 Q(3,1)，$C_n(\pi_1)\geq r_j+1>S_{n-1}(\sigma_1)+1$。由引理 4.1(ii)，可知

$$WF_{\max}(\pi_1)\geq w_nF_n(\pi_1)=w_n(C_n(\pi_1)-r_n)>w_n(S_{n-1}(\sigma_1)+1-r_n)。$$

注意到，在将 σ_1 中，

$$WF_{\max}(\sigma_1)=w_nF_n(\sigma_1)=w_n(C_n(\sigma_1)-r_n)\leq w_n(S_{n-1}(\sigma_1)+2-r_n)$$

于是可得

$$WF_{\max}(\sigma_1)\leq w_n(S_{n-1}(\sigma_1)+2-r_n)\leq 2w_n(S_{n-1}(\sigma_1)+1-r_n)<2WF_{\max}(\pi_1)，$$

这与 I_1 是最小反例相矛盾。

对问题 Q(3,2)，由于工件不能被延迟加工结合引理 4.1(i)和(iii)，在最优序 π_2 中工件 J_n^2 依然在工件 J_{n-1}^2 开工之后开工且工件的开工时间不会变小，即 $S_n(\pi_2)=S_n(\sigma_2)$。于是有

$$WF_{\max}(\pi_2)\geq w_nF_n(\pi_2)=w_n(S_n(\sigma_2)+1-r_n)=WF_{\max}(\sigma_2)。$$

这与 I_2 是最小反例相矛盾。

综上可知，不存在最小反例 I_i 满足 $WF_{\max}(\sigma_i)>\rho_iWF_{\max}(\pi_i)$，$i=1,2$。此结论反过来证明对于问题 Q(3,1)和 Q(3,2)的任何工件集合都有 $WF_{\max}(\sigma_i)\leq\rho_iWF_{\max}(\pi_i)$ 成立。在结合定理 3.3 最终得出本节的两个主要结论。

定理 4.1 对于问题 Q(3,1)：$P_m|online,r_j,p_j=1,p-batch,b=\infty|WF_{\max}$，LSB 算法的竞争比是 2。特别的，当 $m=1$ 时，LSB 是 2-竞争的最好可能在线算法。 □

定理 4.2 对于问题 Q(3,2)：$P_m|online,r_j,p_j=1,p-batch,b=\infty,NDP|WF_{\max}$，LSB 算法是最优在线算法(竞争比是 1)。

5. 结论与展望

本文在 NDP 加工模式下分析了经典 SPT 算法关于最小化总完工时间平行机在线排序问题的性能，证明 SPT 算法是该问题的一个无界算法。在一般加工模式和 NDP 加工模式下研究了最小化最大加权流程时间分批排序问题，对工件加工时间无限制单机情形，证明不存在常数界的在线算法；对等长工件情形，分别证明 LSB 算法是 1-竞争的最优在线算法(在 NDP 加工模式下)和 2-竞争的最好可能在线算法(在一般加工模式下)。但对工件加工时间无限制多机情形的分批排序问题，还未给出确定的结果有待进一步研究。

基金项目

国家自然科学基金面上项目(No. 12371319)；河南省自然科学基金面上项目(Nos. 252300420337, 222300420503)；河南省高等学校重点科研项目(Nos. 24A110014, 22A110015)。

参考文献

[1] 唐国春, 张峰, 罗守成, 刘丽丽. 现代排序论[M]. 上海: 上海科学普及出版社, 2003.

[2] 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.

[3] Blazewicz, J., Ecker, K., Pesch, E., Schmidt G., Sterna, M. and Weglarz, J. (2019) Handbook on Scheduling—From Theory to Practice. Springer.

[4] Baker, K.R. (1974) Introduction to Sequencing and Scheduling. John Wiley & Sons.

[5] Li, W. and Yuan, 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

[6] Li, W. and Liu, H. (2023) Online NDP-Constraint Scheduling of Jobs with Delivery Times or Weights. *Optimization Letters*, **17**, 591-612. https://doi.org/10.1007/s11590-022-01889-3

[7] 李文杰, 杜智慧, 苏孟龙. 加工时间与运输时间具有一致性的单机 NDP 约束在线排序问题研究[J]. 运筹学学报, 2024, 28(4): 18-28.

[8] Qu, Y., Tian, J., Fu, R. and Ge, K. (2024) A Best Possible Online Algorithm for Single-Machine Scheduling with Non-Delayed Processing Constraint and Bounded Delivery Times. *Journal of the Operations Research Society of China*. https://doi.org/10.1007/s40305-024-00541-4

[9] Graham, R.L., Lawler, E.L., Lenstra, J.K. and Kan, A.H.G.R. (1979) Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, **5**, 287-326. https://doi.org/10.1016/s0167-5060(08)70356-x

[10] Lu, X., Sitters, R.A. and Stougie, L. (2003) A Class of On-Line Scheduling Algorithms to Minimize Total Completion Time. *Operations Research Letters*, **31**, 232-236. https://doi.org/10.1016/s0167-6377(03)00016-6

[11] Liu, P. and Lu, X. (2009) On-line Scheduling of Parallel Machines to Minimize Total Completion Times. *Computers & Operations Research*, **36**, 2647-2652. https://doi.org/10.1016/j.cor.2008.11.008

[12] Chai, X., Li, W. and Zhu, Y. (2021) Online Scheduling to Minimize Maximum Weighted Flow-Time on a Bounded Parallel-Batch Machine. *Annals of Operations Research*, **298**, 79-93. https://doi.org/10.1007/s10479-019-03352-6

[13] Mastrolilli, M. (2004) Scheduling to Minimize Max Flow Time: Off-Line and On-Line Algorithms. *International Journal of Foundations of Computer Science*, **15**, 385-401. https://doi.org/10.1142/s0129054104002480

[14] Ambühl, C. and Mastrolilli, M. (2005) On-line Scheduling to Minimize Max Flow Time: An Optimal Preemptive Algorithm. *Operations Research Letters*, **33**, 597-602. https://doi.org/10.1016/j.orl.2004.10.006

[15] Li, W. and Yuan, J. (2011) Online Scheduling on Unbounded Parallel-Batch Machines to Minimize Maximum Flow-Time. *Information Processing Letters*, **111**, 907-911. https://doi.org/10.1016/j.ipl.2011.06.008

[16] Jiao, C., Li, W. and Yuan, J. (2014) A Best Possible Online Algorithm for Scheduling to Minimize Maximum Flow-Time on Bounded Batch Machines. *Asia-Pacific Journal of Operational Research*, **31**, Article ID: 1450030. https://doi.org/10.1142/s0217595914500304

[17] Biskup, D. (1999) Single-Machine Scheduling with Learning Considerations. *European Journal of Operational Research*, **115**, 173-178. https://doi.org/10.1016/s0377-2217(98)00246-x

[18] Graham, R.L. (1966) Bounds for Certain Multiprocessing Anomalies. *Bell System Technical Journal*, **45**, 1563-1581. https://doi.org/10.1002/j.1538-7305.1966.tb01709.x