AN IMPROVED 2.11-COMPETITIVE ALGORITHM FOR ONLINE SCHEDULING ON PARALLEL MACHINES TO MINIMIZE TOTAL WEIGHTED COMPLETION TIME

Ran Ma#
School of Mathematics and Information Science
Henan Polytechnic University, Jiaozuo, 454000, China

Jiping Tao∗
Department of Automation
Xiamen University, Xiamen, 361005, China

(Communicated by Jie Sun)

ABSTRACT. We revisit the classical online scheduling problem on parallel machines for minimizing total weighted completion time. In the problem, a set of independent jobs arriving online over time has to be scheduled on identical machines, where the information of each job including its processing time and weight is not known in advance. The goal is to minimize the total weighted completion time of the jobs. For this problem, we propose an improved 2.11-competitive online algorithm based on a kind of waiting strategy.

1. Introduction. Until two decades ago, one of the basic assumptions made in deterministic scheduling was that all of the information needed to define the problem instance was known in advance. This assumption is usually not valid in practice, however. Abandoning it has led to the rapidly emerging field of on-line scheduling. A simple instance is how to arrange the printing materials that can arrive at any time in a print shop. The introduction of online scheduling greatly enriched the field of scheduling. Since then, volumes of research has focused on design, implementation, and analysis algorithms for an online scheduling problem. In contrast to the off-line version, an online algorithm must produce a sequence of decisions based on past events without any information about the unreleased jobs. The lack of knowledge of the future does not generally guarantee the optimality of the schedule generated by an online algorithm. Thus a natural issue is how to evaluate different online algorithms for a same scheduling problem. A widely used approach to evaluate...
an online algorithm is the competitive ratio. For a minimization problem, the competitive ratio \( \rho_A \) of an online algorithm \( A \) is defined to be

\[
\rho_A = \sup \{ \frac{A(I)}{\text{OPT}(I)} : I \text{ is an instance with } \text{OPT}(I) > 0 \},
\]

where, for an instance \( I \), \( A(I) \) is used to denote the objective value of the schedule generated by the online algorithm \( A \), and \( \text{OPT}(I) \) is the objective value of an optimal (off-line) schedule. An online algorithm \( A \) is called best possible if no online algorithm has a competitive ratio less than \( \rho_A \).

In this paper, we focus on a classical online scheduling problem on identical parallel machines to minimize total weighted completion time. More formally, a set of independent jobs arriving online over time has to be scheduled on \( m \) identical machines, where preemption is not allowed and the information of each job, including its processing time \( p_j \geq 0 \), release date \( r_j \geq 0 \), and weight \( w_j > 0 \), is not known in advance. The objective is to minimize the total weighted completion time of the jobs. This widely used scheduling objective is a proxy measure for work-in-process inventory cost in a make-to-stock environment and for customer service in a make-to-order environment. Then the problem studied in this paper can be written in the three-field notation of Graham et al. [5] as \( P | \text{online}, r_j | \sum w_j C_j \).

**Previous work:** There has been an enormous amount of work on online scheduling. We do not intend to do a complete review of results in the area and restrict our attention to some of the most relevant literature on online scheduling problems directly related to the matter of this paper.

To minimize total (weighted) completion time is one of the most basic objective functions in scheduling research. For problem \( 1 || \sum w_j C_j \), Smith [14] showed that the shortest weighted processing time (SWPT) rule, in which jobs are sequenced in nondecreasing order of \( \frac{p_j}{w_j} \), provides an optimal solution. In this paper, we call \( \tau_j = \frac{p_j}{w_j} \) the Smith-ratio of job \( J_j \).

For problem \( 1 | \text{online}, r_j | \sum C_j \), Vestjens [17] showed that no online algorithm can have a competitive ratio of less than 2. Vestjens [17] and Philips et al. [12] presented distinct online algorithms with a best possible competitive ratio of 2. As for the weighted version on a single machine, the first deterministic online algorithm for \( 1 | \text{online}, r_j | \sum w_j C_j \) was presented by Hall et al. [6]. They designed a \((3 + \epsilon)\)-competitive algorithm by taking advantage of a general online framework. Using the idea of \( \alpha \)-points and mean-busy-time relaxations, Goemans et al. [4] designed a deterministic \( 2.4143 \)-competitive algorithm and a randomized \( 1.6853 \)-competitive algorithm for \( 1 | \text{online}, r_j | \sum w_j C_j \). Until 2004, for problem \( 1 | \text{online}, r_j | \sum w_j C_j \), Anderson and Potts [1] presented a best possible deterministic online algorithm, called Delayed SWPT (DSWPT for short), with a competitive ratio of 2 under the assumption that all release dates and processing times are integers. The main technique used in the competitive ratio analysis in Anderson and Potts [1] is to create a “doubled problem” and an “extended problem”.

For problem \( P | \text{online}, r_j | \sum w_j C_j \), Vestjens [17] proved that any deterministic algorithm has a competitive ratio of at least 1.309. This is the best known lower bound for the problem we focus on. To the best of our knowledge, the first online algorithm with constant competitive performance guarantee for the problem above is given by Hall et al. [6]. They designed a \((4 + \epsilon)\)-competitive online algorithm also based on the general online framework. Megow and Schulz [11] provided an improved deterministic \( 3.28 \)-competitive online algorithm. Liu and Lu [8] generalized the online algorithm and the analysis technique of “quasi-schedule” in Vestjens [17].
from a single machine to identical machines for the case that all jobs have identical weights and obtained a 2-competitive online algorithm. In a celebrated paper, Correa and Wagner [3] proposed a 2.618-competitive online algorithm for problem \( P_{\text{online}}, r_j \sum w_j C_j \) by combining the \( \alpha \)-point method with linear programming relaxation. In [13], Sitters designed an online algorithm with the competitive ratio not greater than \( \alpha_m = \left(\frac{1+1/\sqrt{m}}{2}+\frac{3}{1+1/\sqrt{m}}\right)^2 \) by utilizing the technique of shifting release times. The value \( \alpha_m \) is very great when the machine number \( m \) is small, although it tends to 1.791 when \( m \) tends to infinity.

In a recent study, Tao [15] revisited problem \( P_{\text{online}}, r_j \sum w_j C_j \) and presented an online algorithm AD-SWPT (Average Delayed Shortest Weighted Processing Time). By the technique of instance reduction, they showed that AD-SWPT is \( (2.5 - 1/2m) \)-competitive, where \( m \) is the number of the machines. Also based on the same instance reduction, by supplying the following algorithm Improved AD-SWPT, Tao et al. [16] improved the above result to \( (1 + \frac{m}{4m^2 - 2m + 1}) \)-competitive, which tends to 2.28 when \( m \) tends to infinity. Let \( t \) be the current decision time, \( \hat{p}_j(t) \) the remaining processing time of job \( J_j \) at time \( t \) in an online schedule, \( S_j \) the starting time of job \( J_j \) in the online schedule. The Improved AD-SWPT rule can be described in detail as follows:

**Improved AD-SWPT:** Whenever there is one idle machine and some jobs are available, choose a job with the smallest Smith-ratio \( p_j/w_j \) among all the arrived and unscheduled jobs. Suppose that \( J_i \) is chosen as the candidate job. If
\[
\frac{\hat{p}_i(t)}{m} \leq \alpha \frac{\sum \hat{p}_j(t)}{m},
\]
when \( \sum \hat{p}_j(t) \) is the total remaining processing time at all the busy machines at time \( t \), then we schedule \( J_i \) from \( t \) at the idle machine; otherwise, wait until a new job arrives or the above equation satisfies, where \( \alpha \geq 1 \) is a parameter to be designed.

From the description of Improved AD-SWPT, Improved AD-SWPT is an algorithm obtained by pumping a flexible parameter \( \alpha \) into AD-SWPT presented in [15], and so Improved AD-SWPT can be reduced to AD-SWPT when parameter \( \alpha = 1 \). Thus, the competitive ratio of Improved AD-SWPT should be lower than or equal to that of AD-SWPT.

For the online scheduling problem \( P_{\text{online}}, r_j \sum w_j C_j \) considered in our paper, Tao et al. in [16] provided Improved AD-SWPT algorithm and showed that Improved-ADSWPT is a polynomial-time online algorithm with a competitive ratio of at most 2.28. We improve the above result and show that Improved-ADSWPT is a polynomial-time online algorithm with a competitive ratio of at most 2.11.

**Our contribution:** For online scheduling problem \( P_{\text{online}}, r_j \sum w_j C_j \), we establish that Improved AD-SWPT is 2.11-competitive by combining the technique “improved instance reduction” together with the lower bounds provided respectively in Kawaguchi and Kyan [7] and Chou et al. [2]. In this case, though we cannot show that our analysis is tight, according to the computational study provided in [15], the remaining gap is at most 0.11. Finally, we present our conjecture that Improved AD-SWPT has a competitive ratio of 2.

2. **Preliminaries.** The following notations will be used throughout this paper:

- \( I \): the job instance \( I \).
- \( J_j \): the job of index \( j \) in \( I, j = 1, 2, ..., n \).
- \( p_j \): the processing time of job \( J_j \).
- \( r_j \): the release date of job \( J_j \).
- \( w_j \): the weight of job \( J_j \).
\begin{itemize}
  \item $\tau_j = p_j / w_j$: the Smith-ratio of job $J_j$.
  \item $\tau_{\text{max}}$: the maximum Smith-ratio of the jobs.
  \item $(I, \sigma)$: an instance-schedule pair if $\sigma$ is a feasible schedule of the job instance $I$.
  \item $S_j(I, \sigma)$: the starting time of job $J_j$ in $(I, \sigma)$.
  \item $C_j(I, \sigma) = S_j(I, \sigma) + p_j$: the completion time of job $J_j$ in $(I, \sigma)$.
  \item For a subset $T \subseteq I$, we use $S_T(I, \sigma)$ and $C_T(I, \sigma)$ to denote the first starting time and the last completion time of the jobs in $T$ in $(I, \sigma)$, respectively. Especially, when $T = I$, we write $S_I(I, \sigma)$ and $C_I(I, \sigma)$ for short.
  \item $Z_j(I, \sigma) = w_j(I) C_j(I, \sigma)$: the contribution of job $J_j \in I$ to the objective value of $(I, \sigma)$.
  \item $Z(I, \sigma) = \sum_{J \in I} Z_j(I, \sigma)$: the objective value of $(I, \sigma)$.
  \item For a subset $T \subseteq I$, we use $Z_T(I, \sigma) = \sum_{J \in T} Z_j(I, \sigma)$ to denote the total contribution of the jobs in $T$ to $Z(I, \sigma)$.
  \item $\pi(I)$: an optimal (offline) schedule of the job instance $I$.
  \item $\text{OPT}(I)$: the objective value of an optimal (offline) schedule $\pi(I)$.
  \item $t$: the current decision time.
  \item $\hat{p}_j(t)$: the remaining processing time of job $J_j$ at time $t$ in a feasible schedule.
\end{itemize}

The following lemmas, which will be repeatedly used in our competitive analysis, are provided without proof, since they can be derived from the basic mathematics and have been frequently used in the literature such as [10, 9, 15].

**Lemma 2.1.** Let $f(x)$ and $g(x)$ be two positive-valued functions defined in the interval $[u, v]$, where $f(x)$ is convex and $g(x)$ is concave. Then $\frac{f(x)}{g(x)}$ reaches its maximum value at one endpoint of the interval, i.e., $\frac{f(x)}{g(x)} \leq \max \left\{ \frac{f(u)}{g(u)}, \frac{f(v)}{g(v)} \right\}$ for any $x \in [u, v]$.

**Lemma 2.2.** Let $f(I)$ and $g(I)$ be two positive-valued functions defined on the job instance $I$. If $I$, $I_1$, and $I_2$ are three instances such that $f(I) \leq f(I_1) + f(I_2)$ and $g(I) \geq g(I_1) + g(I_2)$, then $\frac{f(I)}{g(I)} \leq \frac{f(I_1) + f(I_2)}{g(I_1) + g(I_2)} \leq \max \left\{ \frac{f(I_1)}{g(I_1)}, \frac{f(I_2)}{g(I_2)} \right\}$.

**Lemma 2.3.** Let $I$ be a job instance. Let $I'$ be a job instance obtained from $I$ by modifying the release date of job $J_j$ as $r'_j \leq r_j$. Then $\text{OPT}(I') \leq \text{OPT}(I)$.

**Lemma 2.4.** Let $I'$ and $I''$ be a partition of the job instance $I$. Then $\text{OPT}(I) \geq \text{OPT}(I') + \text{OPT}(I'')$.

Let $I$ be a job instance. For each $T \subseteq I$ and each positive number $\delta$, we use $I^{(T, \delta)}$ to denote the job instance obtained from $I$ satisfying that the weight $w_j(I)$ of each job $J_j \in T$ is modified as $w_j(I^{(T, \delta)}) = \delta \cdot w_j(I)$. Since each job $J_j \in I^{(T, \delta)}$ has a release date $r_j(I^{(T, \delta)}) = r_j(I)$, a feasible schedule of $I$ is also a feasible schedule of $I^{(T, \delta)}$, and vice versa. Denote by $\sigma|T$ the subschedule of the jobs restricted to $T$ in schedule $\sigma(I)$. We have the following useful lemmas, which are similar to the lemmas in Ma and Yuan [10]. For the sake of completeness, we provide the lemmas together with the proof.

**Lemma 2.5.** Let $X(I^{(T, \delta)})$ be a schedule of $I^{(T, \delta)}$ which processes each job in the same time interval as its corresponding job in $\sigma(I)$ for each $\delta \in [u, v]$. Then...
Let \( \Pi \) be the set of all feasible schedules of instance \( I^{(T,\delta)} \). For each \( \delta \in [0, \delta_0] \), we have \( Z(I^{(T,\delta)}, \Pi) = Z(I, \sigma) + (\delta - 1) \cdot \sum_{j \in T} Z_j(I, \sigma) \), which implies that \( Z(I^{(T,\delta)}, \theta) \) is a linear function in \( \delta \in [0, \delta_0] \). Since \( \text{OPT}(I^{(T,\delta)}) = \min\{Z(I^{(T,\delta)}, \theta) : \theta \in \Pi\} \), we conclude that \( \text{OPT}(I^{(T,\delta)}) \) is a concave function in \( \delta \in [0, \delta_0] \). Thus, the lemma follows.

**Lemma 2.6.** Let \((I, \sigma)\) be an instance-schedule pair such that \( Z(I, \sigma) > \lambda(\alpha) \cdot \text{OPT}(I) \). Suppose that \( T \) is a proper subset of \( I \) such that \( Z_T(I, \sigma) \leq \lambda(\alpha) \cdot \text{OPT}(T) \). Let \( \delta \) be an arbitrary positive number with \( \delta \leq 1 \). Let \( X^{(T,\delta)} \) be a schedule of \( I^{(T,\delta)} \) which processes each job in the same time interval as its corresponding job in \( \sigma(I) \) for each \( \delta \in [\delta_0, M] \) where \( M > 1 \). Then \( X^{(T,\delta)} \) is an instance-schedule pair such that \( Z(I^{(T,\delta)}, X) / \text{OPT}(I^{(T,\delta)}) \geq Z(I, \sigma) / \text{OPT}(I) \). Consequently, \( Z(I^{(T,\delta)}, X) > \lambda(\alpha) \cdot \text{OPT}(I^{(T,\delta)}) \).

**Proof.** Suppose to the contrary that \( Z(I^{(T,\delta)}, X) / \text{OPT}(I^{(T,\delta)}) < Z(I, \sigma) / \text{OPT}(I) \). For each \( \delta \in [\delta_0, M] \), from Lemma 2.5, \( Z(I^{(T,\delta)}, X) \) is a monotonously increasing linear function and \( \text{OPT}(I^{(T,\delta)}) \) is a concave function in \( \delta \in [\delta_0, M] \), respectively. By Lemma 2.1, we have

\[
\frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})} \leq \max \left\{ \frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})}, \frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})} \right\}
\]

for all \( \delta \in [\delta_0, M] \). Note that \( 1 \in [\delta_0, M] \), we have

\[
\frac{Z(I, \sigma)}{\text{OPT}(I)} \leq \max \left\{ \frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})}, \frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})} \right\}.
\]

Together with the assumption that \( Z(I^{(T,\delta)}, X) / \text{OPT}(I^{(T,\delta)}) < Z(I, \sigma) / \text{OPT}(I) \), we have

\[
\frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})} \geq \frac{Z(I, \sigma)}{\text{OPT}(I)},
\]

for every \( M > 1 \).

From the fact that \( T \) is a proper subset of \( I \) such that \( Z_T(I, \sigma) \leq \lambda(\alpha) \cdot \text{OPT}(T) \), we have

\[
\lim_{M \to +\infty} \frac{Z(I^{(T,\delta)}, X)}{\text{OPT}(I^{(T,\delta)})} = \frac{Z_T(I, \sigma)}{\text{OPT}(T)} \leq \lambda(\alpha).
\]

Thus, we conclude that \( Z(I, \sigma) / \text{OPT}(I) \leq \lambda(\alpha) \). This contradicts the assumption that \( Z(I, \sigma) / \text{OPT}(I) > \lambda(\alpha) \). The lemma follows.

**Lemma 2.7.** Let \((I, \sigma)\) be an instance-schedule pair such that \( Z(I, \sigma) > \lambda(\alpha) \cdot \text{OPT}(I) \). Suppose that \( T \) is a proper subset of \( I \) such that \( Z_T(I, \sigma) \leq \lambda(\alpha) \cdot \text{OPT}(I \setminus T) \). Let \( \delta \) be an arbitrary number with \( \delta > 1 \). Let \( X(I^{(T,\delta)}) \) be a schedule of \( I^{(T,\delta)} \) which processes each job in the same time interval as its corresponding job in \( \sigma(I) \) for each \( \delta \in [0, \delta] \). Then \( X(I^{(T,\delta)}) \) is an instance-schedule pair such
that $Z(I(T,\overline{\sigma}), X)/\operatorname{OPT}(I(T,\overline{\sigma})) \geq Z(I, \sigma)/\operatorname{OPT}(I)$. Consequently, $Z(I(T,\overline{\sigma}), X) > \lambda(\alpha) \cdot \operatorname{OPT}(I(T,\overline{\sigma}))$.

**Proof.** Let $\overline{T} = I \setminus T$. Then $Z(I, \sigma) \leq \lambda(\alpha) \cdot \operatorname{OPT}(I \setminus T)$. Let $\delta = 1/\lambda$. Then $\delta \leq 1$. From Lemma 2.6, we have $Z(I(T,\overline{\sigma}), X)/\operatorname{OPT}(I(T,\overline{\sigma})) \geq Z(I, \sigma)/\operatorname{OPT}(I)$. Since $Z(I(T,\overline{\sigma}), X) = \delta \cdot Z(I(T,\overline{\sigma}), \sigma)$ and $\operatorname{OPT}(I(T,\overline{\sigma})) = \delta \cdot \operatorname{OPT}(I(T,\overline{\sigma}))$, we also have $Z(I(T,\overline{\sigma}), X)/\operatorname{OPT}(I(T,\overline{\sigma})) \geq Z(I, \sigma)/\operatorname{OPT}(I)$. The lemma follows. $\square$

Suppose that $(I, \sigma)$ be an instance-schedule pair. In what follows of this paper, according to the increasing order of the starting times of jobs in $\sigma(I)$, all jobs are denoted by $J_1, J_2, \ldots, J_n$, with ties broken in nondecreasing Smith-ratio. We call $I$ a *regular instance* if all jobs in $I$ have a common Smith-ratio $\tau$ and there does not exist a time $t$ between the earliest processing time and the latest completion time in online schedule $\sigma(I)$ such that all the machines remain idle at time $t$. Now we will give Lemma 2.8 and Lemma 2.9 for a regular instance. As dedicated in the proof, Lemma 2.8 and Lemma 2.9 assert that if $I$ is a regular instance, then we have $Z(I, \sigma)/\operatorname{OPT}(I) \leq \lambda(\alpha) = \max\{1 + \alpha, 1 + \frac{\sqrt{2}}{2} + \frac{1}{\alpha}\}$. Though the proof in Lemma 2.8 is similar with [16], we can simplify the complicated deduction in [15] and [16] by taking advantage of the following lower bound in [2] instead of the lower bound from mean-busy-time in [15] and [16].

For problem $Pm[\text{online}]$, $r_j | \sum w_j C_j$ with $p_j = w_j$ for each job, from [2], the objective value $u(I)$ of optimal schedules satisfies an inequality of

$$u(I) \geq r_{\min} \sum p_j + \frac{1}{2m} (\sum p_j)^2 + \frac{1}{2} \sum p_j^2,$$

where $r_{\min}$ is the earliest release time.

Since $I$ is a regular instance, we can normalize the ratio of $p_j/w_j$ to 1 by rescaling the weights of all jobs. Consider the time in $\sigma(I)$, say $r_L$, from which jobs are continuously processed after $r_L$ at each machine without idle time between jobs. Now we present Lemma 2.8 and Lemma 2.9 according to two different cases of the jobs processed after $r_L$, respectively.

**Lemma 2.8.** If there does not exist a job which is released before $r_L$ and is scheduled at or after $r_L$ in $\sigma(I)$, then $Z(I, \sigma)/\operatorname{OPT}(I) \leq 1 + \alpha$.

**Proof.** Consider these jobs that start at or after $r_L$, also denoted by $J_s, J_{s+1}, \ldots, J_n$. The assumption in the lemma implies that $J_s, J_{s+1}, \ldots, J_n$ must be released at or after $r_L$, and so these jobs have no effect on the jobs starting before $r_L$. Construct an intermediate instance $I_1 = I \setminus \{J_s, J_{s+1}, \ldots, J_n\}$. Then we have

$$Z(I, \sigma) = Z(I_1, \sigma) + \sum_{j=s}^{n} (S_j + p_j)w_j.$$  

(2)

Note that the jobs $J_s, J_{s+1}, \ldots, J_n$ are continuously processed after $r_L$ at each machine. So we can limit the starting time of the $j$th job in $\{J_s, J_{s+1}, \ldots, J_n\}$ as

$$S_j \leq r_L + \frac{\sum_{i<s} \hat{p}_i(r_L) + \sum_{s<i<j} p_i}{m}, \quad j = s, s + 1, \ldots, n,$$

where the second term is to average the total processing time which has to be finished between $r_L$ and $S_j$ over all the machines.
Lemma 2.9. If there exists at least one job which is released before \( r_L \), then \( Z(I, \sigma) \) can be bounded by an upper bound as

\[
Z(I, \sigma) = Z(I_1, \sigma) + \frac{n}{m} \sum_{j=s}^{n} (S_j + p_j)w_j
\]

Proof. According to Improved AD-SWPT, we can limit \( Z(I, \sigma) \) by an upper bound as

\[
Z(I, \sigma) \leq Z(I_1, \sigma) + \frac{n}{m} \sum_{j=s}^{n} \left( r_L + \frac{A + \sum_{k=s}^{j-1} p_k}{m} + p_j \right) p_j
\]

\[
= Z(I_1, \sigma) + \frac{n}{m} \sum_{j=s}^{n} \left( r_L + \frac{A}{m} + \frac{\sum_{k=s}^{j} p_k}{m} \right) p_j + \frac{n}{m} \sum_{j=s}^{n} \frac{p_j^2}{m}
\]

\[
= Z(I_1, \sigma) + \frac{(r_L + A/m)B + B^2}{2m} + \frac{1}{2} \sum_{j=s}^{n} \frac{p_j^2}{m}.
\]

Consider the set of \( \{J_s, J_{s+1}, \ldots, J_n\} \) as a separate instance, and further relax the release times of all the jobs to \( r_L \), then we can develop a lower bound on the optimal schedule \( \pi(I) \) according to (1).

\[
\text{OPT}(I) \geq \text{OPT}(I_1) + \text{OPT}(\{J_s, J_{s+1}, \ldots, J_n\})
\]

\[
\geq \text{OPT}(I_1) + r_LB + \frac{B^2}{2m} + \frac{1}{2} \sum_{j=s}^{n} \frac{p_j^2}{m}.
\]

According to Improved AD-SWPT, we have \( \sum_{S_j \leq r_L} \hat{p}_j(r_L)/m \leq \alpha r_L \), and so \( A/m \leq \alpha r_L \). Combining (4) and (5), together with the fact \( \alpha \geq 1 \), we have

\[
\frac{Z(I, \sigma)}{\text{OPT}(I)} \leq \max \left\{ \frac{Z(I_1, \sigma)}{\text{OPT}(I_1)}, \frac{(r_L + A/m)B + B^2}{2m} + \frac{1}{2} \sum_{j=s}^{n} \frac{p_j^2}{m} \right\}
\]

\[
\leq \max \left\{ \frac{Z(I_1, \sigma)}{\text{OPT}(I_1)}, \frac{(r_L + A/m)B}{r_L B} \right\}
\]

\[
\leq \max \left\{ \frac{Z(I_1, \sigma)}{\text{OPT}(I_1)}, 1 + \alpha \right\}.
\]

Rewrite \( I_1 \) as \( I \) and repeat the analysis above. Ultimately the performance ratio of \( I \) can be bounded by \( 1 + \alpha \). The lemma follows.

\[Q \]

Lemma 2.9. If there exists at least one job which is released before \( r_L \) and starts at or after \( r_L \) in \( \sigma(I) \), then \( Z(I, \sigma)/\text{OPT}(I) \leq \frac{1 + \sqrt{2}}{2} + \frac{1}{8} \).

Proof. According to Improved AD-SWPT, there is a job \( J_k \) must satisfy

\[
p_k + \sum_{S_j < r_L} \hat{p}_j(r_L) \geq \alpha r_L.
\]

Consider these jobs which are completed after \( r_L \), also denoted by \( J_s, J_{s+1}, \ldots, J_n \). Construct an intermediate instance \( I_1 = I \setminus \{J_s, J_{s+1}, \ldots, J_n\} \). Divide the set of \( \{J_s, J_{s+1}, \ldots, J_n\} \) into two subsets as follows:

\[
Q_1 = \{J_j : \text{S}_j < r_L, \text{C}_j > r_L\} \cup \{J_k\}, \quad Q_2 = \{J_j : \text{S}_j \geq r_L\} \setminus \{J_k\}.
\]

Let \( \sum_{J_j \in Q_1} p_j := A \), and \( \sum_{J_j \in Q_2} p_j := B \).

For the purpose of our performance analysis, now we take only the jobs in \( Q_1 \cup Q_2 \) into consideration and create a new schedule \( \sigma^*(Q_1 \cup Q_2) \) of \( Q_1 \cup Q_2 \) as follows:
LRF (Largest-Ratio-First) schedule is at most a factor \( K \). Kawaguchi and Kyan [7] showed that the total weighted completion time of an \( r \) and so that \( \text{OPT}(I) \)

we can regard this as a new lower bound of our problem. From (1), we can obtain

Then we have

Recall that in a well-known paper, for scheduling problem \( Pm|| \sum w_j C_j \), Kawaguchi and Kyan [7] showed that the total weighted completion time of an LRF (Largest-Ratio-First) schedule is at most a factor \( \frac{1 + \sqrt{2}}{2} \) larger than the optimal total weighted completion time. This means that, for scheduling problem \( Pm|| \sum w_j C_j \), any LS schedule is at most a factor \( \frac{1 + \sqrt{2}}{2} \) larger than the optimal total weighted completion time for an instance in which \( w_j = p_j \) holds for all jobs. We can regard this as a new lower bound of our problem. From (1), we can obtain that

\[
\text{OPT}(I') \leq \frac{(A + B)^2}{2m} + \frac{1}{2} \sum_{Q_1 \cup Q_2} p_j^2.
\]

From (7), we can derive that \( A/m \geq \alpha r_L \) and so \( r_L \leq \frac{A}{\alpha m} \). Furthermore we can obtain \( \sum_{J \in Q_j} p_j^2 \geq A^2/m \) because there are at most \( m \) jobs in \( Q_1 \). Thus, we have

\[
Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j
\]

\[
= Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j
\]

\[
\leq \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
\leq \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
\leq \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
\leq \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
\leq \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
= \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

\[
= \max\{ Z(I, \sigma) + \frac{1 + \sqrt{2}}{2} \text{OPT}(I') + \sum_{J \in Q_1 \cup Q_2} r_L p_j \}
\]

Rewrite \( I \) as \( I \) and repeat the analysis above. Ultimately the performance ratio of \( I \) can be bounded by \( \frac{1 + \sqrt{2}}{2} + \frac{1}{\alpha} \). The lemma follows.

3. **Algorithm analysis.** In order to apply the approach of “improved instance reduction” correctly, we present the following new online algorithm Flexible Improved AD-SWPT or FIADSWPT in short by pumping into Improved AD-SWPT some flexibility.

**FIADSWPT:** Whenever there is one idle machine and some jobs are available, choose a job with the smallest Smith-ratio \( \tau_j = p_j/w_j \) among all the arrived and
unscheduled jobs. Suppose $J_i$ is chosen as the candidate job. If 
\[
\frac{p_i + \sum_{j \leq t} \hat{p}_j(t)}{m} \leq \alpha t,
\]
where $\sum_{j \leq t} \hat{p}_j(t)$ is the total remaining processing time at all the busy machines 
at time $t$, then we schedule $J_i$ from $t$ at the idle machine; otherwise, wait until the 
next time, where $\alpha \geq 1$ is a parameter to be designed. \qed

In FIADSWPT, a candidate job at time $t$ is a job with the smallest Smith-ratio among all the arrived and unscheduled jobs. Moreover, a candidate job $J_i$ is called a ready job at time $t$ if 
\[
\frac{p_i + \sum_{j \leq t} \hat{p}_j(t)}{m} \leq \alpha t
\]
at time $t$. Due to the flexibility, algorithm FIADSWPT is not polynomial-time; in addition, for a given job instance, FIADSWPT may generate different schedules, each one of them is called a possible schedule generated by FIADSWPT. But, it is the flexibility of FIADSWPT that enables us to use the technique of “improved instance-reduction” freely in the 
analysis of the competitive ratio of FIADSWPT. We finally will show in this paper that, for a given job instance, each possible schedule generated by FIADSWPT has an objective value at most $\lambda(\alpha)$ times the objective value of an optimal (off-line) schedule. Note that the schedule generated by Improved AD-SWPT is also a possible schedule generated by FIADSWPT. Then we can conclude that Improved AD-SWPT is a $\lambda(\alpha)$-competitive polynomial-time online algorithm for problem $P|\text{online}, r_j| \sum w_j C_j$.

For a job instance $I$ and a schedule $\sigma$ obtained by FIADSWPT on $I$, we call $(I, \sigma)$ an online instance-schedule pair. The following lemma characterizes the online instance-schedule pairs $(I, \sigma)$.

**Lemma 3.1.** Let $I$ be a job instance and let $\sigma$ be a feasible schedule of $I$. $(I, \sigma)$ is an online instance-schedule pair if and only if the following two conditions are satisfied:

(i) For each job $J_i$, 
\[
\frac{p_i + \sum_{j \leq t} \hat{p}_j(t)}{m} \leq \alpha S_i(I, \sigma),
\]
and $J_i$ is a job with the smallest Smith-ratio at time $S_i(I, \sigma)$.

(ii) For each job $J_j$ with $S_j(I, \sigma) > \max\{r_j(I), \frac{p_j + \sum_{i \leq t} \hat{p}_i(t)}{\alpha m}\}$, at each idle time $t$ with 
\[
\max\{r_j(I), \frac{p_j + \sum_{i \leq t} \hat{p}_i(t)}{\alpha m}\} \leq t < S_j(I, \sigma),
\]
in $(I, \sigma)$, there is a candidate job $J_i \neq J_j$ with the smallest Smith-ratio among all jobs at time $t$ in $(I, \sigma)$ such that 
\[
\tau_i(I) \leq \tau_j(I) \text{ and } \frac{p_i + \sum_{j \leq t} \hat{p}_j(t)}{m} \leq \alpha S_i(I, \sigma).
\]

It is sufficient to show that $Z(I, \sigma)/\text{OPT}(I) \leq \lambda(\alpha)$ for each online instance-schedule pair $(I, \sigma)$. We will take advantage of the approach of “improved instance reduction” in our analysis. By combining “improved instance reduction” with contradiction, we will search for the instances $I$ with a special structure such that 
$Z(I, \sigma)/\text{OPT}(I) \geq \lambda(\alpha)$ until a specific instance with a more special structure is obtained. In fact, the specific instance is just a regular instance. However, Lemma 2.8 and Lemma 2.9 told us that $Z(I, \sigma)/\text{OPT}(I) \leq \lambda(\alpha)$ for any regular instance. Hence, we can come to the conclusion that $Z(I, \sigma)/\text{OPT}(I) \leq \lambda(\alpha)$ for each online instance-schedule pair $(I, \sigma)$.

An instance $I$ is said to be a counterexample if $Z(I, \sigma)/\text{OPT}(I) > \lambda(\alpha)$ for a certain possible schedule $\sigma(I)$ generated by FIADSWPT on instance $I$. Furthermore, $I$ is called a smallest counterexample if $I$ is a counterexample such that the number of jobs in $I$ is as small as possible. For the case that $I$ is a smallest counterexample, we also say that $(I, \sigma)$ is a smallest violated online instance-schedule pair. We define $\Omega$ to be the set of all smallest violated online instance-schedule pairs. Suppose to
the contrary that FIADSWPT has a competitive ratio greater than $\lambda(\alpha)$. Then $\Omega$ is not empty.

We call a set of jobs a block if there does not exist a time $t$ in the time interval of the processing of the set of jobs such that all the machines remain idle at $t$ in $\sigma(I)$. Thus, the jobs in $\sigma(I)$ are divided by idle intervals into some blocks of $\sigma(I)$. We further partition the jobs in each block of $\sigma(I)$ into subblocks such that within each subblock the jobs are scheduled in SWPT-rule, and that the ratio $\tau_j$ of the last job of a subblock is larger than that of the first job of the succeeding subblock if it exists. We first establish some lemmas to reveal the properties of $(I, \sigma)$ in the following.

**Lemma 3.2.** For each $(I, \sigma) \in \Omega$, there is no job $J_j$ in $I$ such that $\tau_j = 0$.

*Proof.* It suffices to show that there is no job $J_j$ in $I$ such that $p_j = 0$. Suppose to the contrary that there is some job $J_j \in I$ with $p_j = 0$. By algorithm FIADSWPT, $J_j$ will be processed as soon as one of the machines becomes idle after it is released at time $r_j$. We distinguish the following into two cases:

**Case 1.** $J_j$ starts at time $r_j$ in $\sigma(I)$. Then $I' = I \setminus \{J_j\}$ is a counterexample with a smaller number of jobs since $Z(I, \sigma)/\OPT(I) \leq Z(I', \sigma)/\OPT(I')$. This contradicts the minimality of $I$.

**Case 2.** $J_j$ is processed after time $r_j$ in $\sigma(I)$. This means that all machines are busy at time $r_j$. Let $J_k$ denote the latest processed job before $r_j$ and $S_i$ the starting time of job $J_i$. Note that $S_i$ is the latest starting time before $r_j$, then all machines are busy at time $S_i$. By FIADSWPT, we can obtain that $\frac{p_j + \sum_{s_j \leq s_i} \hat{p}_j(s_i)}{\alpha S_i} \leq \alpha S_i$, and so there must exist a job, say $J_k$, such that $S_k \leq r_j$ and $\hat{p}_k(S_k) \leq \alpha S_k$. Thus, $C_j(\sigma(I)) = C_k(\sigma(I)) = S_i(\sigma(I)) + \hat{p}_k(S_i) \leq (1 + \alpha) S_i \leq (1 + \alpha) r_j$. Let $I' = I \setminus \{J_j\}$. Then we have

$$Z(I, \sigma) \leq \frac{Z(I', \sigma) + w_j C_j(\sigma(I))}{\OPT(I') + w_j r_j} \leq \frac{Z(I', \sigma) + (1 + \alpha) w_j S_i}{\OPT(I') + w_j S_i} \leq \max \left\{ \frac{Z(I', \sigma)}{\OPT(I')}, 1 + \alpha \right\}.$$ 

Note that $Z(I, \sigma)/\OPT(I) > 1 + \alpha$. Thus, $Z(I, \sigma)/\OPT(I) \leq Z(I', \sigma)/\OPT(I')$, and so $I'$ is a counterexample with a smaller number of jobs, contradicting the choice of $I$. The lemma follows. □

Consider an instance-schedule pair $(I, \sigma) \in \Omega$. The last job in a block $B$ of $\sigma(I)$ is called the end-job of $B$ in $\sigma(I)$. For each job $J_k \in \sigma(I)$, we use $B^{(k)}$ to denote the block in $\sigma(I)$ including $J_k$.

Let $J_k \in \sigma(I)$ which is not the end-job of $B^{(k)}$. We use $J_{k+}$ to denote the successor job of $J_k$, i.e., $J_{k+} \in B^{(k)}$ and there is no job to start between time $S_k(\sigma(I))$ and time $S_k(\sigma(I))$. We call $(J_k, J_{k+})$ an SWPT-reverse pair in $\sigma(I)$ if $\tau_k(\alpha) > \tau_{k+}(\alpha)$. In the case that there is no other SWPT-reverse pair $(J_i, J_{i+})$ in $\sigma(I)$ with $S_j(\sigma(I)) > S_k(\sigma(I))$, we also call $(J_k, J_{k+})$ the last SWPT-reverse pair in $\sigma(I)$.

For an SWPT-reverse pair $(J_k, J_{k+})$ in $\sigma(I)$, the following notations are used in our deduction.
• $I^{(k)} = I^{(k)}(\sigma(I))$ is the maximal subset of jobs such that
  (i) $J_{k^+} \in I^{(k)}$,
  (ii) the jobs in $I^{(k)}$ are consecutively scheduled in $\sigma(I)$, and
  (iii) $\tau_j(I) = \tau_{k^+}(I)$ for all jobs $J_j \in I^{(k)}$.
• $S_{f(I^i)}(\sigma(I))$, $\bar{S}_{f(I^i)}(\sigma(I))$ and $C_{f(I^i)}(\sigma(I))$ are the first starting time, the last starting time and the last completion time of the jobs in $I^{(k)}$, respectively, in $\sigma(I)$. Then $S_{f(I^i)}(\sigma(I)) = S_{k^+}(\sigma(I))$.
  • $Q^{(k,1)}(\sigma(I)) = \{ J_j \in B^{(k)} : S_j(\sigma(I)) \geq \bar{S}_{f(I^i)}(\sigma(I)) \}$.
  • $Q^{(k,2)}(\sigma(I)) = \{ J_j \in I : r_j(I) < C_{f(I^i)}(\sigma(I)), S_j(\sigma(I)) > C_{f(I^i)}(\sigma(I)) \}$.
• For $i = 1, 2$, $\tau^{(k,i)}(\sigma(I)) = \min \{ \tau_j(I) : J_j \in Q^{(k,i)}(\sigma(I)) \}$. In the case that $Q^{(k,i)}(\sigma(I)) = \emptyset$, we just define $\tau^{(k,i)}(I, \sigma) = \infty$.
• $|Q^{(k,1)}(\sigma(I))|$ is called the SWPT-reverse index of the SWPT-reverse pair $(J_k, J_{k^+})$ in $\sigma(I)$.

**Lemma 3.3.** There is an instance-schedule pair $(I, \sigma) \in \Omega$ such that the jobs in each block are scheduled in SWPT order in $\sigma(I)$.

**Proof.** Suppose to the contrary that every instance-schedule pair $(I, \sigma) \in \Omega$ has at least one block in which the jobs are not scheduled in the SWPT order. Then every instance $I$ is a smallest counterexample such that $\sigma(I)$ has at least one SWPT-reverse pair. Let $(I, \sigma) \in \Omega$ such that the following two conditions are satisfied for an online schedule $\sigma(I)$ of $I$:
  (i) the number of SWPT-reverse pairs in $\sigma(I)$ is as small as possible, and
  (ii) subject to (i), the SWPT-reverse index of the last SWPT-reverse pair in $\sigma(I)$ is as small as possible.

Let $(J_k, J_{k^+})$ be the last SWPT-reverse pair in $\sigma(I)$. Then $\tau_j(I) = \tau_{k^+}(I) < \tau_k(I)$ for all jobs $J_j \in I^{(k)}$. From FIADSWPT and Lemma 3.2, for each job $J_j \in I^{(k)}$ we have $r_j(I) > S_k(\sigma(I))$ and $\tau_{k^+}(I) > 0$. Let $I'$ be the job instance obtained from $I^{(k)}$ by resetting $r_j(I') = r_{\min}(I^{(k)})$ for each $J_j \in I^{(k)}$, where $r_{\min}(I^{(k)}) = \min \{ r_j(I) : J_j \in I^{(k)} \}$. Then $I'$ is a regular instance in which all jobs have a common release date $r_{\min}(I^{(k)}) > S_k(\sigma(I))$ and a common Smith-ratio $\tau_{k^+}(I)$. Set $S_{k^+}(\sigma(I)) = r_L$. By Lemma 2.8, we can conclude that $Z_{f(I^i)}(I, \sigma) \leq (1 + \alpha) \cdot OPT(I')$. By Lemma 2.3, we have $OPT(I') \leq OPT(I^{(k)})$. Consequently, we have $Z_{f(I^i)}(I, \sigma) \leq (1 + \alpha) \cdot OPT(I^{(k)})$.

Now we claim that $\tau^{(k,1)}(\sigma(I)) \leq \tau^{(k,2)}(\sigma(I))$.

In fact, if $Q^{(k,1)}(\sigma(I)) \neq \emptyset$, from the implementation of FIADSWPT, we have $\tau^{(k,1)}(\sigma(I)) \leq \tau^{(k,2)}(\sigma(I))$. If $Q^{(k,1)}(\sigma(I)) = \emptyset$, then $\tau^{(k,1)}(\sigma(I)) = \infty$ and so does $\tau^{(k,2)}(\sigma(I))$. Consequently, $Q^{(k,2)}(\sigma(I)) = \emptyset$. The claim follows.

Set $\tilde{\tau} = \tau_{k^+}(I) / \min \{ \tau_k(I), \tau^{(k,1)}(\sigma(I)) \}$. Then $\tilde{\tau} < 1$. Since $\tau^{(k,2)}(\sigma(I)) \geq \tau^{(k,1)}(\sigma(I))$, each job $J_j \in I^{(k)}$ is an available job with the minimum Smith-ratio at time $S_j(\sigma(I)) = S_j(\sigma(I^{(k)}; \tilde{\tau}))$ in $\sigma(I^{(k)}; \tilde{\tau})$. Then $\sigma(I^{(k)}; \tilde{\tau})$ is a possible schedule of $I^{(k)}$ generated by FIADSWPT in which the jobs are processed in the same time intervals as that in $\sigma(I)$. From Lemma 2.6, together with the fact that $Z(I^{(k)}, \sigma) \leq (1 + \alpha) \cdot OPT(I^{(k)})$, we have $Z(I^{(k)}; \tilde{\tau}, \sigma) > (1 + \alpha) \cdot OPT(I^{(k)}; \tilde{\tau})$. Hence, $I^{(k)}; \tilde{\tau}, \sigma$ is also a smallest counterexample and so $(I^{(k)}; \tilde{\tau}, \sigma, \sigma) \in \Omega$.

If $\tilde{\tau} = \tau_{k^+}(I) / \tau_k(I)$, then $\tau_{k^+}(I^{(k)}; \tilde{\tau}) = \tau_k(I) = \tau_{k^+}(I^{(k)}; \tilde{\tau})$, and so, the number of SWPT-reverse pairs in $\sigma(I^{(k)}; \tilde{\tau})$ is less than that in $\sigma(I)$. This contradicts the assumption that $(I, \sigma)$ is chosen under condition (i).
If \( \delta = \frac{\tau_k(I)/\tau^{(k,1)}(I)}{\tau_k(I)/\tau_k(I)} > \frac{\tau_k(I)}{\tau_k(I)} \), then the number of SWPT-reverse pairs in \( \sigma(I^{(k,1)}, \delta) \) is the same as that in \( \sigma(I) \) and \( (J_k, J_{k-1}) \) is also the last SWPT-reverse pair in \( \sigma(I^{(k,1)}, \delta) \), but \( |Q^{(k,1)}(\sigma(I^{(k,1)}, \delta))| < |Q^{(k,1)}(\sigma(I))| \). This contradicts the assumption that \((I, \sigma)\) is chosen under condition (ii). The lemma follows. \[ \Box \]

Based on Lemma 3.3, we define \( \Omega_1 \) to be the set of online instance-schedule pairs \((I, \sigma)\) in \( \Omega \) such that the jobs in each block of some associated online schedule \( \sigma(I) \) are scheduled in SWPT order. Then \( \Omega_1 \) is not empty.

**Lemma 3.4.** There is an online instance-schedule pair \((I, \sigma)\) in \( \Omega_1 \) in which all jobs have a common Smith-ratio.

**Proof.** Let \((I, \sigma)\) in \( \Omega_1 \) such that the number of distinct Smith-ratios of the jobs in \( I \) is as small as possible. From the definition of \( \Omega_1 \), there is an online instance-schedule pair \((I, \sigma)\) such that the jobs in each block of \( \sigma(I) \) are scheduled in SWPT order. If possible, suppose that the jobs in \( I \) have at least two distinct Smith-ratios. Let \( \tau_{\max}(I) = \max\{\tau_j : J_j \in I\} \) and let \( T = \{J_j \in I : \tau_j(I) = \tau_{\max}(I)\} \). Then each job in \( T \) is an end-job of some block in \( \sigma(I) \).

Let \( T^* \) be the set of jobs \( J_j \in T \) such that there is some job \( J_i \in I \setminus T \) with \( S_j(\sigma(I)) < r_j(I) \leq C_j(\sigma(I)) \) and \( S_j(\sigma(I)) > C_j(\sigma(I)) \). For each job \( J_j \in T^* \), we define \( J_{j'} \) to be the candidate job at time \( C_j(\sigma(I)) \) in \( \sigma(I) \). Since FIADSWPT does not schedule \( J_{j'} \) starting at time \( C_j(\sigma(I)) \), the only possibility is that

\[
\frac{p_j}{\lambda} + \sum_{S_j(\sigma(I)) \leq C_j(\sigma(I))} \hat{p}_i(C_j(\sigma(I))) > \alpha C_j(\sigma(I))
\]

for each \( J_j \in T^* \). Let \( I' \) be the instance obtained from \( I \setminus T \) by resetting \( r_j(I') = S_j(\sigma(I)) < r_j(I) \) for each job \( J_j \in T' \). It can be observed that \( Z_{I'}(I', \sigma) = Z_{I \setminus T}(I, \sigma) \).

Since \( |I'| < |I| \) and \( I \) is a smallest counterexample, we have \( Z_{I'}(I', \sigma) \leq \lambda(\alpha) \cdot \text{OPT}(I') \). Since \( \text{OPT}(I') \leq \text{OPT}(I \setminus T) \) (from Lemma 2.3) and \( Z_{I'}(I', \sigma) = Z_{I \setminus T}(I, \sigma) \), we have \( Z_{I \setminus T}(I, \sigma) \leq \lambda(\alpha) \cdot \text{OPT}(I \setminus T) \).

Let \( \tau^* = \max\{\tau_j(I) : J_j \in I \setminus T\} \) be the second maximum Smith-ratio of the jobs in \( I \). Set \( \delta = \tau_{\max}/\tau^* \). Then \( \delta > 1 \). From Lemma 2.7, we have \( Z(I^{(T, \delta)}, \sigma) > \lambda(\alpha) \cdot \text{OPT}(I^{(T, \delta)}) \). It can be observed that \((I^{(T, \delta)}, \sigma)\) in \( \Omega_1 \). But then, the number of Smith-ratios of the jobs in \( I^{(T, \delta)} \) is less than that in \( I \). This contradicts the assumption that \( I \) has the smallest number of Smith-ratios. Hence, we may assume that \( \tau_j(I) = 1 \) and so \( w_j(I) = p_j \) for all jobs \( J_j \in I \). The lemma follows. \[ \Box \]

Based on Lemma 3.4, we define \( \Omega_2 \) to be the set of instance-schedule pairs \((I, \sigma)\) in \( \Omega_1 \) such that all jobs have a common Smith-ratio in \( I \). Then \( \Omega_2 \) is not empty.

**Lemma 3.5.** There is an online instance-schedule pair \((I, \sigma)\) in \( \Omega_2 \) such that \( I \) is a regular instance.

**Proof.** By the definition of \( \Omega_2 \), all jobs in \( \Omega_2 \) have a common Smith-ratio. Let \( s^* \) denote the starting time of the last block of \( \sigma(I) \). Let \( I_1 = \{J_j \in I : S_j(\sigma(I)) < s^*\} \) and \( I_2 = \{J_j \in I : S_j(\sigma(I)) \geq s^*\} \). We distinguish the following two cases.

**Case 1.** \( p_j/m < \alpha s^* \) for all jobs \( J_j \in I_2 \). Then the implementation of FIADSWPT implies that \( r_j \geq s^* \) for all jobs \( J_j \in I_2 \). We now claim that \( I_1 = \emptyset \).

In fact, if \( I_1 \neq \emptyset \), from the minimality of \( I \), we have \( Z(I_1, \sigma)/\text{OPT}(I_1) \leq \lambda(\alpha) \) and \( Z(I_2, \sigma)/\text{OPT}(I_2) \leq \lambda(\alpha) \). Since \( Z(I, \sigma) = Z(I_1, \sigma) + Z(I_2, \sigma) \) and \( \text{OPT}(I) = \text{OPT}(I_1) + \text{OPT}(I_2) \), from Lemma 2.2, we have \( Z(I, \sigma)/\text{OPT}(I) \leq \lambda(\alpha) \). This contradicts the assumption that \( I \) is a counterexample. The claim follows.
Thus, all jobs in $I$ have a common Smith-ratio $\tau$ and there does not exist a time $t$ in the online schedule $\sigma(I)$ such that all the machines remain idle at time $t$. From the definition of a regular instance, the above claim implies that $I$ is a regular instance.

**Case 2.** There is a job $J_j \in I_2$ such that $p_j/m \geq \alpha s^*$. We may assume that $p_j = p_{\text{max}}$. Then we can obtain a new instance $I''$ by modifying $r_j$ of all jobs to 0. Since the jobs in $I''$ have a common Smith-ratio, a possible online schedule $\sigma''(I'')$ can be obtained from FIADSWPT by the way that $J_j$ is always the candidate from time 0 until time $s^*$. Then $\sigma''(I'')$ schedules all jobs in $I''$ consecutively from time $s^*$. Since the jobs have a common Smith-ratio, we can observe that $Z(I'', \sigma'') \geq Z(I, \sigma)$. Then $I''$ is a new counterexample such that (i) all jobs in $I$ have a common release date $r = 0$ and (ii) all jobs in $I$ have a common Smith-ratio. It follows that $I''$ is a regular instance in $\Omega_2$. The lemma follows.

From Lemma 2.8 and 2.9, for a regular instance, we have $Z(I, \sigma)/\text{OPT}(I) \leq \lambda(\alpha)$. This contradicts the assumption that $I$ is a counterexample. In order to minimize the upper bound, let $1 + \alpha = \frac{1 + \sqrt{2}}{2} + \frac{1}{\alpha}$, we can obtain that $\alpha = \frac{\sqrt{2} - 1 + \sqrt{19 - 2\sqrt{2}}}{4}$. Thus the resulted upper bound is $1 + \alpha = 1 + \frac{\sqrt{2} - 1 + \sqrt{19 - 2\sqrt{2}}}{4} \approx 2.11$. Then we have the following lemma.

**Lemma 3.6.** For problem $P|\text{online}, r_j| \sum w_j C_j$, FIADSWPT is an online algorithm with a competitive ratio of at most 2.11.

Note that the schedule generated by Improved-ADSWPT is also a possible schedule generated by FIADSWPT, then we have the following final result.

**Theorem 3.7.** For problem $P|\text{online}, r_j| \sum w_j C_j$, Improved-ADSWPT is a polynomial-time online algorithm with a competitive ratio of at most 2.11.

Not providing a better proof, combining with the numerical experiments showed in [15], we give the following conjecture:

**Conjecture 1.** Algorithms Improved AD-SWPT and AD-SWPT both have a competitive ratio of 2.

**REFERENCES**

[1] E. J. Anderson and C. N. Potts, Online scheduling of a single machine to minimize total weighted completion time, *Mathematics of Operations Research*, 29 (2004), 686–697.

[2] M. C. Chou, M. Queyranne and D. Simchi-Levi, The asymptotic performance ratio of an on-line algorithm for uniform parallel machine scheduling with release dates, *Mathematical Programming*, 106 (2006), 137–157.

[3] J. R. Correa and M. R. Wagner, LP-based online scheduling: from single to parallel machines, *Mathematical Programming*, 119 (2009), 109–136.

[4] M. X. Goemans, M. Queyranne, A. S. Schulz, M. Skutella and Y. Wang, Single machine scheduling with release dates, *SIAM Journal on Discrete Mathematics*, 15 (2002), 165–192.

[5] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. R. Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics*, 5 (1979), 287–326.

[6] L. A. Hall, A. S. Schulz, D. B. Shmoys and J. Wein, Scheduling to minimize average completion time: Off-line and on-line approximation algorithms, *Mathematics of Operations Research*, 22 (1997), 513–544.

[7] T. Kawaguchi and S. Kyan, Worst case bound of an LRF schedule for the mean weighted flow-time problem, *SIAM Journal on Computing*, 15 (1986), 1119–1129.
[8] P. H. Liu and X. W. Lu, On-line scheduling of parallel machines to minimize total completion times, *Computers & Operations Research*, 36 (2009), 2647–2652.

[9] R. Ma, J. P. Tao and J. J. Yuan, Online scheduling with linear deteriorating jobs to minimize the total weighted completion time, *Applied Mathematics And Computation*, 273 (2016), 570–583.

[10] R. Ma and J. J. Yuan, Online tradeoff scheduling on a single machine to minimize makespan and total weighted completion time, *International Journal of Production Economics*, 158 (2014), 114–119.

[11] N. Megow and A. S. Schulz, On-line scheduling to minimize average completion time revisited, *Operations Research Letters*, 32 (2004), 485–490.

[12] C. Phillips, C. Stein and J. Wein, Minimizing average completion time in the presence of release dates, *Mathematical Programming*, 82 (1998), 199–223.

[13] R. Sitters, Efficient algorithms for average completion time scheduling, in *Integer Programming and Combinatorial Optimization* (eds. F. Eisenbrand and F. Shepherd), vol. 6080 of Lecture Notes in Computer Science, Springer, (2010), 411–423.

[14] W. Smith, Various optimizers for single-stage production problem, *Naval Research Logistics Quarterly*, 3 (1956), 59–66.

[15] J. P. Tao, A better online algorithm for the parallel machine scheduling to minimize the total weighted completion time, *Computers & Operations Research*, 43 (2014), 215–224.

[16] J. P. Tao, R. H. Huang and T. D. Liu, A 2.28-competitive algorithm for online scheduling on identical machines, *Journal of Industrial and Management Optimization*, 11 (2015), 185–198.

[17] A. P. A. Vensens, *On-line Machine Scheduling*, PhD thesis, Technische Universiteit Eindhoven, Eindhoven, Netherlands, 1997.

Received February 2016; revised September 2016.

E-mail address: sungirlmr@hpu.edu.cn

E-mail address: jipingtao@gmail.com