

Research Article

WSPT's Competitive Performance for Minimizing the Total Weighted Flow Time: From Single to Parallel Machines

Jiping Tao^{1,2} and Tundong Liu^{1,2}

¹ Department of Automation, Xiamen University, Xiamen, China

² Center for Cloud Computing and Big Data, Xiamen University, Xiamen, China

Correspondence should be addressed to Tundong Liu; ltd@xmu.edu.cn

Received 9 June 2013; Accepted 5 September 2013

Academic Editor: Dane Quinn

Copyright © 2013 J. Tao and T. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider the classical online scheduling problem over single and parallel machines with the objective of minimizing total weighted flow time. We employ an intuitive and systematic analysis method and show that the weighted shortest processing time (WSPT) is an optimal online algorithm with the competitive ratio of $P+1$ for the case of single machine, and it is $(P+(3/2)-(1/2m))$ -competitive for the case of parallel machines ($m > 1$), where P is the ratio of the longest to the shortest processing time.

1. Introduction

In the context of scheduling problems, the flow time of a job is the total time it stays in the system. Sometimes it is also called response time. It is equal to the delay of waiting for service plus the actual service time. The total (weighted) flow time captures the overall quality of service of the system, which is a natural and important measure in many applications such as networks and parallel computing [1, 2]. From the perspective of optimal solutions, the objective of minimizing the total (weighted) flow time is equivalent to minimizing the total (weighted) completion time. However, online algorithms and approximation results are largely different between the two objectives.

In this work, we consider the online scheduling over parallel machines with the objective of minimizing the total weighted flow time. Formally, there is a set of n jobs and m identical machines. Each job J_j is associated with processing time p_j , release date r_j , and a positive weight w_j . All the information about one job is not revealed until it is released. Also, the total number n of jobs is unknown a priori. An online algorithm determines which machine to be chosen and what time to start for processing each job without interruption. Denote the completion time and flow time of J_j by C_j and F_j , respectively. Then, $F_j = C_j - r_j$ holds. The problem can be denoted by $Pm|r_j|\sum w_j F_j$ in terms of the standard three field notations for scheduling problems [3].

When the objective is to minimize the total (weighted) completion time, it is well known that optimal deterministic online algorithms with the competitive ratio of 2 have been presented for $1|r_j|\sum C_j$ [4, 5] and $1|r_j|\sum w_j C_j$, respectively [6, 7]. As for the parallel machine scheduling $P|r_j|\sum w_j C_j$, the current best online algorithm is given by Correa and Wagner [8]; they design a 2.618-competitive algorithm based on linear programming techniques and the concept of α -point. However, no online algorithm with bounded competitive ratio exists for the case of total (weighted) flow time scheduling. Even for offline approximate algorithms, the lower bound on worst-case performance guarantee is still at least $\Omega(n^{\sqrt{1/2}-\epsilon})$ for the single machine case [9] and $\Omega(n^{\sqrt{1/3}-\epsilon})$ for the multiple machines case [2].

Since the lower bound is so strong, an intuitive idea is to impose some additional assumptions on the original problem. A widely used assumption is that the ratio of the longest to the shortest processing time is not greater than a constant, say P . With this assumption, Bunde [10] proves that the shorting processing time (SPT) rule is optimal with the competitive ratio of $(P+1)/2$ for $1|r_j|\sum F_j$. Most of other previous works focus on the preemptive setting. Chekuri et al. [11] consider the problem $1|r_j, pmtn|\sum w_j F_j$. They present a $O(\log^2 P)$ -competitive online algorithm when the weights are also bounded. Leonardi and Raz [2] prove that the shortest

remaining processing time (SRPT) algorithm has a competitive ratio of $O(\log(\min((n/m), P)))$ for $Pm|r_j, pmtn| \sum w_j F_j$. The same authors also provide a $\Omega(\log(\min((n/m), P)))$ lower bound on the competitive ratio of any randomized algorithm for this problem. Bansal and Dhamdhere [12] make another kind of assumption. They assume that the job weights belong to at most k different arbitrary weight classes. Under this assumption, they construct an online algorithm for $1|r_j, pmtn| \sum w_j F_j$ and prove that it is k -competitive. Combining this conclusion and the assumption that the ratio of the maximum to the minimum job weight is W , the authors further obtain a $O(\log W)$ -competitive algorithm, since weights can be rounded up to a power of 2.

To the best of our knowledge, no positive results exist when nontrivial weights are considered and preemption is not allowed. In this work, we show that the weighted shortest processing time (WSPT) rule is optimal with the competitive ratio of $P + 1$ for $1|r_j| \sum w_j F_j$, and it is $(P + (3/2) - (1/2m))$ -competitive for $Pm|r_j, pmtn| \sum w_j F_j$ ($m > 1$). The competitive analysis method is based on the idea of instance transformation, which is first introduced by us in [13] and is further developed in [14, 15]. The method tries to search for the worst-case instance in the instance space. It starts from an arbitrary instance and modifies the instance towards the possible structure of the worst-case instance with respect to the given online algorithm. The modification guarantees that the performance ratio of the modified instance does not decrease. Eventually, the modification ends up with an instance with a relatively simple and special structure, whose performance ratio can be much easily evaluated.

The paper is organized as follows. In Section 2 we show that the worst-case instance with respect to the WSPT rule is characterized by two kinds of special structures. Then, we analyze the performance ratio of these two kinds of instances for the single machine and parallel machines in Sections 3 and 4, respectively. Concluding remarks are given in Section 5.

2. Block Structure of the Schedule and Instance Transformation

For any instance I , denote the schedule obtained by WSPT by $\sigma(I)$, the optimal offline schedule by $\sigma^*(I)$, and the objective values of the two schedules by $ALG(I)$ and $OPT(I)$, respectively. For any job J_j in I , denote its flow time in $\sigma(I)$ and $\sigma^*(I)$ by F_j and F_j^* .

It can be easily shown that the worst-case instance can be obtained among such instances for which the schedules by WSPT rule are composed of a single block. A block means a time interval in which jobs are processed contiguously without keeping all the machines idle at the same time. Denote any one of these instances by I_1 . Without loss of generality, assume that the first job is released at time 0; that is, the single block begins at time 0. According to the nondecreasing order of starting times of jobs in $\sigma(I_1)$, we further partition jobs into subblocks, such that the jobs within each subblock are ordered according to the WSPT rule and the last job of a subblock has a greater ratio of processing time to weight (hereafter, we use weighted processing time to refer

to the ratio) than the first job of the succeeding subblock if it exists.

We will further show that a new instance with a more simple and special structure can be obtained by modifying the weights of jobs in I_1 . First we introduce an important lemma without proof since it can be derived from basic mathematics. The following lemma will be repeatedly used in the competitive analysis.

Lemma 1. *Let $f(x)$ and $g(x)$ be two positive functions defined in the interval $[u, v]$; moreover, $f(x)$ is convex and $g(x)$ is concave. Then, $f(x)/g(x)$ reaches its maximum at one endpoint of the interval; that is, $f(x)/g(x) \leq \max\{f(u)/g(u), f(v)/g(v)\}$ for all $x \in [u, v]$.*

It can be easily shown that the above lemma still holds when the interval is open at some endpoint in the condition that the limitation of $f(x)/g(x)$ exists at the corresponding endpoint.

Lemma 2. *For an arbitrary instance I_1 , a new instance can be constructed by modifying the weights of jobs in I_1 , such that all the jobs in the new instance have the same weighted processing time or there are jobs which not only have the same weighted processing time but also have positive infinite weights in the last subblock of the schedule for the new instance; denote the former new instance by I_2 and the latter one by I_3 ; then*

$$\frac{ALG(I_1)}{OPT(I_1)} \leq \max \left\{ \frac{ALG(I_2)}{OPT(I_2)}, \frac{ALG(I_3)}{OPT(I_3)} \right\}. \quad (1)$$

Proof. Assume that $\sigma(I_1)$ is composed of K subblock. We first consider the case of $K > 1$. Next, we will show that an intermediate instance I' can be obtained by modifying I_1 , such that either $\sigma(I')$ is composed of $K - 1$ subblocks or there are jobs with positive infinite weights in the last subblock of $\sigma(I')$ with K subblocks.

Denote the last job of the next-to-last subblock in $\sigma(I_1)$ by J_k . Denote the set of all the jobs of the last subblock by Q' . Divide Q' into two subsets in terms of the relations between their weighted processing times and J_k 's:

$$Q'_1 = \left\{ J_j \mid J_j \in Q', 0 < \frac{p_j}{w_j} < \frac{p_k}{w_k} \right\}, \quad (2)$$

$$Q'_2 = \left\{ J_j \mid J_j \in Q', \frac{p_j}{w_j} \geq \frac{p_k}{w_k} \right\}.$$

Modify the weight w_j of each job J_j in Q'_1 to δw_j , where δ is a parameter to be chosen later. Let

$$\underline{\delta} = \frac{\max \left\{ (p_j/w_j) \mid J_j \in Q'_1 \right\}}{p_k/w_k}. \quad (3)$$

Denote the intermediate instance after this adjustment by I' . Since this adjustment does not change the mutual relationships of the weighted processing time among jobs in the last subblock, jobs are processed in $\sigma(I')$ in the same order as in $\sigma(I_1)$ for any $\delta \in [\underline{\delta}, +\infty)$.

$ALG(I')$ is composed of the linear combination of all jobs' weights, where the coefficient of each weight is the corresponding job's flow time. Since the schedule does not change after adjustment and each job's flow time does not change, $ALG(I')$ is a monotonously increasing linear function with respect to δ . Consider how $OPT(I')$ changes with respect to δ . If jobs keep the same processing order as in $\sigma^*(I_1)$, the objective value of the obtained schedule is also a monotonously increasing linear function with respect to δ . Since the optimal schedule $\sigma^*(I')$ is the one with the minimal objective value among all the feasible schedules, $OPT(I')$ is a piecewise linear function with respect to δ ; moreover, its slope does not increase with δ increasing. It means that $OPT(I')$ is a concave function with respect to δ . According to Lemma 1, we can obtain an instance with a worse performance ratio by choosing δ as either $\underline{\delta}$ or $+\infty$. There are two cases.

- (i) If δ is chosen as $\underline{\delta}$, there is at least one job belonging to Q'_1 whose weighted processing time is modified to p_k/w_k according to (3). Then, update Q'_1 and Q'_2 according to (2). If there remain jobs in Q'_1 , modify their weights by repeating the above procedure; otherwise all the jobs in Q'_1 have been removed into Q'_2 . Since all the jobs in Q'_2 have greater weighted processing time than J_k 's, the last two subblocks in the original schedule $\sigma(I_1)$ can be combined into a single subblock in the new instance according to the description of "subblock" in Section 2; that is, $\sigma(I')$ is composed of $K - 1$ subblocks.
- (ii) If δ tends to infinity, all the jobs in Q'_1 have weights $+\infty$. This intermediate instance is just the instance which we want to obtain at the beginning of the proof.

By repeatedly applying the above transformation procedure, we can eventually obtain a new instance, such that the schedule by WSPT for the new instance is composed of a single subblock, or there are jobs with positive infinite weights in the last subblock of the schedule for the new instance. For the former one, for ease of exposition, we still use I' to denote it. Next we will show that I' can be modified such that all the jobs have the same weighted processing time.

Denote the first job in the single subblock of $\sigma(I')$ by J_1 . Divide all the jobs into other two sets in terms of the relations between their weighted processing time and J_1 's:

$$\begin{aligned}
 Q''_1 &= \left\{ J_j \mid J_j \in Q'_2, +\infty > \frac{p_j}{w_j} > \frac{p_1}{w_1} \right\}, \\
 Q''_2 &= \left\{ J_j \mid J_j \in Q'_2, \frac{p_j}{w_j} = \frac{p_1}{w_1} \right\}.
 \end{aligned} \tag{4}$$

Similar to the previous adjustment, modify the weight w_j of each job J_j in Q''_1 to δw_j , where δ is a parameter to be chosen later. Let

$$\bar{\delta} = \frac{\min \left\{ (p_j/w_j) \mid J_j \in Q''_1 \right\}}{p_1/w_1}. \tag{5}$$

We use \bar{I}' to denote the intermediate instance after this adjustment. Completely similar to the previous analysis, when δ lies in $(0, \bar{\delta}]$, $ALG(\bar{I}')$ is a monotonously increasing linear function with respect to δ . $OPT(\bar{I}')$ is a concave function with respect to δ . According to Lemma 1, we can obtain an instance with a worse performance ratio by choosing δ as either $\bar{\delta}$ or an infinitely small value ε . If δ is chosen as $\bar{\delta}$, there is at least one job belonging to Q''_1 whose weighted processing time is modified to p_1/w_1 according to (5). Then, update Q''_1 and Q''_2 according to (4). If there remain jobs in Q''_1 , modify their weights by repeating the above adjustment procedure. If δ is chosen as an infinitely small value, all the jobs in Q''_1 have weights tending to 0. These jobs are processed at last both in $\sigma(\bar{I}')$ and $\sigma^*(\bar{I}')$. Thus, we can directly delete these jobs, since they contribute nothing to the objective value.

After finite steps, all the jobs lie in Q''_2 ; that is, all the jobs have the same weighted processing time as J_1 's. It is just the instance I_2 which we want to obtain in the theorem. For the intermediate instance, where there are jobs with positive infinite weights in the last subblock, we can apply the same above modification and modify these jobs such that they have the same weighted processing time. The resulted instance is I_3 which we want to obtain in the theorem. \square

3. WSPT's Optimality for $1|r_j| \sum w_j F_j$

In this section, we show that WSPT is optimal with the competitive ratio of $P + 1$ for $1|r_j| \sum w_j F_j$. We obtain the result by proving that $P + 1$ is both an upper bound on the competitive ratio of WSPT and a lower bound on the competitive ratio of any online algorithm.

We know that the worst-case instance can be achieved among such instances, for which the schedules by WSPT are composed of a single block. From Lemma 2, we show that any such instance can be modified to a new instance with a worse performance ratio. Next, we can directly analyze the performance ratio of the new instance obtained in Lemma 2 and get the competitive performance of WSPT algorithm.

Theorem 3. *The WSPT algorithm is optimal with the competitive ratio of $P + 1$ for $1|r_j| \sum w_j F_j$.*

Proof. For the new instance I_2 described in Lemma 2, where all the jobs have the same weighted processing time, it is obvious that the schedule obtained by WSPT is just the optimal one for $1|r_j| \sum w_j F_j$.

As for another new instance I_3 , there are jobs which not only have the same weighted processing time but also have positive infinite weights in the last subblock. Denote the last job of the next-to-last subblock by J_k . Denote the set of all these jobs with positive infinite weights in the last subblock by Q^∞ . We can directly analyze the performance ratio of the instance by only considering these jobs in Q^∞ since other jobs' weighted flow times can be omitted in the sense of limitation. All these jobs in Q^∞ are released after J_k starts being processed in $\sigma(I_3)$; otherwise they would be processed before J_k . These jobs are continuously processed immediately after J_k is finished in $\sigma(I_3)$. They are processed in the optimal

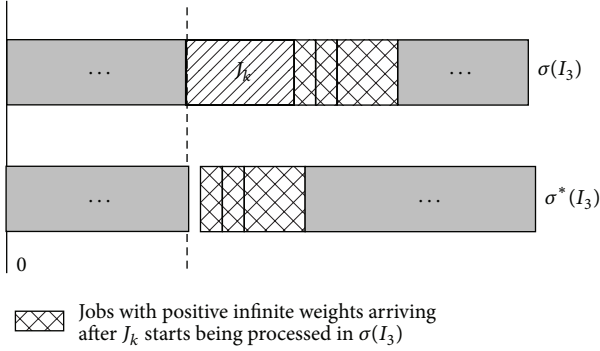


FIGURE 1: The WSPT schedule and optimal schedule for I_3 in the case of single machine.

schedule $\sigma^*(I_3)$ in the same order as in $\sigma(I_3)$; moreover, they start being processed not earlier than the time when J_k starts in $\sigma(I_3)$. Figure 1 illustrates the schedules $\sigma(I_3)$ and $\sigma^*(I_3)$. Thus, we have

$$\begin{aligned}
 \frac{\text{ALG}(I_3)}{\text{OPT}(I_3)} &= \frac{\sum_{j \in Q^\infty} w_j F_j}{\sum_{j \in Q^\infty} w_j F_j^*} \\
 &\leq \max_{j \in Q^\infty} \frac{F_j}{F_j^*} \\
 &\leq \max_{j \in Q^\infty} \frac{F_j^* + p_k}{F_j^*} \quad (6) \\
 &\leq 1 + \max \frac{p_k}{p_j} \\
 &\leq 1 + P.
 \end{aligned}$$

So, we get an upper bound on the competitive ratio of WSPT algorithm, i.e.,

$$\rho \leq 1 + P. \quad (7)$$

Next, we show that $1 + P$ is also a lower bound on the competitive ratio of any deterministic online algorithm for $1|r_j|\sum w_j F_j$.

Consider the following instance. The first job J_1 arrives at time 0 with $p_1 = 1$ and $w_1 = 1$. The online algorithm decides to schedule J_1 at time S . The second job J_2 is released job at time $S + \varepsilon$ with $p_2 = 1/P$ and w_2 , where ε is an infinitely small value. The optimal schedule performs no worse than the one which first schedules J_2 at time $S + \varepsilon$, then, followed by J_1 . So we have

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \geq \frac{S + 1 + w_2(1 + 1/P - \varepsilon)}{S + \varepsilon + 1/P + 1 + w_2(1/P)}. \quad (8)$$

When ε tends to 0 and w_2 tends to infinity, the above ratio tends to $(P + 1)$. Since a competitive ratio of an online algorithm is the maximal performance ratio over all the instances, we can directly obtain that the competitive ratio of any deterministic online algorithm is no less than $(P + 1)$.

According to the above analysis, we can immediately obtain that the WSPT algorithm is optimal with the competitive ratio of $P + 1$. \square

4. Competitive Analysis of WSPT Algorithm for $Pm|r_j|\sum w_j F_j$

Next, we will extend the result for the single machine in Section 3 to the case of parallel machines. Similar to the proof for Theorem 3, we can analyze the performance ratio of the new instances I_2 and I_3 obtained in Lemma 2. First, an appropriate lower bound on the optimal schedule has to be established in order to further analyze the performance ratio of I_2 and I_3 . Chou et al. [16] show a lower bound on the corresponding weighted completion time problem of $Pm|r_j|\sum w_j C_j$ by introducing a virtual m -times faster single machine problem and the concept of LP schedule [17]. We can readily extend this result to the weighted flow time problem due to the equivalence between optimal schedules for $Pm|r_j|\sum w_j C_j$ and $Pm|r_j|\sum w_j F_j$.

Corollary 4. For an instance I of $P|r_j|\sum w_j F_j$, construct a virtual m -times faster single machine instance $I' = \{r_j, p_j/m, w_j\}$. For the corresponding LP schedule of I' , denote the mean-busy-time and flow time of job J_j by M_j^{LP} and F_j^{LP} , respectively, and then,

$$\text{OPT}(I) \geq \sum_{j \in I} w_j M_j^{LP} + \frac{1}{2} \sum_{j \in I} w_j p_j - \sum_{j \in I} w_j r_j. \quad (9)$$

Moreover, if the LP schedule is nonpreemptive, then,

$$\text{OPT}(I) \geq \sum_{j \in I} w_j F_j^{LP} + \frac{1}{2} \left(1 - \frac{1}{m}\right) \sum w_j p_j. \quad (10)$$

Hereafter, we refer to the lower bound in Corollary 4 as LP lower bound and denote it by $\text{LB}^m(\cdot)$.

4.1. Performance Analysis of I_2 and I_3 . For simplicity, we give the result via three Lemmas.

Lemma 5. For any instance I for $Pm|r_j|\sum w_j F_j$, if all the jobs have the same weighted processing time and there is no idle time on each machine before it completes its processing in the corresponding schedule $\sigma(I)$, then,

$$\frac{\text{ALG}(I)}{\text{OPT}(I)} \leq \frac{3}{2} - \frac{1}{2m}. \quad (11)$$

Proof. All the jobs in the instance I have the same weighted processing time. Moreover, there is no idle time on each machine before it completes its processing in the corresponding schedule in $\sigma(I)$. We can find out that for the corresponding virtual m -times faster single machine problem, the schedule which processes jobs without preemption and delay in the same starting order as in $\sigma(I)$ is just a LP schedule. Without loss of generality, we can assume that $w_j = p_j$ in

order to simplify the analysis. Thus, the LP lower bound in (10) can be rewritten as

$$\text{OPT}(I) \geq \sum p_j F_j^{\text{LP}} + \frac{1}{2} \left(1 - \frac{1}{m}\right) \sum p_j^2. \quad (12)$$

Another trivial lower bound on $\text{OPT}(I)$ is

$$\text{OPT}(I) \geq \sum w_j p_j = \sum p_j^2. \quad (13)$$

Taking how the LP schedule is constructed, we can bound the flow time F_j in the WSPT schedule as

$$F_j \leq F_j^{\text{LP}} + \left(1 - \frac{1}{m}\right) p_j. \quad (14)$$

So

$$\text{ALG}(I) \leq \sum p_j F_j^{\text{LP}} + \left(1 - \frac{1}{m}\right) \sum p_j^2. \quad (15)$$

Denote the two bounds in (12) and (13) by LB1 and LB2 respectively. Next, we analyze the performance ratio by two cases.

- (i) If $\text{LB1} \leq \text{LB2}$, that is, $\sum p_j F_j^{\text{LP}} + (1/2)(1 - (1/m)) \sum p_j^2 \leq \sum p_j^2$, then

$$\begin{aligned} \frac{\text{ALG}(I)}{\text{OPT}(I)} &\leq \frac{\sum p_j F_j^{\text{LP}} + (1 - (1/m)) \sum p_j^2}{\sum p_j^2} \\ &\leq \frac{\sum p_j^2 + (1/2)(1 - (1/m)) \sum p_j^2}{\sum p_j^2} \\ &= \frac{3}{2} - \frac{1}{2m}. \end{aligned} \quad (16)$$

- (ii) If $\text{LB1} \geq \text{LB2}$, that is, $\sum p_j F_j^{\text{LP}} + (1/2)(1 - (1/m)) \sum p_j^2 \geq \sum p_j^2$,

$$\begin{aligned} \frac{\text{ALG}(I)}{\text{OPT}(I)} &\leq \frac{\sum p_j F_j^{\text{LP}} + (1 - (1/m)) \sum p_j^2}{\sum p_j F_j^{\text{LP}} + (1/2)(1 - (1/m)) \sum p_j^2} \\ &= 1 + \frac{(1/2)(1 - (1/m)) \sum p_j^2}{\sum p_j F_j^{\text{LP}} + (1/2)(1 - (1/m)) \sum p_j^2} \\ &\leq 1 + \frac{(1/2)(1 - (1/m)) \sum p_j^2}{\sum p_j^2} \\ &= \frac{3}{2} - \frac{1}{2m}. \end{aligned} \quad (17)$$

□

Lemma 6. In the case of parallel machines, for the instance I_2 obtained in Lemma 2, one has

$$\frac{\text{ALG}(I_2)}{\text{OPT}(I_2)} \leq P + \frac{3}{2} - \frac{1}{2m}. \quad (18)$$

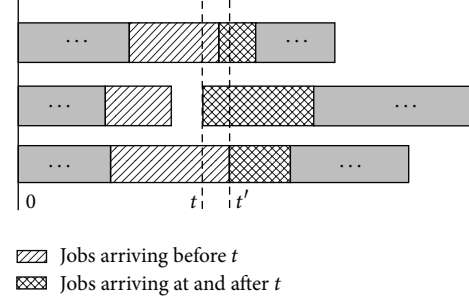


FIGURE 2: The schedule obtained by WSPT for I_2 in the case of parallel machines.

Proof. When there is no idle time on each machine before it completes its processing in $\sigma(I_2)$, this result can be directly obtained from Lemma 5. Suppose that there are totally L ($L > 1$) periods of idle time in $\sigma(I_2)$. Next, we will show that an intermediate instance I'_2 can be obtained such that there are $L - 1$ periods of idle time in $\sigma(I'_2)$; moreover,

$$\frac{\text{ALG}(I_2)}{\text{OPT}(I_2)} \leq \max \left\{ \frac{\text{ALG}(I'_2)}{\text{OPT}(I'_2)}, P + \frac{3}{2} - \frac{1}{2m} \right\}. \quad (19)$$

For the sake of exposition, we take an instance with three machines; for example, see Figure 2. t is the ending time of the last (with respect to the ending point of the idle time) period of idle time. It means that all the jobs are continuously processed without idle time on each machine after t . Since WSPT is a no-waiting strategy, we know that all the arrived jobs have been finished or are being processed at t . t' is the latest completion time of jobs which start before t . Denote the maximal processing time among all the jobs by p_{\max} ; then, $t' - t \leq p_{\max}$.

Construct an intermediate instance denoted by I'_2 , which is composed of all the jobs released before t . In addition, we construct an auxiliary instance \tilde{I}_2 . It includes all the jobs released at and after t . We have

$$\begin{aligned} \frac{\text{ALG}(I_2)}{\text{OPT}(I_2)} &\leq \frac{\text{ALG}(I'_2) + \sum_{j \in \tilde{I}_2} w_j F_j}{\text{OPT}(I'_2) + \text{OPT}(\tilde{I}_2)} \\ &\leq \max \left\{ \frac{\text{ALG}(I'_2)}{\text{OPT}(I'_2)}, \frac{\sum_{j \in \tilde{I}_2} w_j F_j}{\text{OPT}(\tilde{I}_2)} \right\}. \end{aligned} \quad (20)$$

In order to bound the later term in the above expression, similar to the proof of Lemma 5, consider the virtual m -times faster single machine problem for \tilde{I}_2 . Assume that jobs are continuously processed on the virtual m -times faster single machine from time t in the same starting order as in $\sigma(I_2)$ without considering the release time constrain. Denote the corresponding flow time of J_j by F'_j . Then, we can bound the flow time in the WSPT schedule as

$$F_j \leq F'_j + \left(1 - \frac{1}{m}\right) p_j + t' - t, \quad j \in \tilde{I}_2. \quad (21)$$

Without loss of generality, let $w_j = p_j$; we have

$$\begin{aligned} \sum_{j \in \tilde{I}_2} w_j F_j &\leq \sum_{j \in \tilde{I}_2} w_i F_j' \\ &+ \left(1 - \frac{1}{m}\right) \sum_{j \in \tilde{I}_2} p_j^2 + (t' - t) \sum_{j \in \tilde{I}_2} p_j. \end{aligned} \quad (22)$$

In addition, for the virtual m -times faster single machine problem for \tilde{I}_2 , consider its LP schedule. Denote by F_j^{LP} the flow time in the LP schedule. We can get a lower bound on $\text{OPT}(\tilde{I}_2)$ as

$$\begin{aligned} \text{OPT}(\tilde{I}_2) &\geq \sum_{j \in \tilde{I}_2} w_j F_j^{\text{LP}} + \frac{1}{2} \left(1 - \frac{1}{m}\right) \sum_{j \in \tilde{I}_2} w_j p_j \\ &\geq \sum_{j \in \tilde{I}_2} p_j F_j' + \frac{1}{2} \left(1 - \frac{1}{m}\right) \sum_{j \in \tilde{I}_2} p_j^2. \end{aligned} \quad (23)$$

Another trivial lower bound on $\text{OPT}(\tilde{I}_2)$ is

$$\text{OPT}(\tilde{I}_2) \geq \sum_{j \in \tilde{I}_2} w_j p_j = \sum_{j \in \tilde{I}_2} p_j^2. \quad (24)$$

Combining (22), (23), and (24), we have

$$\begin{aligned} &\frac{\sum_{j \in \tilde{I}_2} w_j F_j}{\text{OPT}(\tilde{I}_2)} \\ &\leq \frac{\sum_{j \in \tilde{I}_2} p_j F_j' + (1 - (1/m)) \sum_{j \in \tilde{I}_2} p_j^2 + (t' - t) \sum_{j \in \tilde{I}_2} p_j}{\max \left\{ \sum_{j \in \tilde{I}_2} p_j^2, \sum_{j \in \tilde{I}_2} p_j F_j' + (1/2) (1 - (1/m)) \sum_{j \in \tilde{I}_2} p_j^2 \right\}}. \end{aligned} \quad (25)$$

Similar to the analysis of the two cases in the proof of Lemma 5, we can obtain

$$\begin{aligned} &\frac{\sum_{j \in \tilde{I}_2} p_j F_j' + (1 - (1/m)) \sum_{j \in \tilde{I}_2} p_j^2}{\max \left\{ \sum_{j \in \tilde{I}_2} p_j^2, \sum_{j \in \tilde{I}_2} p_j F_j' + (1/2) (1 - (1/m)) \sum_{j \in \tilde{I}_2} p_j^2 \right\}} \\ &\leq \frac{3}{2} - \frac{1}{2m}. \end{aligned} \quad (26)$$

In addition, it can be obtained from $t' - t \leq p_{\max}$ that

$$\frac{(t' - t) \sum_{j \in \tilde{I}_2} p_j}{\sum_{j \in \tilde{I}_2} p_j^2} \leq \max_{j \in \tilde{I}_2} \frac{p_{\max}}{p_j} \leq P. \quad (27)$$

According to the above analysis, we can obtain

$$\frac{\sum_{j \in \tilde{I}_2} w_j F_j}{\text{OPT}(\tilde{I}_2)} \leq P + \frac{3}{2} - \frac{1}{2m} \quad (28)$$

and prove relation (19).

By repeating the above transformation, we can get a sequence of intermediate instances with (19) satisfied and finish the proof. \square

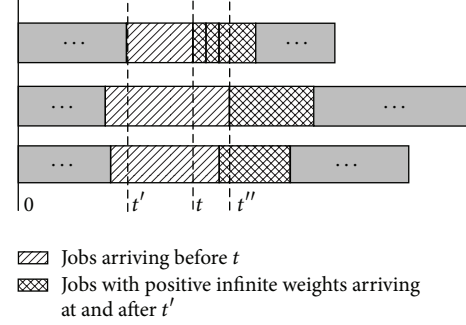


FIGURE 3: The schedule obtained by WSPT for I_3 in the case of parallel machines.

Lemma 7. *In the case of parallel machines, for the instance I_3 obtained in Lemma 2, one has*

$$\frac{\text{ALG}(I_3)}{\text{OPT}(I_3)} \leq P + \frac{3}{2} - \frac{1}{2m}. \quad (29)$$

Proof. For ease of presentation, we still take an instance with three machines; for example, see Figure 3. Assume that these jobs with positive infinite weights start being processed at the time t . t' is the latest starting time of jobs among all the machines before t . t'' is the latest completion time of jobs which start before t . Denote the maximal processing time among all the jobs by p_{\max} ; then, $t'' - t' \leq p_{\max}$. We only need to consider these jobs with positive infinite weights in the last subblock. Then, the proof is similar to the one for Lemma 6 and the detailed procedure is omitted. \square

Following Lemmas 2, 6, and 7, we can immediately get the competitive performance of WSPT for the case of parallel machines.

Theorem 8. *The WSPT algorithm is $(P + (3/2) - (1/2m))$ -competitive for $Pm|r_j| \sum w_j F_j (m > 1)$.*

5. Concluding Remarks

In this work, we consider the single and parallel machines online scheduling problem of minimizing the total weighted flow time. We show that WSPT rule is optimal with the competitive ratio of $P + 1$ for the single machine problem, where P is the ratio of the longest to the shortest processing time. We further prove that the WSPT rule is $(P + (3/2) - (1/2m))$ -competitive for the case of parallel machines ($m > 1$), which is consistent with the case of single machine problem.

In the competitive analysis, we introduce an intuitive and systematic method. The method is aiming to derive an upper bound on the competitive ratio of the online algorithm. The analysis method exploits the possible structure of the worst-case instance with respect to the given online algorithm. The basic idea behind it is to begin with an arbitrary instance and transform it by modifying the instance, such that the modified instance shows a more special structure of which we can take advantage to analyze the performance ratio. The

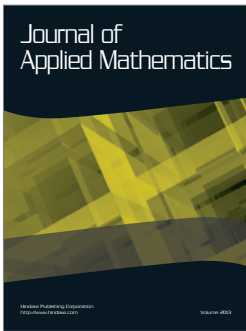
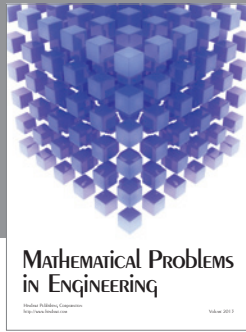
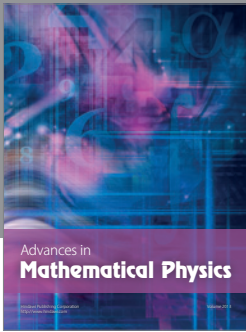
analysis method deserves to be extended to other online algorithms in our further work.

Acknowledgments

The authors would like to thank Junjie Zhou and Ye Tao for the useful discussion. This work is supported by the National Science Foundation of China (no. 11201391 and no. 61203176) and Fujian Natural Science Foundation of China (no.2013J01103).

References

- [1] S. Leonardi, "A simpler proof of preemptive total flow time approximation on parallel machines," in *Efficient Approximation and Online Algorithms*, vol. 3484 of *Lecture Notes in Computer Science*, pp. 203–212, 2006.
- [2] S. Leonardi and D. Raz, "Approximating total flow time on parallel machines," *Journal of Computer and System Sciences*, vol. 73, no. 6, pp. 875–891, 2007.
- [3] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.
- [4] J. A. Hoogeveen and A. P. A. Vestjens, "Optimal on-line algorithms for single-machine scheduling," in *Integer Programming and Combinatorial Optimization*, vol. 1084 of *Lecture Notes in Computer Science*, pp. 404–414, 1996.
- [5] C. Phillips, C. Stein, and J. Wein, "Minimizing average completion time in the presence of release dates," *Mathematical Programming*, vol. 82, no. 1-2, pp. 199–223, 1998.
- [6] E. J. Anderson and C. N. Potts, "Online scheduling of a single machine to minimize total weighted completion time," *Mathematics of Operations Research*, vol. 29, no. 3, pp. 686–697, 2004.
- [7] X. Lu, R. A. Sitters, and L. Stougie, "A class of on-line scheduling algorithms to minimize total completion time," *Operations Research Letters*, vol. 31, no. 3, pp. 232–236, 2003.
- [8] J. R. Correa and M. R. Wagner, "Lp-based online scheduling: from single to parallel machines," in *Integer Programming and Combinatorial Optimization*, vol. 3509 of *Lecture Notes in Computer Science*, pp. 196–209, 2005.
- [9] H. Kellerer, T. Tautenhahn, and G. J. Woeginger, "Approximability and nonapproximability results for minimizing total flow time on a single machine," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1155–1166, 1999.
- [10] D. P. Bunde, "SPT is optimally competitive for uniprocessor flow," *Information Processing Letters*, vol. 90, no. 5, pp. 233–238, 2004.
- [11] C. Chekuri, S. Khanna, and A. Zhu, "Algorithms for minimizing weighted flow time," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 84–93, 2001.
- [12] N. Bansal and K. Dhamdhere, "Minimizing weighted flow time," *ACM Transactions on Algorithms*, vol. 3, no. 4, pp. 39.1–39.14, 2007.
- [13] J. Tao, Z. Chao, and Y. Xi, "A novel way to analyze competitive performance of online algorithms," in *Proceedings of the IAENG International Conference on Computer Science*, pp. 592–597, 2009.
- [14] J. P. Tao, Z. J. Chao, and Y. G. Xi, "A semi-online algorithm and its competitive analysis for a single machine scheduling problem with bounded processing times," *Journal of Industrial and Management Optimization*, vol. 6, no. 2, pp. 269–282, 2010.
- [15] J. P. Tao, Z. J. Chao, Y. G. Xi, and Y. Tao, "An optimal semi-online algorithm for a single machine scheduling problem with bounded processing time," *Information Processing Letters*, vol. 110, no. 8-9, pp. 325–330, 2010.
- [16] 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*, vol. 106, no. 1, pp. 137–157, 2006.
- [17] M. X. Goemans, M. Queyranne, A. S. Schulz, M. Skutella, and Y. Wang, "Single machine scheduling with release dates," *SIAM Journal on Discrete Mathematics*, vol. 15, no. 2, pp. 165–192, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

