Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 19, No. 3, pp. 373–390, 2023

ONLINE SCHEDULING FOR DEGRADATION DATA PROCESSING ON A SINGLE PROCESSOR*

Sainan Guo, Yannan Chen, Vincent Chau, Yaping Mao and Xiaoyan Zhang†

Abstract: In pace with the arrival of the big data era, the topic of processing massive data well and quickly has attracted more and more attention over recent years. As we know, scheduling has made an important contribution in the data processing. Noticeably, the more data is accumulated, the longer it takes to process them. In the paper, in consideration of the arbitrariness of data arrival time, we address a class of online scheduling problems for degradation data processing on a single processor. Specifically speaking, there is a great quantity of data reaching online over time, which will be handled on a single processor. In our models, the goals are to minimize the makespan and minimize the maximum delivery time. Most notably, we provide optimal online algorithms for both of them.

Key words: *scheduling, degradation data processing, online algorithm, single processor*

Mathematics Subject Classification: *90B35, 68W27, 68M20*

1 Introduction

With the arrival of the era of big data, the Internet is releasing vast amounts of data. Furthermore, the more the accumulated data, the longer the processing time. As a result, a natural problem is how to deal with these data well and quickly. To our knowledge, scheduling plays a substantial role in this aspect. For the convenience of description, we can regard the processing data model as the model of processing jobs. Especially, in this paper, the data can be viewed as the job. In our study, we will use the language of processing jobs to describe and analyze the problems we focus on.

In the wake of the prosperity of scheduling theory, the papers regarding scheduling have come out in an unending flow (see [2, 7, 12, 23]). Unquestionably, online scheduling, one of the indispensable parts of scheduling, has attracted the attention of numerous scholars [6, 16, 21]. By comparison with off-line scheduling, the scheduler knows nothing about the unreached jobs in online scheduling. Nevertheless, based on the known information, the scheduler is forced to make an enormous number of decisions. On account of the information

*This work was supported by the National Natural Science Foundation of China (Nos. 12271259, 12171168, 12061059, 11971349), Guangdong Basis and Applied Basic Research Foundation (Nos. 2021A1515012032, 2021A1515010368, 2022A1515011123), the Natural Science Foundation of the Jiangsu Higher Education Institutations of China (No. 21KJB110018), the Fundamental Research Funds for the Central Universities (No. 2242022R10024), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No.1812000024761).

†Corresponding author.

deficiency, for every online algorithm, its optimality cannot be guaranteed. Thus, one cannot resist asking:

What is the evaluation criterion for the quality of an online algorithm?

As far as we know, competitive ratio [9] is making an extremely important contribution in this aspect. Let $\mathscr{A}$ be a feasible online algorithm for problem $\mathcal{P}$, where $\mathcal{P}$ is an online minimization problem. For a given instance I, suppose that $G^{\mathscr{E}}(I)$ and $G^{\mathscr{E}^*}(I)$ indicate the objective value of the schedule produced by algorithm $\mathscr{A}$ and the optimal off-line algorithm, respectively. Define

$$\rho(\mathscr{A}) = \inf\left\{\rho \mid \frac{G^{\mathscr{E}}(I)}{G^{\mathscr{E}^*}(I)} \leq \rho \text{ and } I \text{ meets } G^{\mathscr{E}^*}(I) > 0\right\}$$

as the *competitive ratio* of algorithm $\mathscr{A}$. It is important to notice that, for every feasible online algorithm $\mathscr{A}_1$, if $\rho(\mathscr{A}_1) \geq \rho(\mathscr{A})$ holds, then $\mathscr{A}$ is referred to as *optimal*.

Generally speaking, the classical scheduling problems assume that the processing time of every job is constant. Whereas, the processing times may be variable in reality. One of the common situations is that every job gets deteriorated in the waiting process, which could be found in steel production, maintenance, and fire fighting, etc. For instance, in the manufacture of porcelain, the key is to model the raw material in the light of the design. Notably, the raw material is made of china clay, which will get harder and harder with the passage of time. Furthermore, the harder the raw material becomes, the more time will be spent in modeling. Such a phenomenon is precisely the *deterioration* or *degradation*. Obviously, scheduling with deterioration is more realistic, and thus, it is meaningful to address such problems.

Problem formulation. In the paper, there is a group of jobs reaching online over time. More specifically, the concerning information of every job J_l, including the release date $r_l \geq t_0$, the processing deterioration rate $\beta_l \geq 0$, the processing time $p_l \geq 0$, the transport deterioration rate $\alpha_l \geq 0$ and the transportation time $q_l \geq 0$, is not divulged until time r_l, where $t_0 \geq 0$. Noticeably, the processor can only handle one job in unison, and the interruption is not permitted. Our goals are scheduling the jobs so as to minimize the makespan and minimize the maximum delivery time. Define S_l and C_l as the starting time and the completion time of job J_l, respectively. By applying the notation in [10], the problems we focus on can be formulated as:

$\mathcal{P}_1$: $1|\text{online}, r_l \geq t_0, p_l = \beta_l(E + FS_l)|C_{\max}$;

$\mathcal{P}_2$: $1|\text{online}, r_l \geq t_0, p_l = \beta_l(E + FS_l), q_l|D_{\max}$;

$\mathcal{P}_3$: $1|\text{online}, r_l, q_l = \alpha_l(E + FC_l)|D_{\max}$.

Note that, $C_{\max} = \max_{1\leq l\leq n} C_l$ denotes the makespan, and $D_{\max} = \max_{1\leq l\leq n} D_l$ indicates the maximum delivery time among all jobs, where $C_l = S_l + p_l$, and $D_l = C_l + q_l$ is the delivery time of job J_l. More notably, for problems $\mathcal{P}_1$ and $\mathcal{P}_2$, if $E = 0$ and $t_0 = 0$, then the completion times may be 0, which is too trivial. Thus, we may assume without loss of generality that $t_0 > 0$ follows when $E = 0$. Besides, for problems $\mathcal{P}_1$ and $\mathcal{P}_3$, the condition that $E \geq 0$ and $F \geq 0$ are constants meeting $E + F > 0$ is respected. For problem $\mathcal{P}_2$, the conditions that $E \geq 0$, $F > 0$ and $\frac{E}{F} \to 0$ hold, where E and F are constants.

Related work. Scheduling with deterioration, to our knowledge, was first put forward by Browne and Yechiali [1] and Gupta and Gupta [11] independently. From then on, such problems have received a great quantity of attentions. Meanwhile, there have been more and more scholars showing solicitude for the time-dependent models with various objective functions [5, 15, 19, 24]. Now, we mainly review a few of the most related work with the goal of minimizing the makespan. In 1994, for problem $1|r_l = t_0, p_l = \beta_l S_l|C_{\max}$, Mosheiov [20] pointed out that the objective value is independent of the processing order. Later, Cheng and Ding [4] testified that problem $1|r_l, p_l = a_l + \beta S_l|C_{\max}$ is strongly NP-complete, where a_l stands for the normal processing time of job J_l. For problem $1|p_l = a_l + f(t)|C_{\max}$, Wang et al. [22] claimed that, for all $t' < t''$, if $f(t') \geq f(t'')$ and $f(t') + t' \leq f(t'') + t''$ follow, then there is an optimal schedule without processor idleness. Compared with the flourishing of off-line scheduling research, there are relatively fewer results on online scheduling. For problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l|C_{\max}$, Liu et al. [17] attested that the Greedy algorithm is optimal.

In most situations, the jobs need to be transported to the customer or the depository after processing, and the transportation of every job will take some time, which implies that the problem with *delivery time* is closely related to the reality. As we know, minimizing the maximum delivery time is one of the significant objective functions in scheduling. And hence, there has been a generous amount of scholars paying attention to such problems [8, 14, 18]. For problem $1|\text{online}, r_l|D_{\max}$, Hoogeveen and Vestjens [13] offered an optimal online algorithm DLDT (Delayed Largest Delivery Time) with a competitive ratio of $\frac{\sqrt{5}+1}{2}$. Note that, over recent years, for the online scheduling with deterioration and delivery time, some results have emerged. For problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l = \alpha_l C_l|D_{\max}$, Liu et al. [17] attested that no online algorithm $\mathscr{A}$ satisfies $\rho(\mathscr{A}) < \max\{1 + \beta_{\max}, 1 + \alpha_{\max}\}$, where $\beta_{\max} = \max_{1\leq l\leq n} \beta_l$ and $\alpha_{\max} = \max_{1\leq l\leq n} \alpha_l$. In addition, they proved that the Greedy algorithm is optimal. Subsequently, Chai et al. [3] addressed problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l = \alpha_l S_l|D_{\max}$, and proposed an optimal online algorithm. Furthermore, they designed an optimal $\frac{1+2\beta_{\max}}{1+\beta_{\max}}$-competitive online algorithm for problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l|D_{\max}$. Besides, for problem $1|\text{online}, r_l, q_l = \alpha_l C_l|D_{\max}$, when $\alpha_{\max} > 1$, they provided an optimal algorithm with competitive ratio 2; when $\alpha_{\max} \leq 1$, an optimal $1 + \alpha_{\max}$-competitive online algorithm was given.

To the best of our knowledge, in some new models, the processing time is viewed as a linear function of its starting time, i.e., $p_l = \beta_l(E + FS_l)$, where $E \geq 0$ and $F \geq 0$ are constants satisfying $E + F > 0$ or other conditions. For example, the factory purchases a batch of equipment, owing to improper operation or unsuitable material, the equipment may have been damaged from the beginning. Therefore, the factory is forced to maintain them. The maintenance will take a certain time, that is, the maintenance time is always no less than 0. Most notably, the longer the equipment is damaged, the more time it takes to maintain. Undoubtedly, the above model covers almost all possible situations in real life, it follows that it is much meaningful to consider such a model. In 2016, Ma et al. [19] considered problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l(E + FS_l)|\sum w_l C_l$ and offered an optimal algorithm DSWGR such that the competitive ratio is $\mu(E)$, where, if $E = 0$, then $\mu(E) = 1 + \beta_{\max} F$; and if $E > 0$, then $\mu(E) = 2 + \beta_{\max} F$.

For clarity, the relevant results mentioned above will be shown in Table 1 and Table 2.

Table 1: The relevant results of off-line scheduling.

Problem	Result	Reference
$1\|r_l = t_0, p_l = \beta_l S_l\|C_{\max}$	The objective value is independent of the processing order.	[20]
$1\|r_l, p_l = a_l + \beta S_l\|C_{\max}$	It is strongly NP-complete.	[4]
$1\|p_l = a_l + f(t)\|C_{\max}$	There is an optimal schedule without processor idleness (for all $t' < t''$, $f(t') \geq f(t'')$ and $f(t') + t' \leq f(t'') + t''$).	[22]

Table 2: The relevant results of online scheduling.

Problem	Result			Reference
	Algorithm	Competitive ratio	Optimal	
$1\|\text{online}, r_l \geq t_0, p_l = \beta_l S_l\|C_{\max}$	Greedy algorithm	1	√	[17]
$1\|\text{online}, r_l\|D_{\max}$	DLDT	$\frac{\sqrt{5}+1}{2}$	√	[13]
$1\|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l = \alpha_l C_l\|D_{\max}$	Greedy algorithm	$\max\{1+\beta_{\max}, 1+\alpha_{\max}\}$	√	[17]
$1\|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l = \alpha_l S_l\|D_{\max}$	-	$\max\{1+\beta_{\max}, 1+\alpha_{\max}\}$	√	[3]
$1\|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l\|D_{\max}$	-	$\frac{1+2\beta_{\max}}{1+\beta_{\max}}$	√	[3]
$1\|\text{online}, r_l, q_l = \alpha_l C_l\|D_{\max}$	-	2 ($\alpha_{\max} > 1$)	√	[3]
	-	$1+\alpha_{\max}$ ($\alpha_{\max} \leq 1$)	√	
$1\|\text{online}, r_l \geq t_0, p_l = \beta_l(E+FS_l)\|\sum w_l C_l$	DSWGR	$\mu(E)$	√	[19]

Contribution. Competitive ratio, as far as we know, is one of the most momentous evaluation criteria for the quality of an online algorithm. Unfortunately, for an online algorithm, originating from information deficiency, it is very laborious to provide the competitive analysis. In the paper, to be precise, the main contribution is composed of the following three aspects:

(1) For problem $\mathcal{P}_1$, an optimal online algorithm is presented.

(2) For problem $\mathcal{P}_2$, we show that there exists no online algorithm $\mathscr{A}$ such that $\rho(\mathscr{A}) < \frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$; furthermore, we design an online algorithm with competitive ratio $\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$, matching the lower bound we proposed, which means that our algorithm is optimal.

(3) For problem $\mathcal{P}_3$, we give an online algorithm with a competitive ratio of $\varphi(E)$ when $\alpha_{\max}F \leq 1$, where $\varphi(E) = 1 + \alpha_{\max}F$ or $\varphi(E) = 2 + \alpha_{\max}F$ depending on whether $E = 0$ or $E > 0$; and we offer an optimal online algorithm and point out that its competitive ratio is 2 when $\alpha_{\max}F > 1$.

Organization. The remaining structure of the work is arranged as below: some fundamental notations and definitions are provided in Section 2; then, for problem $\mathcal{P}_1$, we provide an optimal online algorithm in Section 3; next, two online scheduling problems with deterioration and delivery time are addressed in Section 4; conclusively, a summary of the study and the future prospects are presented in Section 5.

2 Preliminaries

As a matter of convenience, some basic notations supplied throughout the work will be given.

- I: an instance.
- J_l: the job with index l, where $l = 1, 2, \ldots, n$.
- r_l: the release date of job J_l.
- β_l: the processing deterioration rate of job J_l.
- $\beta_{\max} = \max\{\beta_l \mid J_l \in I\}$: the largest processing deterioration rate among all jobs in I.
- α_l: the transport deterioration rate of job J_l.
- $\alpha_{\max} = \max\{\alpha_l \mid J_l \in I\}$: the largest transport deterioration rate among all jobs in I.
- $\mathscr{E}^*(I)$: the schedule of I produced by an optimal off-line algorithm.

- $\mathscr{E}(I)$: the schedule of I produced by a feasible algorithm $\mathscr{A}$.
- $S_l^{\mathscr{E}}$: the starting time of job J_l in $\mathscr{E}(I)$.
- $p_l^{\mathscr{E}}$: the processing time of job J_l in $\mathscr{E}(I)$. Note that, if the processing time is a constant, then it can be denoted as p_l.
- $q_l^{\mathscr{E}}$: the transportation time of job J_l in $\mathscr{E}(I)$. Notice that, if the value of the transportation time is given, then it can be shown as q_l.
- $C_l^{\mathscr{E}} = S_l^{\mathscr{E}} + p_l^{\mathscr{E}}$: the completion time of job J_l in $\mathscr{E}(I)$.
- $C_{\max}^{\mathscr{E}} = \max\{C_l^{\mathscr{E}} \mid J_l \in I\}$: the makespan of I in $\mathscr{E}(I)$.
- $D_l^{\mathscr{E}} = C_l^{\mathscr{E}} + q_l^{\mathscr{E}}$: the delivery time of job J_l in $\mathscr{E}(I)$.
- $D_{\max}^{\mathscr{E}} = \max\{D_l^{\mathscr{E}} \mid J_l \in I\}$: the maximum delivery time among the jobs of I in $\mathscr{E}(I)$.

Assume that $A \subseteq I$, then

- $|A|$: the cardinality of A.
- $S_A^{\mathscr{E}} = \min\{S_l^{\mathscr{E}} \mid J_l \in A\}$: the earliest starting time among all jobs of A in $\mathscr{E}(I)$.
- $C_A^{\mathscr{E}} = \max\{C_l^{\mathscr{E}} \mid J_l \in A\}$: the latest completion time among all jobs of A in $\mathscr{E}(I)$.

Noticeably, at each time t, if job J_l has arrived and has not been processed, then it is known as *available*. In the rest of the work, to simplify the presentation, define $\mathcal{A}(t)$ as the available job set at time t.

3 Makespan Minimization

For an instance I, there is a group of deteriorating jobs reaching online over time. Notice that, the processing time of each job J_l is denoted as $p_l = \beta_l(E + FS_l)$, where $E \geq 0$ and $F \geq 0$ are constants meeting $E + F > 0$. In this section, we want to schedule the jobs so as to minimize the makespan. As we know, Liu et al. [17] presented an optimal online algorithm for problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l|C_{\max}$, a special case of problem $\mathcal{P}_1$. Next, we will offer an online algorithm for problem $\mathcal{P}_1$, which works as below:

Algorithm 3.1. $\mathscr{A}_1$

Step 1: If the processor is available and $\mathcal{A}(t) \neq \emptyset$, then choose an arbitrary job in $\mathcal{A}(t)$ to process;

Step 2: If all jobs have been handled, stop; otherwise, go to Step 1.

Let $\mathscr{F}(I)$ stand for the schedule of I produced by algorithm $\mathscr{A}_1$. For ease of description, some useful definitions will be provided as follows:

Suppose that $A = \{J_a, J_{a+1}, \ldots, J_b\}$ is the maximal job set such that the processor retains busy in $[S_A^{\mathscr{F}}, C_A^{\mathscr{F}})$, then A is referred to as a *block*, where J_l is the lth job in $\mathscr{F}(I)$, $1 \leq a \leq l \leq b \leq n$. Undoubtedly, $\mathscr{F}(I)$ may consist of plenty of blocks. Now, we first consider the situation that the number of blocks is one, which plays a crucial role in the algorithm analysis.

Lemma 3.2. *For problem $\mathcal{P}_1$, if $\mathscr{F}(I)$ is composed of one block, then, we have*

$$C^{\mathscr{F}}_{\max} = \begin{cases} S + E\sum_{l=1}^{n} \beta_l, & F = 0; \\ \left(S + \dfrac{E}{F}\right)\prod_{l=1}^{n}(\beta_l F + 1) - \frac{E}{F}, & F > 0; \end{cases} \tag{3.1}$$

where $S = \min\{S^{\mathscr{F}}_l \mid J_l \in I\}$.

Proof. We proceed by induction concerning the cardinality of I. Without loss of generality, we may assume that J_l indicates the lth job in $\mathscr{F}(I)$.

Step 1: $|I| = 1$. It is clear that, $C^{\mathscr{F}}_{\max} = C^{\mathscr{F}}_1 = S + \beta_1 E$ when $F = 0$, and $C^{\mathscr{F}}_{\max} = C^{\mathscr{F}}_1 = S + \beta_1(E + FS) = \beta_1 E + (1 + \beta_1 F)S = \left(S + \dfrac{E}{F}\right)(\beta_1 F + 1) - \frac{E}{F}$ when $F > 0$, which means that equation (3.1) holds.

Step 2: Assume that equation (3.1) follows when $|I| = m - 1$. That is,

$$C^{\mathscr{F}}_{\max} = \begin{cases} S + E\sum_{l=1}^{m-1} \beta_l, & F = 0, \\ (S + \frac{E}{F})\prod_{l=1}^{m-1}(\beta_l F + 1) - \frac{E}{F}, & F > 0. \end{cases}$$

Step 3: $|I| = m$. If $F = 0$, then, we derive that

$$C^{\mathscr{F}}_{\max} = C^{\mathscr{F}}_m = C^{\mathscr{F}}_{m-1} + p^{\mathscr{F}}_m = S + E\sum_{l=1}^{m-1}\beta_l + E\beta_m = S + E\sum_{l=1}^{m}\beta_l.$$

If $F > 0$, then, we infer that

$$\begin{aligned} C^{\mathscr{F}}_{\max} =& C^{\mathscr{F}}_m = C^{\mathscr{F}}_{m-1} + p^{\mathscr{F}}_m = \beta_m E + (1 + \beta_m F)C^{\mathscr{F}}_{m-1} \\ =& \beta_m E + (1 + \beta_m F)\left[\left(S + \frac{E}{F}\right)\prod_{l=1}^{m-1}(\beta_l F + 1) - \frac{E}{F}\right] \\ =& \beta_m E + \left(S + \frac{E}{F}\right)\prod_{l=1}^{m}(\beta_l F + 1) - \frac{E}{F} - \beta_m E \\ =& \left(S + \frac{E}{F}\right)\prod_{l=1}^{m}(\beta_l F + 1) - \frac{E}{F}. \end{aligned}$$

The lemma is true. □

With the help of Lemma 3.2, the competitive ratio will be received more smoothly. Thereby, we obtain the following conclusion.

Theorem 3.3. *For problem $\mathcal{P}_1$, algorithm $\mathscr{A}_1$ is optimal.*

Proof. For each instance I, the number of blocks in $\mathscr{F}(I)$ may be k, where $k \geq 1$. In consideration of our objective function, it follows that, it is ample to consider the last block $\mathscr{B}_k$ in $\mathscr{F}(I)$. Let $\mathscr{B}_k = \{J_a, J_{a+1}, \ldots, J_n\}$, where J_l is the lth job in $\mathscr{F}(I)$, $1 \leq a \leq l \leq n$.

According to algorithm $\mathscr{A}_1$, we deduce that $S^{\mathscr{F}}_{\mathscr{B}_k} = r$, where $r = \min\{r_l \mid J_l \in \mathscr{B}_k\}$. From Lemma 3.2, we gain that

$$C^{\mathscr{F}}_{\max} = \begin{cases} r + E\sum_{l=a}^{n}\beta_l, & F = 0, \\ \left(r + \dfrac{E}{F}\right)\prod_{l=a}^{n}(\beta_l F + 1) - \frac{E}{F}, & F > 0. \end{cases}$$

Note that, for each job $J_l \in \mathscr{B}_k$, $r_l \geq r$, and the objective value is independent of the processing order of the jobs in $\mathscr{B}_k$, therefore, we get that

$$C^{\mathscr{E}^*}_{\max} \geq \begin{cases} r + E\sum_{l=a}^{n}\beta_l, & F = 0, \\ \left(r + \dfrac{E}{F}\right)\prod_{l=a}^{n}(\beta_l F + 1) - \frac{E}{F}, & F > 0. \end{cases}$$

In the light of the minimality of the value of $C^{\mathscr{E}^*}_{\max}$, we conclude that $C^{\mathscr{F}}_{\max} = C^{\mathscr{E}^*}_{\max}$, which means that algorithm $\mathscr{A}_1$ is optimal. The proof is completed. □

4 Maximum Delivery Time Minimization

In the majority of cases, the jobs may need to be transported to the customer or the depository after processing, and the transportation of every job will take some time. There is no doubt that the problem with delivery time is closely relevant to practice. As far as we know, minimizing the maximum delivery time is one of the momentous objective functions in scheduling. In this section, we will consider two online scheduling problems with deterioration and delivery time.

To facilitate narration, we will provide some fundamental definitions, which will be employed in the this section.

Note that, in $\mathscr{E}(I)$, if $D^{\mathscr{E}}_{\max} = D^{\mathscr{E}}_{m}$ holds, then job J_m is said to be *decisive*, where $1 \leq m \leq n$. Suppose that J_l is the lth job in $\mathscr{E}(I)$. Set $A = \{J_a, J_{a+1}, \ldots, J_m\}$ and $A_1 = \{J_{a-1}, J_a, \ldots, J_m\}$, where $1 \leq a \leq m \leq n$. In $\mathscr{E}(I)$, if the processor keeps available in $[t_1, t_2) \subset [S^{\mathscr{E}}_{A_1}, C^{\mathscr{E}}_{A_1})$, and it retains busy in $[S^{\mathscr{E}}_{A}, C^{\mathscr{E}}_{A})$, then A is referred to as a *significant set.*

4.1 Deteriorating processing time

For an instance I, there exists a series of deteriorating jobs reaching online over time. Notably, for each job J_l, its processing time p_l is defined as $p_l = \beta_l(E + FS_l)$, where $E \geq 0$, $F > 0$, and $\frac{E}{F} \to 0$. In 2020, Chai et al. [3] addressed problem $1|\text{online}, r_l \geq t_0, p_l = \beta_l S_l, q_l|D_{\max}$, a special case of problem $\mathcal{P}_2$. Next, we will offer the lower bound for problem $\mathcal{P}_2$. Furthermore, we will introduce an online algorithm, whose competitive ratio matches the lower bound we propose, which means that our online algorithm is optimal.

Before providing the lower bound, a fundamental lemma with proof is put forward, which is central for our work.

Lemma 4.1. *Let a, b, c and d be four positive numbers. Then, we have*

$$\frac{a+c}{b+d} \geq \min\left\{\frac{a}{b},\ \frac{c}{d}\right\}.$$

Proof. Consider two cases as below:

Case A: $\frac{a}{b} \leq \frac{c}{d}$, which signifies $ad \leq bc$. Combining with the property that all numbers are positive, we derive that

$$\frac{a+c}{b+d} - \frac{a}{b} = \frac{b(a+c) - a(b+d)}{b(b+d)} = \frac{bc - ad}{b(b+d)} \geq 0.$$

Case B: $\frac{a}{b} > \frac{c}{d}$, which implies $ad > bc$. Recall that all numbers are positive, we infer that

$$\frac{a+c}{b+d} - \frac{c}{d} = \frac{d(a+c) - c(b+d)}{d(b+d)} = \frac{ad - bc}{d(b+d)} > 0.$$

The lemma is true. □

Next, we will employ the adversary strategy to present the lower bound for problem $\mathcal{P}_2$. Benefiting from Theorem 4.1, the lower bound will be obtained more smoothly.

Theorem 4.2. *For problem $\mathcal{P}_2$, there is no online algorithm $\mathscr{A}$ with $\rho(\mathscr{A}) < \frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$.*

Proof. Define $\mathscr{A}$ as an arbitrary feasible online algorithm for problem $\mathcal{P}_2$. Let $\mathscr{E}(I)$ indicate the schedule of I produced by algorithm $\mathscr{A}$. Suppose that job J_1 with $\beta_1 = \beta$ and $q_1 = 0$ is released at time t_0. We may assume that $S_1^{\mathscr{E}} = t$, where $t \geq t_0$, then the adversary will release job J_2 with $\beta_2 = 0$ and $q_2 = \beta(E + Ft)$ at time $t + \epsilon$, where $\epsilon \to 0$.

Case A: $E = 0$. Then, we deduce that $D_1^{\mathscr{E}} = C_1^{\mathscr{E}} + q_1 = t + \beta Ft + 0 = (1 + \beta F)t$ and $D_2^{\mathscr{E}} = C_2^{\mathscr{E}} + q_2 = (1 + \beta F)t + \beta Ft = (1 + 2\beta F)t$, which means that

$$D_{\max}^{\mathscr{E}} = D_2^{\mathscr{E}} = (1 + 2\beta F)t.$$

Noticeably, in $\mathscr{E}^*(I)$, the scheduler will assign job J_2 at time $t + \epsilon$ followed by job J_1. Hence, we infer that $D_1^{\mathscr{E}^*} = C_1^{\mathscr{E}^*} + q_1 = t + \epsilon + \beta F(t + \epsilon) + 0 = (1 + \beta F)(t + \epsilon)$ and $D_2^{\mathscr{E}^*} = C_2^{\mathscr{E}^*} + q_2 = t + \epsilon + \beta Ft = (1 + \beta F)t + \epsilon$, it follows that

$$D_{\max}^{\mathscr{E}^*} = \max\{(1 + \beta F)(t + \epsilon), (1 + \beta F)t + \epsilon\} \to (1 + \beta F)t \quad (\epsilon \to 0).$$

Therefore, we obtain that

$$\frac{D_{\max}^{\mathscr{E}}}{D_{\max}^{\mathscr{E}^*}} \to \frac{(1 + 2\beta F)t}{(1 + \beta F)t} = \frac{1 + 2\beta F}{1 + \beta F} = \frac{1 + 2\beta_{\max}F}{1 + \beta_{\max}F}.$$

Case B: $E > 0$. Then, we derive that $D_1^{\mathscr{E}} = C_1^{\mathscr{E}} + q_1 = t + \beta(E + Ft) + 0 = (1 + \beta F)t + \beta E$ and $D_2^{\mathscr{E}} = C_2^{\mathscr{E}} + q_2 = (1 + \beta F)t + \beta E + \beta(E + Ft) = (1 + 2\beta F)t + 2\beta E$, which signifies that

$$D_{\max}^{\mathscr{E}} = D_2^{\mathscr{E}} = (1 + 2\beta F)t + 2\beta E.$$

Notably, in $\mathscr{E}^*(I)$, job J_2 will be processed at time $t + \epsilon$ followed by job J_1. Thus, we have $D_1^{\mathscr{E}^*} = C_1^{\mathscr{E}^*} + q_1 = t + \epsilon + \beta[E + F(t + \epsilon)] + 0 = (1 + \beta F)(t + \epsilon) + \beta E$ and $D_2^{\mathscr{E}^*} = C_2^{\mathscr{E}^*} + q_2 = t + \epsilon + \beta(E + Ft) = (1 + \beta F)t + \epsilon + \beta E$, which implies that

$$D_{\max}^{\mathscr{E}^*} = \max\{(1 + \beta F)(t + \epsilon) + \beta E, (1 + \beta F)t + \epsilon + \beta E\} \to (1 + \beta F)t + \beta E \quad (\epsilon \to 0).$$

Along with Lemma 4.1, we get that

$$\frac{D_{\max}^{\mathscr{E}}}{D_{\max}^{\mathscr{E}^*}} \to \frac{(1 + 2\beta F)t + 2\beta E}{(1 + \beta F)t + \beta E} \geq \min\left\{\frac{1 + 2\beta F}{1 + \beta F}, 2\right\} = \frac{1 + 2\beta F}{1 + \beta F} = \frac{1 + 2\beta_{\max}F}{1 + \beta_{\max}F}.$$

The proof is completed. □

From Theorem 4.2, we know that the lower bound for problem $\mathcal{P}_2$ is $\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$. Now, an online algorithm $\mathscr{A}_2$ is given, which works as below:

Algorithm 4.3. $\mathscr{A}_2$

Step 1: If the processor is available and $\mathcal{A}(t) \neq \emptyset$, then choose a job with the largest transportation time from $\mathcal{A}(t)$ to process, and transport it instantly after processing;

Step 2: If all jobs have been transported, stop; otherwise, go to Step 1.

Define $\mathscr{F}(I)$ as the schedule of I produced by algorithm $\mathscr{A}_2$. In order to facilitate the description, we may assume that J_l is the lth job in $\mathscr{F}(I)$, where $1 \leq l \leq n$. Next, we will analyze the competitive ratio of algorithm $\mathscr{A}_2$.

Lemma 4.4. $\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \leq \frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$.

Proof. Suppose that job J_m is the decisive job in $\mathscr{F}(I)$, which signifies that $D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m$. Let $B = \{J_a, J_{a+1}, \ldots, J_m\}$ be a significant set in $\mathscr{F}(I)$. According to algorithm $\mathscr{A}_2$, we infer that, $S^{\mathscr{F}}_a = \min\{\ r_l \mid J_l \in B\}$ follows, which means that, for every job $J_l \in B$, the condition $r_l \geq S^{\mathscr{F}}_a$ is respected. Set $q_{\min}(B) = \min\{\ q_l \mid J_l \in B\}$.

Case A: $E = 0$. Hence, we have

$$D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m = S^{\mathscr{F}}_a \prod_{l=a}^{m}(1+\beta_l F) + q_m. \tag{4.1}$$

Case A1: $q_{\min}(B) = q_m$. It is clear that

$$D^{\mathscr{E}^*}_{\max} \geq S^{\mathscr{F}}_a \prod_{l=a}^{m}(1+\beta_l F) + q_m.$$

In consideration of the minimality of the value of $D^{\mathscr{E}^*}_{\max}$, we receive that

$$D^{\mathscr{E}^*}_{\max} = D^{\mathscr{F}}_{\max}.$$

Case A2: $q_{\min}(B) < q_m$. Suppose that, in $\mathscr{F}(I)$, job J_b is the last job of B meeting $q_b < q_m$. Let $E = \{J_{b+1}, J_{b+2}, \ldots, J_m\}$. There is no doubt that $q_m = q_{\min}(E) > q_b$. From algorithm $\mathscr{A}_2$, we observe that $S^{\mathscr{F}}_b < r_{\min}(E)$, where $r_{\min}(E) = \min\{\ r_l \mid J_l \in E\}$. It follows that

$$D^{\mathscr{E}^*}_{\max} \geq r_{\min}(E) \prod_{l=b+1}^{m}(1+\beta_l F) + q_m > S^{\mathscr{F}}_b \prod_{l=b+1}^{m}(1+\beta_l F) + q_m. \tag{4.2}$$

In view of the definition of significant set, we have

$$D^{\mathscr{E}^*}_{\max} \geq S^{\mathscr{F}}_a \prod_{l=a}^{m}(1+\beta_l F) = S^{\mathscr{F}}_b \prod_{l=b}^{m}(1+\beta_l F). \tag{4.3}$$

Together with equation (4.1), we deduce that

$$D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m = S^{\mathscr{F}}_a \prod_{l=a}^{m}(1+\beta_l F) + q_m = S^{\mathscr{F}}_b \prod_{l=b}^{m}(1+\beta_l F) + q_m. \tag{4.4}$$

Combining with inequalities (4.2)–(4.4), we get that

$$\begin{aligned}
\frac{D_{\max}^{\mathscr{F}} - D_{\max}^{\mathscr{E}^*}}{D_{\max}^{\mathscr{E}^*}} &< \frac{S_b^{\mathscr{F}} \prod_{l=b}^{m}(1+\beta_l F) + q_m - (S_b^{\mathscr{F}} \prod_{l=b+1}^{m}(1+\beta_l F) + q_m)}{S_b^{\mathscr{F}} \prod_{l=b}^{m}(1+\beta_l F)} \\
&= \frac{S_b^{\mathscr{F}} \prod_{l=b}^{m}(1+\beta_l F) - S_b^{\mathscr{F}} \prod_{l=b+1}^{m}(1+\beta_l F)}{S_b^{\mathscr{F}} \prod_{l=b}^{m}(1+\beta_l F)} \\
&= \frac{\beta_b F}{1+\beta_b F} \leq \frac{\beta_{\max} F}{1+\beta_{\max} F},
\end{aligned}$$

which signifies that

$$\frac{D_{\max}^{\mathscr{F}}}{D_{\max}^{\mathscr{E}^*}} \leq \frac{1+2\beta_{\max} F}{1+\beta_{\max} F}.$$

Case B: $E > 0$. In view of $F > 0$, along with equation (3.1), we infer that

$$D_{\max}^{\mathscr{F}} = D_m^{\mathscr{F}} = \left(S_a^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=a}^{m}(1+\beta_l F) - \frac{E}{F} + q_m. \tag{4.5}$$

Case B1: $q_{\min}(B) = q_m$. It is obvious that

$$D_{\max}^{\mathscr{E}^*} \geq \left(S_a^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=a}^{m}(1+\beta_l F) - \frac{E}{F} + q_m.$$

In view of the minimality of the value of $D_{\max}^{\mathscr{E}^*}$, we gain that

$$D_{\max}^{\mathscr{E}^*} = D_{\max}^{\mathscr{F}}.$$

Case B2: $q_{\min}(B) < q_m$. Suppose that, in $\mathscr{F}(I)$, job J_b is the last job of B with $q_b < q_m$. Set $E = \{J_{b+1}, J_{b+2}, \ldots, J_m\}$. It is clear that $q_m = q_{\min}(E) > q_b$. By algorithm $\mathscr{A}_2$, we derive that $S_b^{\mathscr{F}} < \min\{\, r_l \mid J_l \in E\}$ holds. Thereby, we obtain that

$$D_{\max}^{\mathscr{E}^*} \geq (S_b^{\mathscr{F}} + \frac{E}{F}) \prod_{l=b+1}^{m}(1+\beta_l F) - \frac{E}{F} + q_m. \tag{4.6}$$

Note that,

$$D_{\max}^{\mathscr{E}^*} \geq \left(S_a^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=a}^{m}(1+\beta_l F) - \frac{E}{F} = \left(S_b^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=b}^{m}(1+\beta_l F) - \frac{E}{F}. \tag{4.7}$$

In addition, along with equation (4.5), we have

$$D_{\max}^{\mathscr{F}} = D_m^{\mathscr{F}} = \left(S_a^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=a}^{m}(1+\beta_l F) - \frac{E}{F} + q_m = \left(S_b^{\mathscr{F}} + \frac{E}{F}\right) \prod_{l=b}^{m}(1+\beta_l F) - \frac{E}{F} + q_m. \tag{4.8}$$

In the light of $\frac{E}{F} \to 0$, together with inequalities (4.6)–(4.8), we get that

$$\begin{aligned}
&\frac{D^{\mathscr{F}}_{\max} - D^{\mathscr{E}^*}_{\max}}{D^{\mathscr{E}^*}_{\max}} \\
&\quad \le \frac{\left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b}^{m}(1+\beta_l F) - \frac{E}{F} + q_m - \left[\left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b+1}^{m}(1+\beta_l F) - \frac{E}{F} + q_m\right]}{\left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b}^{m}(1+\beta_l F) - \frac{E}{F}} \\
&\quad = \frac{\left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b}^{m}(1+\beta_l F) - \left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b+1}^{m}(1+\beta_l F)}{\left(S^{\mathscr{F}}_b + \dfrac{E}{F}\right)\prod_{l=b}^{m}(1+\beta_l F) - \frac{E}{F}} \\
&\quad \to \frac{\beta_b F}{1+\beta_b F} \le \frac{\beta_{\max} F}{1+\beta_{\max} F},
\end{aligned}$$

which implies that

$$\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \le \frac{1+2\beta_{\max}F}{1+\beta_{\max}F}.$$

The proof is completed. □

From Lemma 4.4, we receive that the upper bound of $\rho(\mathscr{A}_2)$ is $\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$, matching the lower bound shown in Theorem 4.2, which signifies that our algorithm is optimal. As a result, we draw a conclusion as below:

Theorem 4.5. *For problem $\mathcal{P}_2$, algorithm $\mathscr{A}_2$ is an optimal online algorithm with a competitive ratio of $\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$.*

4.2 Deteriorating transportation time

For an instance I, there exists a set of jobs reaching online over time. Noticeably, for every job J_l, its transportation time q_l is denoted as $q_l = \alpha_l(E + FC_l)$, where $E \ge 0$ and $F \ge 0$ are constants respecting $E + F > 0$. Recall that Chai et al. [3] considered problem $1|\text{online}, r_l, q_l = \alpha_l C_l|D_{\max}$, a special case of problem $\mathcal{P}_3$. In addition, they pointed out that every online algorithm $\mathscr{A}$ meets $\rho(\mathscr{A}) \ge \min\{2, 1+\alpha_{\max}\}$. Therefore, we obtain the following conclusion.

Theorem 4.6. *For problem $\mathcal{P}_3$, there exists no online algorithm $\mathscr{A}$ such that*

$$\rho(\mathscr{A}) < \begin{cases} 2, & \alpha_{\max}F > 1, \\ 1+\alpha_{\max}F, & \alpha_{\max}F \le 1. \end{cases}$$

From Theorem 4.6, we receive that the lower bound for problem $\mathcal{P}_3$. In the following, we will provide online algorithms and performance analysis for two cases as below: $\alpha_{\max}F > 1$ and $\alpha_{\max}F \le 1$.

When $\alpha_{\max}F > 1$, we introduce an optimal online algorithm, which works as follows:

Algorithm 4.7. $\mathscr{A}_3$

Step 1: When the processor is available and $\mathcal{A}(t) \ne \emptyset$, choose the job with the largest transport deterioration rate in $\mathcal{A}(t)$, say J_l;

Step 2: If $t \geq p_l$, then process job J_l, and transport it instantly after processing; otherwise, wait until time p_l or a new job reaches, whichever comes first, and go to Step 1;

Step 3: If all jobs have been transported, stop; otherwise, go to Step 1.

Let $\mathscr{F}(I)$ indicate the schedule of I produced by algorithm $\mathscr{A}_3$. For simplicity, we suppose that J_l stands for the lth job in $\mathscr{F}(I)$, where $1 \leq l \leq n$. Now, we begin to analyze the competitive ratio of algorithm $\mathscr{A}_3$.

Lemma 4.8. $\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \leq 2$.

Proof. Let J_m be the decisive job in $\mathscr{F}(I)$, which means that $D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m$. Define $B = \{J_a, J_{a+1}, \ldots, J_m\}$ as a significant set in $\mathscr{F}(I)$. There is no doubt that,

$$\begin{aligned} D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m = C^{\mathscr{F}}_m + q^{\mathscr{F}}_m &= S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l + \alpha_m \left[E + F\left(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l\right)\right] \\ &= (1+\alpha_m F)\left(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l\right) + \alpha_m E. \end{aligned} \tag{4.9}$$

Set $\alpha_{\min}(B) = \min\{\ \alpha_l \mid J_l \in B\}$.

Case A: $\alpha_m = \alpha_{\min}(B)$. By algorithm $\mathscr{A}_3$, we observe that $\alpha_a \geq \alpha_m$. Notice that, if $S^{\mathscr{F}}_a > r_a + p_a$, then the scheduler will assign job J_a before time $S^{\mathscr{F}}_a$ in $\mathscr{F}(I)$, and thus, we have $S^{\mathscr{F}}_a \leq r_a + p_a$. In addition, we infer that

$$\begin{aligned} D^{\mathscr{E}^*}_{\max} &\geq C^{\mathscr{E}^*}_a + q^{\mathscr{E}^*}_a \geq r_a + p_a + \alpha_a[E + F(r_a + p_a)] \\ &\geq (1+\alpha_a F)(r_a + p_a) + \alpha_a E \\ &\geq (1+\alpha_m F) S^{\mathscr{F}}_a + \alpha_m E. \end{aligned} \tag{4.10}$$

Furthermore, we derive that

$$D^{\mathscr{E}^*}_{\max} \geq \sum_{l=a}^{m} p_l + \alpha_m \left(E + F\sum_{l=a}^{m} p_l\right) = (1+\alpha_m F)\sum_{l=a}^{m} p_l + \alpha_m E \geq (1+\alpha_m F)\sum_{l=a}^{m} p_l. \tag{4.11}$$

According to inequalities (4.9)–(4.11), we receive that

$$\begin{aligned} D^{\mathscr{F}}_{\max} - D^{\mathscr{E}^*}_{\max} &\leq (1+\alpha_m F)\left(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l\right) + \alpha_m E - [(1+\alpha_m F) S^{\mathscr{F}}_a + \alpha_m E] \\ &= (1+\alpha_m F)\sum_{l=a}^{m} p_l \\ &\leq D^{\mathscr{E}^*}_{\max}, \end{aligned}$$

that is,

$$\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \leq 2.$$

Case B: $\alpha_m > \alpha_{\min}(B)$. Let J_b be the last job of B with $\alpha_b < \alpha_m$ in $\mathscr{F}(I)$. Set $E = \{J_{b+1}, J_{b+2}, \ldots, J_m\}$. It is obvious that $\alpha_m = \alpha_{\min}(E) > \alpha_b$. From algorithm $\mathscr{A}_3$, we know that $S_b^{\mathscr{F}} < \min\{ r_l \mid J_l \in E\}$ follows. Hence, we gain that

$$D_{\max}^{\mathscr{E}^*} \geq S_b^{\mathscr{F}} + \sum_{l=b+1}^{m} p_l + \alpha_m \left[E + F\left(S_b^{\mathscr{F}} + \sum_{l=b+1}^{m} p_l \right) \right] = (1+\alpha_m F)\left(S_b^{\mathscr{F}} + \sum_{l=b+1}^{m} p_l \right) + \alpha_m E. \tag{4.12}$$

In view of the definition of significant set, it follows that $S_a^{\mathscr{F}} + \sum_{l=a}^{m} p_l = S_b^{\mathscr{F}} + \sum_{l=b}^{m} p_l$. Together with equation (4.9), we have

$$D_{\max}^{\mathscr{F}} = D_m^{\mathscr{F}} = (1+\alpha_m F)\left(S_a^{\mathscr{F}} + \sum_{l=a}^{m} p_l \right) + \alpha_m E = (1+\alpha_m F)\left(S_b^{\mathscr{F}} + \sum_{l=b}^{m} p_l \right) + \alpha_m E. \tag{4.13}$$

By algorithm $\mathscr{A}_3$, we deduce that $S_b^{\mathscr{F}} \geq p_b$. Combining with inequalities (4.12) and (4.13), we obtain that

$$\begin{aligned}
\frac{D_{\max}^{\mathscr{F}}}{D_{\max}^{\mathscr{E}^*}} &\leq \frac{(1+\alpha_m F)(S_b^{\mathscr{F}} + \sum_{l=b}^{m} p_l) + \alpha_m E}{(1+\alpha_m F)(S_b^{\mathscr{F}} + \sum_{l=b+1}^{m} p_l) + \alpha_m E} \\
&= 1 + \frac{(1+\alpha_m F) p_b}{(1+\alpha_m F)(S_b^{\mathscr{F}} + \sum_{l=b+1}^{m} p_l) + \alpha_m E} \\
&\leq 2.
\end{aligned}$$

The proof is completed. □

From Lemma 4.8, we receive that, 2 is an upper bound of $\rho(\mathscr{A}_3)$. Next, we will propose a simple example to show the bound is tight.

Example 4.9. Let $I = \{(r_1 = 0, p_1 = p, \alpha_1 = 0)\}$, where $p > 0$. Obviously, $q_1^{\mathscr{E}^*} = q_1^{\mathscr{F}} = 0$. In addition, we observe that $S_1^{\mathscr{E}^*} = 0$, $C_1^{\mathscr{E}^*} = S_1^{\mathscr{E}^*} + p_1 = p$ and $D_{\max}^{\mathscr{E}^*} = D_1^{\mathscr{E}^*} = C_1^{\mathscr{E}^*} + q_1^{\mathscr{E}^*} = p + 0 = p$. According to algorithm $\mathscr{A}_3$, we have $S_1^{\mathscr{F}} = p$, $C_1^{\mathscr{F}} = S_1^{\mathscr{F}} + p_1 = 2p$ and $D_{\max}^{\mathscr{F}} = D_1^{\mathscr{F}} = C_1^{\mathscr{F}} + q_1^{\mathscr{F}} = 2p + 0 = 2p$. Thus, we gain that $\frac{D_{\max}^{\mathscr{F}}}{D_{\max}^{\mathscr{E}^*}} = \frac{2p}{p} = 2$.

For the sake of intuition, we give Figure 1 to show the corresponding schedules.

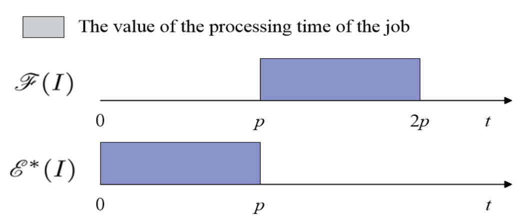

Figure 1: The corresponding schedules.

Combining with Theorem 4.6 and Example 4.9, we come to the following conclusion:

Theorem 4.10. *For problem $\mathcal{P}_3$, if $\alpha_{\max} F > 1$, then algorithm $\mathscr{A}_3$ is an optimal online algorithm with a competitive ratio of* 2.

When $\alpha_{\max} F \leq 1$, we design an online algorithm, which can be described as below:

Algorithm 4.11. $\mathscr{A}_4$

Step 1: If the processor is available and $\mathcal{A}(t) \neq \emptyset$, then choose an arbitrary job in $\mathcal{A}(t)$ to process, and transport it instantly after processing;

Step 2: If all jobs have been transported, stop; otherwise, go to Step 1.

Define $\mathscr{F}(I)$ as the schedule of I produced by algorithm $\mathscr{A}_4$. For ease of description, we assume that J_l denotes the lth job in $\mathscr{F}(I)$, where $1 \le l \le n$. Set $\varphi(E) = 1 + \alpha_{\max}F$ when $E = 0$ and $\varphi(E) = 2 + \alpha_{\max}F$ when $E > 0$.

Next, we will provide the competitive analysis for algorithm $\mathscr{A}_4$.

Lemma 4.12. $\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \le \varphi(E)$.

Proof. Suppose that J_m is the decisive job in $\mathscr{F}(I)$, it follows that $D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m$. Let $B = \{J_a, J_{a+1}, \dots, J_m\}$ be a significant set in $\mathscr{F}(I)$. Thus, we obtain that

$$\begin{aligned} D^{\mathscr{F}}_{\max} = D^{\mathscr{F}}_m = C^{\mathscr{F}}_m + q^{\mathscr{F}}_m &= S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l + \alpha_m \left[E + F\left(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l\right)\right] \\ &= (1+\alpha_m F)\left(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l\right) + \alpha_m E. \end{aligned} \tag{4.14}$$

From algorithm $\mathscr{A}_4$, we have $S^{\mathscr{F}}_a = \min\{\, r_l \mid J_l \in B\}$, which signifies that, $r_l \ge S^{\mathscr{F}}_a$ holds for every job $J_l \in B$. Hence, we infer that

$$D^{\mathscr{E}^*}_{\max} \ge S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l. \tag{4.15}$$

Case A: $E = 0$. Combining with inequalities (4.14) and (4.15), we get that

$$\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \le \frac{(1+\alpha_m F)(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l)}{S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l} = 1 + \alpha_m F \le 1 + \alpha_{\max} F.$$

Case B: $E > 0$. Notice that

$$D^{\mathscr{E}^*}_{\max} \ge C^{\mathscr{E}^*}_m + q^{\mathscr{E}^*}_m = C^{\mathscr{E}^*}_m + \alpha_m(E + FC^{\mathscr{E}^*}_m) = (1+\alpha_m F)C^{\mathscr{E}^*}_m + \alpha_m E \ge \alpha_m E. \tag{4.16}$$

From inequalities (4.14)–(4.16), we derive that

$$\begin{aligned} \frac{D^{\mathscr{F}}_{\max} - (1+\alpha_m F)D^{\mathscr{E}^*}_{\max}}{D^{\mathscr{E}^*}_{\max}} &\le \frac{(1+\alpha_m F)(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l) + \alpha_m E - (1+\alpha_m F)(S^{\mathscr{F}}_a + \sum_{l=a}^{m} p_l)}{\alpha_m E} \\ &= 1, \end{aligned}$$

which implies that

$$\frac{D^{\mathscr{F}}_{\max}}{D^{\mathscr{E}^*}_{\max}} \le 2 + \alpha_m F \le 2 + \alpha_{\max} F.$$

The lemma is true. ☐

According to Lemma 4.12, we get that, $\varphi(E)$ is an upper bound for the competitive ratio of algorithm $\mathscr{A}_4$. Now, two simple examples are given to attest that the bound is tight.

Example 4.13. Let $I = \{(r_l, p_l, \alpha_l) \mid l = 1, 2\}$, $E = 0$ and $F = \frac{1}{2}$. Furthermore, the information of each job will be shown in Table 3.

Table 3: The information of each job.

J_l	r_l	p_l	α_l
J_1	1	2	0
J_2	$1+\epsilon$	0	2

It is clear that, $\alpha_{\max} F = 2 \times \frac{1}{2} = 1$ and $q_1^{\mathscr{E}^*} = q_1^{\mathscr{F}} = 0$. From algorithm $\mathscr{A}_4$, we know that $D_1^{\mathscr{F}} = C_1^{\mathscr{F}} + q_1^{\mathscr{F}} = 1 + 2 + 0 = 3$ and $D_2^{\mathscr{F}} = C_2^{\mathscr{F}} + q_2^{\mathscr{F}} = 3 + 2 \times (0 + \frac{1}{2} \times 3) = 6$, which means that $D_{\max}^{\mathscr{F}} = D_2^{\mathscr{F}} = 6$. Note that, in $\mathscr{E}^*(I)$, job J_2 is processed at time $1+\epsilon$, and followed by job J_1. Hence, we infer that $D_1^{\mathscr{E}^*} = C_1^{\mathscr{E}^*} + q_1^{\mathscr{E}^*} = 1 + \epsilon + 2 + 0 = 3 + \epsilon$, $D_2^{\mathscr{E}^*} = C_2^{\mathscr{E}^*} + q_2^{\mathscr{E}^*} = 1 + \epsilon + 2 \times [0 + \frac{1}{2} \times (1+\epsilon)] = 2(1+\epsilon)$, and $D_{\max}^{\mathscr{E}^*} = D_1^{\mathscr{E}^*} = 3 + \epsilon$. Thus, we receive that

$$\frac{D_{\max}^{\mathscr{F}}}{D_{\max}^{\mathscr{E}^*}} = \frac{6}{3+\epsilon} \to 2 = 1 + 2 \times \frac{1}{2} = 1 + \alpha_{\max} F \quad (\epsilon \to 0).$$

For intuition, we provide Figure 2 to show the corresponding schedules.

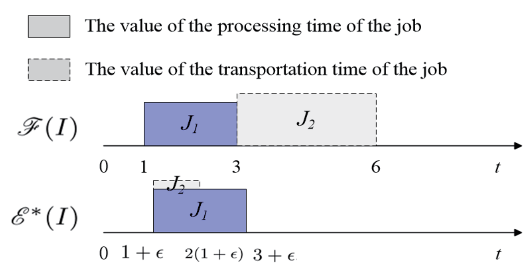

Figure 2: The corresponding schedules.

Example 4.14. Set $I = \{(r_l, p_l, \alpha_l) \mid l = 1, 2\}$, $E = 1$ and $F = \frac{1}{2}$. Moreover, the information of I will be provided in Table 4.

Table 4: The information of each job.

J_l	r_l	p_l	α_l
J_1	0	2	0
J_2	ϵ	0	2

Undoubtedly, $\alpha_{\max} F = 2 \times \frac{1}{2} = 1$ and $q_1^{\mathscr{E}^*} = q_1^{\mathscr{F}} = 0$. According to algorithm $\mathscr{A}_4$, we derive that $D_1^{\mathscr{F}} = C_1^{\mathscr{F}} + q_1^{\mathscr{F}} = 0 + 2 + 0 = 2$ and $D_2^{\mathscr{F}} = C_2^{\mathscr{F}} + q_2^{\mathscr{F}} = 2 + 2 \times (1 + \frac{1}{2} \times 2) = 6$, which signifies that $D_{\max}^{\mathscr{F}} = D_2^{\mathscr{F}} = 6$. Notice that, in $\mathscr{E}^*(I)$, we will assign job J_2 to process at time ϵ, and followed by job J_1. Therefore, we deduce that $D_1^{\mathscr{E}^*} = C_1^{\mathscr{E}^*} + q_1^{\mathscr{E}^*} = \epsilon + 2 + 0 = 2 + \epsilon$, $D_2^{\mathscr{E}^*} = C_2^{\mathscr{E}^*} + q_2^{\mathscr{E}^*} = \epsilon + 2 \times (1 + \frac{1}{2} \times \epsilon) = 2(1+\epsilon)$, and $D_{\max}^{\mathscr{E}^*} = D_2^{\mathscr{E}^*} = 2(1+\epsilon)$. Thus, we obtain that

$$\frac{D_{\max}^{\mathscr{F}}}{D_{\max}^{\mathscr{E}^*}} = \frac{6}{2(1+\epsilon)} \to 3 = 2 + 2 \times \frac{1}{2} = 2 + \alpha_{\max} F \quad (\epsilon \to 0).$$

To be more intuitive, we propose Figure 3 to show the corresponding schedules.

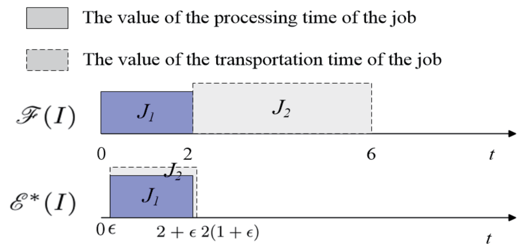

Figure 3: The corresponding schedules.

In view of the lower bound, together with Examples 4.13 and 4.14, we conclude the final conclusion as below:

Theorem 4.15. *For problem $\mathcal{P}_3$, if $\alpha_{\max}F \leq 1$, then the algorithm $\mathscr{A}_4$ is an online algorithm with competitive ratio $\varphi(E)$, where $\varphi(E) = 1 + \alpha_{\max}F$ or $\varphi(E) = 2 + \alpha_{\max}F$ depending on whether $E = 0$ or $E > 0$.*

5 Conclusions

In this paper, three online scheduling problems with deterioration on a single processor are discussed. Our goals are to minimize the makespan and minimize the maximum delivery time. It is novel and realistic to consider such problems. For clarity, the results proposed in the work will be shown in Table 5.

Table 5: The results of the work.

Problem	Lower bound	Competitive ratio	Optimal
$\mathcal{P}_1$		1	$\surd$
$\mathcal{P}_2$	$\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$	$\frac{1+2\beta_{\max}F}{1+\beta_{\max}F}$	$\surd$
$\mathcal{P}_3(\alpha_{\max}F > 1)$	2	2	$\surd$
$\mathcal{P}_3(\alpha_{\max}F \leq 1)$	$1 + \alpha_{\max}F$	$\varphi(E)$	

Notice that, for problem $1|\text{online}, r_l, q_l = \alpha_l(E + FC_l)|D_{\max}$, there exists an unsatisfactory gap between the competitive ratio and the lower bound when $\alpha_{\max}F \leq 1$. Hence, it is necessary to study it further. In addition, it will be interesting to generalize the results to multi-processor scheduling. Furthermore, it is meaningful to investigate other objectives.

References

[1] S. Browne and U. Yechiali, Scheduling deteriorating jobs on a single processor, *Oper. Res.* 38 (1990) 495–498.

[2] F. Camci, Maintenance scheduling of geographically distributed assets with prognostics information, *European J. Oper. Res.* 245 (2015) 506–516.

[3] X. Chai, W. Li, H. Yuan and L. Wang, Online scheduling on a single machine with linear deteriorating processing times and delivery times, *J. Comb. Optim.* (2020) 1–13.

[4] T.C.E. Cheng and Q. Ding, The complexity of scheduling starting time dependent tasks with release times, *Inform. Process. Lett.* 65 (1998) 75–79.

[5] Y. Cheng and S. Sun, Scheduling linear deteriorating jobs with rejection on a single machine, *European J. Oper. Res.* 194 (2009) 18–27.

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

[7] R.L. Daniels, J.B. Mazzola and D. Shi, Flow shop scheduling with partial resource flexibility, *Manage. Sci.* 50 (2004) 658–669.

[8] Y. Fang, X. Lu and P. Liu, Online batch scheduling on parallel machines with delivery times, *Theor. Comput. Sci.* 412 (2011) 5333–5339.

[9] A. Fiat and G.J. Woeginger, Competitive analysis of algorithms, in: Developments from A June Seminar on Online Algorithms: the State of the Art, 1998.

[10] 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, *Ann. Discret. Math.* 5 (1979) 287–326.

[11] J.N.D. Gupta and S.K. Gupta, Single facility scheduling with nonlinear processing times, *Comput. Ind. Eng.* 14 (1988) 387–393.

[12] H. Hoogeveen, M. Skutella and G.J. Woeginger, Preemptive scheduling with rejection, *Math. Program.* 94 (2-3) (2003) 361–374.

[13] J.A. Hoogeveen and A.P.A. Vestjens, Optimal on-line algorithms for single-machine scheduling, in: *International Conference on Integer Programming* & *Combinatorial Optimization*, Springer, Berlin Heidelberg, 1996.

[14] J.A. Hoogeveen and A.P.A. Vestjens, A best possible deterministic on-line algorithm for minimizing maximum delivery time on a single machine, *SIAM J. Discrete Math.* 13 (2000) 56–63.

[15] M.Y. Kovalyov and D. Šešok, Two-agent scheduling with deteriorating jobs on a single parallel-batching machine: refining computational complexity, *J. Sched.* 22 (2019) 603–606.

[16] K. Lee, F. Zheng and M. Pinedo, Online scheduling of ordered flow shops, *European J. Oper. Res.* 272 (2019) 50–60.

[17] M. Liu, F. Zheng, S. Wang and J. Huo, Optimal algorithms for online single machine scheduling with deteriorating jobs, *Theor. Comput. Sci.* 445 (2012) 75–81.

[18] P. Liu and X. Lu, Online scheduling on two parallel machines with release dates and delivery times, *J. Comb. Optim.* 30 (2) (2015) 347–359.

[19] R. Ma, J. Tao and J. Yuan, Online scheduling with linear deteriorating jobs to minimize the total weighted completion time, *Appl. Math. Comput.* 273 (2016) 570–583.

[20] G. Mosheiov, Scheduling jobs under simple linear deterioration, *Comput. Oper. Res.* 21 (1994) 653–659.

[21] S.G. Nurre and T.C. Sharkey, Online scheduling problems with flexible release dates: Applications to infrastructure restoration, *Comput. Oper. Res.* 92 (2017) 1-16.

[22] C. Wang, Y. Zhang and C. Miao, Two models of single machine scheduling with general processing time functions, *Oper. Res. Trans.* 15 (2) (2011) 59–67.

[23] H. Yang, D. Zhang, K. Meng, Z.Y. Dong and M. Lai, Multi-network combined cooling heating and power system scheduling considering emission trading, *Pacific J. Optim.* 10 (1) (2014) 177-198.

[24] Y. Yin, Y. Wang, T.C.E. Cheng, W. Liu and J. Li, Parallel-machine scheduling of deteriorating jobs with potential machine disruptions, *Omega Int. J. Manage. S.* 69 (2017) 17–28.

Manuscript received 14 November 2021
revised 15 May 2022
accepted for publication 4 July 2022

Sainan Guo
School of Mathematical Science & Institute of Mathematics
Nanjing Normal University
Nanjing 210023, People's Republic of China
E-mail address: athenagsn@163.com

Yannan Chen
School of Mathematical Sciences
South China Normal University
Guangzhou 510631, People's Republic of China
E-mail address: ynchen@scnu.edu.cn.

Vincent Chau
School of Computer Science and Engineering
Southeast University
Nanjing 210023, People's Republic of China
E-mail address: vincentchau@seu.edu.cn

Yaping Mao
Department of Mathematics
Qinghai Normal University
Xining, Qinghai 810008, People's Republic of China;

Academy of Plateau Science and Sustainability
Xining, Qinghai 810008, People's Republic of China
E-mail address: maoyaping@ymail.com

Xiaoyan Zhang
School of Mathematical Science & Institute of Mathematics
Nanjing Normal University
Nanjing 210023, People's Republic of China
E-mail address: zhangxiaoyan@njnu.edu.cn