Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 805294, 7 pages
http://dx.doi.org/10.1155/2015/805294

Research Article

Online Scheduling on a Single Machine with Grouped Processing Times

Qijia Liu,[1] Long Wan,[2] and Lijun Wei[2]

[1] *School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China*
[2] *School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China*

Correspondence should be addressed to Long Wan; cocu3328@163.com

Received 17 October 2014; Accepted 10 November 2014

Academic Editor: Shuenn-Ren Cheng

We consider the online scheduling problem on a single machine with the assumption that all jobs have their processing times in $[p, (1+\alpha)p]$, where $p > 0$ and $\alpha = (\sqrt{5}-1)/2$. All jobs arrive over time, and each job and its processing time become known at its arrival time. The jobs should be first processed on a single machine and then delivered by a vehicle to some customer. When the capacity of the vehicle is infinite, we provide an online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$. When the capacity of the vehicle is finite, that is, the vehicle can deliver at most c jobs at a time, we provide another best possible online algorithm with the competitive ratio of $(\sqrt{5}+1)/2$.

1. Introduction and Problem Formulation

Production and distribution operations are two important segments in a supply chain; it is critical to integrate these two segments in a scheduling problem. In the last few years, scheduling problem with job delivery coordination is a very important topic. For the offline version, the earliest research of this topic is the one by Potts [1]. They considered the single machine scheduling problem to minimize the maximum delivery completion time and provided a heuristic algorithm with the worst-case performance ratio of $3/2$. For more papers, the reader may refer to [2–6].

For the online scheduling, there are three models which are commonly considered [7]. The first one assumes that there exists no release date and the jobs arrive in a list. The second one assumes that the processing time of each job is unknown until the job finishes. The third one assumes that each job arrives over time. When the machine is idle, an online algorithm decides which one of the available jobs is scheduled, if any. In this paper, we study the third model in which jobs arrive over time. For an online algorithm H, we use the competitive ratio

$$\rho_H = \sup\left\{\frac{H(I)}{\mathrm{OPT}(I)} : I \text{ is an instance with } \mathrm{OPT}(I) > 0\right\} \quad (1)$$

to measure the performance of H, where, for an instance I, $H(I)$ is used to denote the objective value of the schedule generated by the online algorithm H and $\mathrm{OPT}(I)$ is the objective value of an offline optimal schedule.

For the online version of scheduling problem with job delivery, Hoogeveen and Vestjens [8] considered the online single machine scheduling problem. They provided a best possible online algorithm with the competitive ratio of $(\sqrt{5}+1)/2$. When restarts are allowed, van den Akker et al. [9] proposed a best possible online algorithm with the competitive ratio of $3/2$. The online integrated production-distribution scheduling problems to minimize the sum of the total weighted flow time and the total delivery cost were considered by Averbakh [10]. For several cases of the problem, the author provided efficient online algorithms and used competitive analysis to study their worst-case performance ratio. Yuan et al. [11] studied a single batch machine online scheduling problem with restricted delivery times; that is, the delivery time of each job is less than or equal to its processing time. They provided a best possible online algorithm with competitive ratio of $(\sqrt{5}+1)/2$. For the case in which the delivery time of each job is greater than or equal to its processing time, Tian et al. [12] proposed a best possible online algorithm with competitive ratio of

$(\sqrt{5}+1)/2$. Liu et al. [13] studied online scheduling problem on a single machine with bounded distribution. They proved a lower bound of competitive ratios for all online algorithms and proposed a best possible online algorithm with a competitive ratio of $(1/2)(\sqrt{5+\beta^2+2\beta}+1-\beta)$. Liu et al. [14] studied online scheduling problems on a single machine with deteriorating jobs. For each model, they presented an optimal online algorithm.

Meanwhile, the scheduling problems in which the jobs have the equal length processing times are widely considered; the reader may refer to [15–17]. However, it is more reasonable to suppose that the jobs have their processing times being in a given interval. In semiconductor manufacturing integrated circuits are produced through the same technical processes. Therefore they have their processing times in a given interval. Fang et al. [18] studied online scheduling problems on a batch machine with the assumption that all jobs have their processing times in $[p,(1+\alpha)p]$, where $p > 0$ and $\alpha = (\sqrt{5}-1)/2$. For the two scheduling problems, they provided best possible online algorithms with the competitive ratio of $(\sqrt{5}+1)/2$.

In this paper, we study the single machine online scheduling problem with job delivery in which the jobs have their processing times in $[p,(1+\alpha)p]$, where $p > 0$ and $\alpha = (\sqrt{5}-1)/2$. The jobs should be first processed on a single machine and then delivered by a vehicle to some customer. When the capacity of the vehicle is infinite, we provide an online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$. When the capacity of the vehicle is finite and fixed, that is, the vehicle can deliver at most c jobs to the customer at a time and $c \geq 2$, we provide another online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$.

2. Problem Formulation and Preliminaries

In the online scheduling problem on a single machine to minimize the maximum delivery completion time, there are n jobs $J_1,\ldots,J_n$ which arrive over time. We do not know any information of these jobs in advance, including the number of the jobs and processing time p_j and arrival time r_j of each job J_j. The jobs are first processed on the single machine. Once the vehicle is available, the completed jobs can be delivered immediately in batches by the vehicle to some customer. Let T be the round-trip transportation between the machine and the customer. Since the customer is unknown in advance, we assume that T is known immediately once the first job arrives. The problem can be denoted by $1 \rightarrow D \mid \text{online}, r_j, v = 1, c, p_j \in [p,(1+\alpha)p] \mid D_{\max}$. Here "$1 \rightarrow D$" means that jobs are first processed on a single machine and then the completed jobs must be delivered in batches to the customer; "$v=1$" means that there is one vehicle to deliver the jobs. "c" means the capacity of the vehicle; "$p_j \in [p,(1+\alpha)p]$" means that the jobs have their processing times in $[p,(1+\alpha)p]$. We use D_j to denote the time at which the vehicle transports J_j to the customer and returns to the machine. $D_{\max} = \max_{j=1}^{n}\{D_j\}$ denotes the time when the vehicle finishes delivering the last delivery batch to the destination and returns to the machine.

Indeed, when $c = 1$ and T is known in advance, Ng and Lu [19] provided an optimal online algorithm for the general case in which the processing time of each job is a nonnegative number. When $c \geq 2$ and T is known in advance, they provided an online algorithm which is not best possible. However, in this paper, we assume that $c \geq 2$ and T is unknown in advance. We deal with two variants of the online scheduling problem: $1 \rightarrow D \mid \text{online}, r_j, v = 1, c = \infty, p_j \in [p,(1+\alpha)p] \mid D_{\max}$ when the capacity of the vehicle is infinite and $1 \rightarrow D \mid \text{online}, r_j, v = 1, c < \infty, p_j \in [p,(1+\alpha)p] \mid D_{\max}$ when the capacity of the vehicle is finite and fixed.

Let σ be a schedule generated by an online algorithm H. Let B_k be a delivery batch in σ. Then we define the following notation:

(i) $s_j(\sigma)$, the starting time of J_j on the machine in σ;

(ii) $c_j(\sigma)$, the completion time of J_j on the machine in σ;

(iii) $s(B_k) = \min\{s_j(\sigma) : J_j \in B_k\}$;

(iv) $c(B_k) = \max\{c_j(\sigma) : J_j \in B_k\}$;

(v) $p(B_k) = \sum_{J_j \in B_k} p_j$;

(vi) $U(t)$, the set of the jobs which are available and unprocessed at time t;

(vii) $A(t)$, the set of jobs which are completed on the machine and are waiting for the vehicle to be delivered at time t;

(viii) $\rho(B_k)$, the ready time of delivery batch B_k, which is the maximum completion time of the jobs assigned to batch B_k; in fact $\rho(B_k) = c(B_k)$;

(ix) $\delta(B_k)$, the departure time for the vehicle to delivery batch B_k from the machine to the customer; note that $\delta(B_k) \geq \rho(B_k)$ in any feasible solution;

(x) $D(B_k) = \delta(B_k) + T$, the delivery completion time of the delivery batch B_k;

(xi) we say that a set of delivery batches $\mathscr{B} = \{B_1,\ldots,B_k\}$ in this order is continuously delivered in a schedule if the time interval $[\delta(B_1), D(B_k)]$ is totally occupied by the vehicle for delivering the batches in $\mathscr{B}$; equivalently, we have $\delta(B_i) = D(B_{i-1}) = \delta(B_{i-1}) + T$ for $2 \leq i \leq k$;

(xii) suppose that $\mathscr{B}' = \{B'_1,\ldots,B'_k\}$ in this order is the set of delivery batches in a schedule; we say that there exists an idle time of the vehicle directly before $\delta(B'_i)$ in a schedule if $D(B'_{i-1}) < \delta(B'_i)$, for $2 \leq i \leq k$; similarly, we say that there exists no idle time of the vehicle directly before $\delta(B'_i)$ in a schedule if $D(B'_{i-1}) = \delta(B'_i)$, for $2 \leq i \leq k$.

3. A Lower Bound

We assume that $c \geq 2$. Let ε be a sufficiently small positive number that tends to zero. Let N be a sufficiently large integer. For any online algorithm H, we consider the following job instance generated by the adversary. Let $D^H_{\max}$ and $D^*_{\max}$ be the objective values obtained by algorithm H and optimal offline algorithm, respectively. All jobs in the instance have a processing time p. Let $T = Np$. At time 0, the adversary releases a job J_1. Assume that algorithm H starts processing J_1 at time t. If $t \geq \alpha(p+T)$, then no other jobs arrive. Clearly, we have $D^H_{\max} \geq \alpha(p+T) + p + T$. However, the optimal schedule is to process J_1 at time 0 and deliver it at time p. Then we have $D^*_{\max} = p + T$. Thus,

$$\frac{D^H_{\max}}{D^*_{\max}} \geq \frac{\alpha(p+T)+p+T}{p+T} = 1+\alpha. \qquad (2)$$

Suppose in the following that $t < \alpha(p+T)$; we assume that the vehicle delivers J_1 at time s. If $s \geq \alpha(p+T)$, then no other jobs arrive. Then we have

$$\frac{D^H_{\max}}{D^*_{\max}} \geq \frac{\alpha(p+T)+T}{p+T} = \frac{\alpha(N+1)p+Np}{(N+1)p} \longrightarrow 1+\alpha, \quad (3)$$

when $N \to +\infty$. If $s < \alpha(p+T)$, then J_2 arrives at time $s+\varepsilon$. Clearly, we have $D^H_{\max} \geq s+T+T$. Note that $s \geq p$. The optimal schedule is to process J_2 at time $s+\varepsilon$ and then to deliver J_1 and J_2 together; that is, $D^*_{\max} = s+\varepsilon+p+T$. Thus, we have

$$\begin{aligned}\frac{D^H_{\max}}{D^*_{\max}} &\geq \frac{s+T+T}{s+\varepsilon+p+T} \longrightarrow \frac{s+T+T}{s+p+T} \\ &= 1 + \frac{T-p}{s+p+T} > 1 + \frac{(N-1)p}{(\alpha+1)(N+1)p} \\ &\longrightarrow 1 + \frac{1}{1+\alpha} = 1+\alpha,\end{aligned} \qquad (4)$$

when $\varepsilon \to 0$ and $N \to +\infty$.

The above discussion implies the following two lemmas.

Lemma 1. *For the problem $1 \to D \mid online, r_j, v = 1, c = \infty, p_j \in [p,(1+\alpha)p] \mid D_{\max}$, there exists no online algorithm with a competitive ratio of less than $1+\alpha$.*

Lemma 2. *For the problem $1 \to D \mid online, r_j, v = 1, c < \infty, p_j \in [p,(1+\alpha)p] \mid D_{\max}$, where $c \geq 2$, there exists no online algorithm with a competitive ratio of less than $1+\alpha$.*

4. The Case of $c = \infty$

In this section, we consider the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty, p_j \in [p,(1+\alpha)p] \mid D_{\max}$, where the capacity of the vehicle is infinite; that is, all jobs can be delivered simultaneously in the same batch.

Algorithm D^∞

Processing Stage. At time t, if the machine is available and $U(t) \neq \emptyset$, we choose the job of $U(t)$ with the shortest processing time to process at time t. Otherwise, do nothing but wait.

Delivery Stage. At time t with $A(t) \neq \emptyset$, we consider $A(t)$ as a delivery batch. If the machine is available, $t \geq \alpha T$, and $U(t) = \emptyset$, we deliver the delivery batch $A(t)$ at time t. Otherwise, do nothing but wait.

Indeed, the algorithm of the processing stage of D^∞ is optimal for the problem $1 \mid \text{online}, r_j \mid C_{\max}$, because the algorithm cannot produce unnecessary idle time on the machine. For the delivery stage, the machine is available which means that there is not a job which is processing at this time and $U(t) = \emptyset$ means that there is not a job which is available and unprocessed at time t. That is, only if there is not an available job which is unprocessed or processing, the vehicle determines to deliver.

Let σ and π be the schedules that are generated by algorithm D^∞ and optimal offline algorithm, respectively. Let $D_{\max}(\sigma)$ and $D_{\max}(\pi)$ be the objective values obtained from the schedule σ and the optimal offline schedule π, respectively. Let $C_{\max}(\sigma)$ and $C_{\max}(\pi)$ be the makespans of the schedule σ and the schedule π, respectively. For the problem $1 \mid \text{online}, r_j \mid C_{\max}$, let $C^*_{\max}$ be the optimal offline makespan. Next, we get the following lemma.

Lemma 3. *Consider the following*

$$C^*_{\max} = C_{\max}(\sigma). \qquad (5)$$

Proof. Since the algorithm of processing stage of D^∞ cannot produce an unnecessary idle time on the machine, that is, the machine is always busy as long as there are jobs which are available and unprocessed, we have $C^*_{\max} = C_{\max}(\sigma)$. □

As the capacity of the vehicle is infinite, we have the following lemma.

Lemma 4. *Consider the following*

$$D_{\max}(\pi) \geq C_{\max}(\pi) + T \geq C^*_{\max} + T \geq r_l + T. \qquad (6)$$

Theorem 5. *The competitive ratio of algorithm D^∞ is at most $1+\alpha$; that is, $D_{\max}(\sigma)/D_{\max}(\pi) \leq 1+\alpha$.*

Proof. Let $B_1, \ldots, B_k$ be the delivery batches in σ. Set $n_B = |\{B_1, \ldots, B_k\}|$. Assume that the last arrival time of the jobs is r_l. If $\delta(B_k) = \alpha T$, then we have $n_B = 1$. Thus $D_{\max}(\sigma) = \delta(B_k) + T \leq (1+\alpha)D_{\max}(\pi)$, as required. Suppose in the following that $\delta(B_k) > \alpha T$. If $\delta(B_k) = C_{\max}(\sigma)$, from Lemmas 3 and 4, we have

$$D_{\max}(\sigma) = C_{\max}(\sigma) + T = C^*_{\max} + T \leq D_{\max}(\pi). \qquad (7)$$

If $\delta(B_k) \neq C_{\max}(\sigma)$, that is, the delivery batch B_k follows immediately behind the delivery batch B_{k-1}, then we have $\delta(B_k) = \delta(B_{k-1}) + T$. Thus,

$$D_{\max}(\sigma) = \delta(B_{k-1}) + 2T. \qquad (8)$$

As the capacity of the vehicle is infinite, by algorithm D^∞, we have $r_l > \delta(B_{k-1})$. According to the algorithm D^∞, we have $\delta(B_j) \geq \alpha T$, for all $1 \leq j \leq k$. Then

$$D_{\max}(\pi) \geq r_l + T > \delta(B_{k-1}) + T \geq \alpha T + T. \qquad (9)$$

Thus, we have $T \leq \alpha D_{\max}(\pi)$. From Lemma 4, (8) can be rewritten as

$$\begin{aligned} D_{\max}(\sigma) &= \delta(B_{k-1}) + 2T < r_l + 2T \\ &\leq D_{\max}(\pi) + \alpha D_{\max}(\pi). \end{aligned} \qquad (10)$$

The result follows. □

Theorem 5 and Lemma 1 imply that D^{∞} is a best possible online algorithm for the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$. Indeed, according to the proof of Theorem 5, D^{∞} is also a best possible online algorithm for the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty \mid D_{\max}$.

5. The Case of $c < \infty$

In this section, we consider the problem $1 \to D \mid \text{online}, r_j, v = 1, c < \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$, where the capacity of the vehicle is finite and fixed; that is, the vehicle can deliver at most c jobs in the same batch to some customer at a time.

Algorithm D^c

Processing Stage. At time t, if the machine is available and $U(t) \neq \emptyset$, we choose the job of $U(t)$ with the shortest processing time to process at time t. Otherwise, do nothing but wait.

Delivery Stage

Step 0. If the vehicle is idle, $t \geq \alpha T$, and $A(t) \neq \emptyset$, determine $|A(t)|$ to be the number of the jobs of $A(t)$.

Step 1. If $|A(t)| \geq c$, select the earliest completed c jobs in $A(t)$ as a delivery batch and deliver this batch at time t.

Step 2. If $0 < |A(t)| < c$, then go to the following steps.

Step 2.A. If the machine is available and $U(t) = \emptyset$, that is, there is not a job which is unprocessed or processing at time t, we consider jobs in $A(t)$ as a delivery batch and deliver this batch at time t.

Step 2.B. If the machine is busy or $U(t) \neq \emptyset$, that is, there exists a job which is unprocessed or processing at time t, wait until the machine is available and $U(t) = \emptyset$ or the next arrival.

Step 3. Go to Step 0.

Similarly, the algorithm of the processing stage of D^c is optimal for the problem $1 \mid \text{online}, r_j \mid C_{\max}$, because the algorithm cannot produce unnecessary idle time. For the delivery stage, the machine is available which means that there is not a job which is processing at this time and $U(t) = \emptyset$ means that there is not a job which is available and unprocessed at time t. That is, only if there is not a job which is unprocessed or processing the vehicle determines to deliver. Meanwhile, the algorithm implies that the job with a smaller completion time is delivered no later than that with a larger completion time.

Let μ and π be the schedules that are generated by algorithm D^c and optimal offline algorithm, respectively. Let $D_{\max}(\mu)$ and $D_{\max}(\pi)$ be the objective values obtained from the schedule μ and the optimal offline schedule π, respectively. Let $C_{\max}(\mu)$ and $C_{\max}(\pi)$ be the makespans of the schedule μ and the schedule π, respectively. For the problem $1 \mid \text{online}, r_j \mid C_{\max}$, let $C^*_{\max}$ be the optimal offline makespan.

As the algorithm of processing stage of D^{∞} is same as the algorithm of processing stage of D^c, we also have the following lemma.

Lemma 6. *Consider the following*

$$C^*_{\max} = C_{\max}(\mu). \tag{11}$$

As the capacity of the vehicle is finite and fixed, there is at least one delivery batch. Let n be the number of the jobs in the schedule. Let $k^* = \lceil n/c \rceil$. Then there are at least k^* delivery batches in any feasible schedule. Thus the following lemma can be observed.

Lemma 7. *Consider the following*

$$D_{\max}(\pi) \geq \max\{C_{\max}(\pi) + T, p + k^* T\}. \tag{12}$$

A delivery batch is called *full* if it contains exactly c jobs. Otherwise, it is *nonfull*. Let $B_1, \ldots, B_k$ be the delivery batches in μ. Assume that the last arrival time of the jobs is r_l.

Lemma 8. *If $T \leq p$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. As $T \leq p$, we have $D_{\max}(\mu) = C_{\max}(\mu) + T$. From Lemmas 6 and 7, we have

$$\begin{aligned} D_{\max}(\mu) &= C_{\max}(\mu) + T \\ &= C^*_{\max} + T \\ &\leq C_{\max}(\pi) + T \leq D_{\max}(\pi). \end{aligned} \tag{13}$$

The lemma follows. □

Lemma 9. *If $T > p$ and $C_{\max}(\mu) \leq \alpha T$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. Note that $C_{\max}(\mu) \leq \alpha T$. According to the algorithm of delivery stage of D^c, we have $\delta(B_1) = \alpha T$ and $\delta(B_j) > \alpha T$ for all $2 \leq j \leq k$. Then we get $k = k^*$. Thus,

$$D_{\max}(\mu) = \delta(B_1) + kT = \alpha T + k^* T \leq (1+\alpha) D_{\max}(\pi). \tag{14}$$

The lemma follows. □

Lemma 10. *If $T > p$ and $C_{\max}(\mu) > \alpha T$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. Let B_t be the earliest delivery batch such that $B_t, \ldots, B_k$ are continuous delivery batch in μ. If $B_t = B_k$, as $C_{\max}(\mu) > \alpha T$, we have $\delta(B_k) = C_{\max}(\mu)$; then $D_{\max}(\mu) = C_{\max}(\mu) + T \leq D_{\max}(\pi)$, as required.

If $B_t \neq B_k$ and $\delta(B_{k-1}) \leq C_{\max}(\mu)$, as $C_{\max}(\mu) > \alpha T$, we have $C_{\max}(\pi) \geq C^*_{\max} = C_{\max}(\mu) > \alpha T$. Then

$$D_{\max}(\pi) \geq C_{\max}(\pi) + T \geq (\alpha + 1) T. \tag{15}$$

Thus, we have $D_{\max}(\mu) = \delta(B_{k-1}) + 2T \leq C_{\max}(\mu) + 2T \leq D_{\max}(\pi) + T \leq (1+\alpha) D_{\max}(\pi)$, as required. Suppose in the following that $B_t \neq B_k$ and $\delta(B_{k-1}) > C_{\max}(\mu)$. Then B_{k-1} follows immediately behind the delivery batch B_{k-2}, and so $|\{B_t, \dots, B_k\}| \geq 3$. Now, there are two possibilities to be discussed.

Case 1. There are some nonfull delivery batches in $B_t, \dots, B_{k-1}$. Denote the last nonfull delivery batch before B_k by B_s, where $t \leq s \leq k-1$. Obviously, we have $\delta(B_s) \geq \alpha T$ and $D_{\max}(\mu) = \delta(B_s) + (k-s+1)T$. As B_s is nonfull, by algorithm D^c, we have $U(\delta(B_s)) = \emptyset$ and there is not a job which is processing at time $\delta(B_s)$. Then, for each $s+1 \leq i \leq k$, the jobs of B_i arrive after $\delta(B_s)$. Note that $B_{s+1}, \dots, B_{k-1}$ are full delivery batches. Thus, we have

$$D_{\max}(\pi) \geq \delta(B_s) + (k-s)T. \tag{16}$$

From (15) and (16), we have

$$\begin{aligned} D_{\max}(\mu) &= \delta(B_s) + (k-s+1)T \\ &\leq D_{\max}(\pi) + T \leq (1+\alpha) D_{\max}(\pi), \end{aligned} \tag{17}$$

as required.

Case 2. In μ, all the delivery batches $B_t, \dots, B_{k-1}$ are full. As $|\{B_t, \dots, B_k\}| \geq 3$, we have $k^* \geq 3$. From Lemma 7, we have $T \leq (1/3)D_{\max}(\pi)$.

Case 2.1 ($\delta(B_t) = \alpha T$). Then $B_t = B_1$ and $D_{\max}(\mu) = \delta(B_1) + kT$. As all the delivery batches $B_t, \dots, B_{k-1}$ are full, we have $k = k^*$. Then $D_{\max}(\mu) = \delta(B_1) + kT = \alpha T + k^* T \leq (1+\alpha)D_{\max}(\pi)$, as required.

Case 2.2 ($\delta(B_t) > \alpha T$). The jobs in B_t are processed on the machine in this interval $[s(B_t), c(B_t)]$. Note that B_t is the earliest delivery batch such that $B_t, \dots, B_k$ are continuous delivery batch in μ. Then there exists an idle time of the vehicle directly before $\delta(B_t)$. Then $\delta(B_t) = \rho(B_t) = c(B_t)$. By algorithm D^c, the job with a smaller completion time is delivered no later than that with a larger completion time; then $s(B_{t+1}) \geq c(B_t)$. Noting that $B_t, \dots, B_k$ are continuous delivery batch, we have $\delta(B_{t+1}) = \delta(B_t) + T$. However, $p_j \in [p, (1+\alpha)p]$, for all $1 \leq j \leq n$, and B_{t+1} is full; we have

$$\begin{aligned} \delta(B_t) + T = \delta(B_{t+1}) &\geq c(B_{t+1}) \\ &\geq s(B_{t+1}) + cp \geq c(B_t) + cp. \end{aligned} \tag{18}$$

Thus, we have

$$p \leq \frac{T}{c}. \tag{19}$$

If there exists an idle time in $[s(B_t), c(B_t)]$ on the machine in μ, assume that s_i is the earliest time such that there exists no idle time in $[s_i, c(B_t)]$, where $s_i > s(B_t)$. Then s_i is the arrival time of some job. By algorithm D^c, all jobs of $B_{t+1}, \dots, B_k$ arrive at or after s_i. Assume that there are m jobs which are processed in $[s_i, c(B_t)]$ on the machine, where $m < c$. Noting that $B_{t+1}, \dots, B_{k-1}$ are full, then there are at least $m + (k-t-1)c + 1$ jobs which arrive at or after s_i. In fact, these jobs need at least $k-t$ delivery batches to be delivered. Thus, we have

$$D_{\max}(\pi) \geq s_i + mp + (k-t)T. \tag{20}$$

As $p_j \in [p, (1+\alpha)p]$, for all $1 \leq j \leq n$, we have

$$\begin{aligned} D_{\max}(\mu) &= \delta(B_t) + (k-t+1)T \\ &\leq s_i + m(1+\alpha)p + (k-t+1)T. \end{aligned} \tag{21}$$

From (19), (20), and (21), we have

$$\begin{aligned} D_{\max}(\mu) - D_{\max}(\pi) &\leq m\alpha p + T \leq \alpha T + T \\ &\leq \frac{1+\alpha}{3} D_{\max}(\pi) < \alpha D_{\max}(\pi), \end{aligned} \tag{22}$$

as required.

If there exists no idle time in $[s(B_t), c(B_t)]$ on the machine in μ, assume that s_0 is the earliest time such that there exists no idle time in $[s_0, c(B_t)]$. Obviously, $s_0 \leq s(B_t)$ and s_0 is the arrival time of some job. By algorithm D^c, all jobs of $B_{t+1}, \dots, B_k$ arrive at or after s_0. Assume that there are n' jobs which arrive at or after s_0 and there are z jobs which are processed on the machine in $[s_0, c(B_t)]$, where $z \geq c$. Then there are $n' - z$ jobs which are processed on the machine in $[c(B_t), C_{\max}(\mu)]$. As $p_j \in [p, (1+\alpha)p]$, we have

$$D_{\max}(\mu) \leq s_0 + z(1+\alpha)p + (k-t+1)T. \tag{23}$$

Let $n' = k'c + \nu$, where $n' \leq n$ and $k' \leq k^* - 1$. Then

$$D_{\max}(\pi) \geq s_0 + \nu p + \left(k'+1\right)T. \tag{24}$$

In addition, $n' - z$ jobs which are processed in $[c(B_t), C_{\max}(\mu)]$ are delivered by $k-t$ delivery batches, because these jobs are delivered in $B_{t+1}, \dots, B_k$. Then $\lceil (n'-z)/c \rceil = k-t$; that is, $k-t-1 < (n'-z)/c \leq k-t$. Hence, $k'c + \nu - z = n' - z > c(k-t-1)$; that is,

$$c(k-t-1) - \left(k'c + \nu\right) < -z. \tag{25}$$

On the other hand, as $|\{B_t, \dots, B_k\}| \geq 3$, we have $|\{B_{t+1}, \dots, B_k\}| \geq 2$; then $c + 1 \leq n' - z = k'c + \nu - z$; that is,

$$\frac{z}{c} < k' + \frac{\nu}{c} - 1. \tag{26}$$

Now, from (23), (24), and (25), we have

$$\begin{aligned} & D_{\max}(\mu) - D_{\max}(\pi) \\ &\quad = z(1+\alpha)p - \nu p + (k-t+1)T - \left(k'+1\right)T \\ &\quad = (z-\nu)p + \alpha z p + (k-t)T - k'T \\ &\quad \leq \frac{z-\nu}{c}T + \alpha\frac{z}{c}T + \frac{c(k-t-1) - \left(ck'+\nu\right)}{c}T \\ &\qquad + T + \frac{\nu}{c}T \\ &\quad \leq \frac{z-\nu}{c}T + \alpha\frac{z}{c}T - \frac{z}{c}T + \frac{\nu}{c}T + T \\ &\quad = \alpha\frac{z}{c}T + T = \alpha\left(\frac{z}{c} + \frac{1}{\alpha}\right)T. \end{aligned} \tag{27}$$

If $z = c$, since $T \le (1/3)D_{\max}(\pi)$, we have $D_{\max}(\mu) - D_{\max}(\pi) \le (1+\alpha)T \le \alpha D_{\max}(\pi)$, as required. If $z > c$, then there exists no idle time immediately before $s(B_t)$ on the machine. As the job with a smaller completion time is delivered no later than that with a larger completion time, we have $\rho(B_{t-1}) = c(B_{t-1}) = s(B_t)$. Furthermore, there exists an idle time of the vehicle before $\delta(B_t)$; then $s(B_t) + T = \rho(B_{t-1}) + T \le \delta(B_{t-1}) + T < \delta(B_t) = c(B_t) \le s(B_t) + c(1+\alpha)p$, implying that

$$p > \frac{\alpha}{c}T. \tag{28}$$

However,

$$\left(\alpha\frac{\nu}{c}+1\right) - \left(\frac{\nu}{c}+\alpha\right) = \frac{(c-\nu)(1-\alpha)}{c} > 0. \tag{29}$$

From (24), (26), (28), and (29), (27) can be rewritten as

$$\begin{aligned} D_{\max}(\mu) - D_{\max}(\pi) &\le \alpha\left(\frac{z}{c}+\frac{1}{\alpha}\right)T \\ &\le \alpha\left(k' + \frac{\nu}{c} - 1 + \frac{1}{\alpha}\right)T \\ &= \alpha\left(k' + \frac{\nu}{c} + \alpha\right)T \\ &\le \alpha\left(k' + \alpha\frac{\nu}{c} + 1\right)T \\ &\le \alpha\left[\nu p + \left(k'+1\right)T\right] \\ &\le \alpha D_{\max}(\pi). \end{aligned} \tag{30}$$

This completes the proof. □

Theorem 11. *For the problem*

$$\begin{gathered} 1 \longrightarrow D \mid online, \quad r_j, \nu = 1, \quad c < \infty, \\ p_j \in [p, (1+\alpha)p] \mid D_{\max}, \end{gathered} \tag{31}$$

where the capacity of the vehicle is finite and fixed, D^c is the best possible online algorithm with a competitive ratio of $1+\alpha$.

Proof. From Lemmas 8, 9, and 10, the competitive ratio of online algorithm D^c is $1+\alpha$. The result follows from Lemma 2. □

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Natural Sciences Foundation (Grant no. 20142BAB211017) of Jiangxi Province and the School Subject (Grant no. 06162015) of Jiangxi University of Finance and Economics.

References

[1] C. N. Potts, “Analysis of a heuristic for one machine sequencing with release dates and delivery times,” *Operations Research*, vol. 28, no. 6, pp. 1436–1441, 1980.

[2] Y. C. Chang and C. Y. Lee, “Machine scheduling with job delivery coordination,” *European Journal of Operational Research*, vol. 158, no. 2, pp. 470–487, 2004.

[3] Z.-L. Chen and G. L. Vairaktarakis, “Integrated scheduling of production and distribution operations,” *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.

[4] N. G. Hall and C. N. Potts, “Supply chain scheduling: batching and delivery,” *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.

[5] C.-Y. Lee and Z.-L. Chen, “Machine scheduling with transportation considerations,” *Journal of Scheduling*, vol. 4, no. 1, pp. 3–24, 2001.

[6] W. Zhong, G. Dósa, and Z. Tan, “On the machine scheduling problem with job delivery coordination,” *European Journal of Operational Research*, vol. 182, no. 3, pp. 1057–1072, 2007.

[7] K. Pruhs, J. Sgall, and E. Torng, “Online scheduling,” in *Handbook of Scheduling: Algorithm, Models, and Performance Analysis*, J. Y.-T. Leung, Ed., Chapman and Hall/CRC, Boca Raton, Fla, USA, 2004.

[8] J. A. Hoogeveen and A. P. Vestjens, “A best possible deterministic on-line algorithm for minimizing maximum delivery time on a single machine,” *SIAM Journal on Discrete Mathematics*, vol. 13, no. 1, pp. 56–63, 2000.

[9] M. van den Akker, H. Hoogeveen, and N. Vakhania, “Restarts can help in the on-line minimization of the maximum delivery time on a single machine,” *Journal of Scheduling*, vol. 3, no. 6, pp. 333–341, 2000.

[10] I. Averbakh, “On-line integrated production-distribution scheduling problems with capacitated deliveries,” *European Journal of Operational Research*, vol. 200, no. 2, pp. 377–384, 2010.

[11] J. Yuan, S. Li, J. Tian, and R. Fu, “A best on-line algorithm for the single machine parallel-batch scheduling with restricted delivery times,” *Journal of Combinatorial Optimization*, vol. 17, no. 2, pp. 206–213, 2009.

[12] J. Tian, R. Fu, and J. Yuan, “An on-line algorithm for the single machine unbounded parallel-batching scheduling with large delivery times,” *Information Processing Letters*, vol. 111, no. 21-22, pp. 1048–1053, 2011.

[13] M. Liu, C. Chu, Y. Xu, and F. Zheng, “An optimal online algorithm for single machine scheduling with bounded delivery times,” *European Journal of Operational Research*, vol. 201, no. 3, pp. 693–700, 2010.

[14] M. Liu, F. F. Zheng, S. J. Wang, and J. Z. Huo, “Optimal algorithms for online single machine scheduling with deteriorating jobs,” *Theoretical Computer Science*, vol. 445, pp. 75–81, 2012.

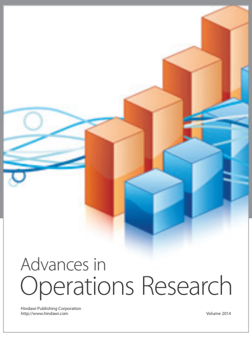
[15] Y. Lin and W. Li, “Parallel machine scheduling of machine-dependent jobs with unit-length,” *European Journal of Operational Research*, vol. 156, no. 1, pp. 261–266, 2004.

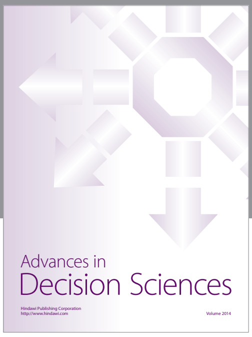
[16] W. Li and S. Li, “Online batch scheduling of equal-length jobs on two identical batch machines to maximise the number of early jobs,” *International Journal of Systems Science*, vol. 46, no. 4, pp. 652–661, 2015.

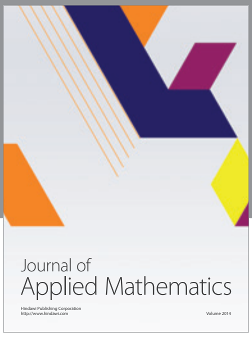
[17] W. Li, Z. Zhang, and S. Yang, “Online algorithms for scheduling unit length jobs on parallel-batch machines with lookahead,” *Information Processing Letters*, vol. 112, no. 7, pp. 292–297, 2012.

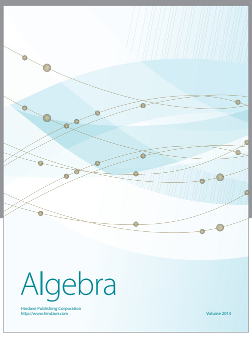
[18] Y. Fang, P. Liu, and X. Lu, “Optimal on-line algorithms for one batch machine with grouped processing times,” *Journal of Combinatorial Optimization*, vol. 22, no. 4, pp. 509–516, 2011.

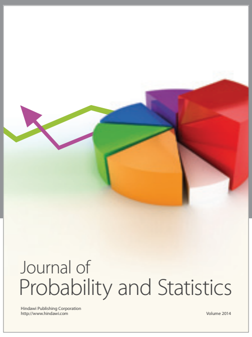
[19] C. T. Ng and L. Lu, “On-line integrated production and outbound distribution scheduling to minimize the maximum delivery completion time,” *Journal of Scheduling*, vol. 15, no. 3, pp. 391–398, 2012.


Advances in
Operations Research

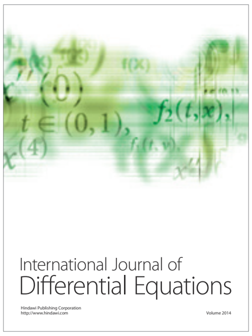

Advances in
Decision Sciences


Journal of
Applied Mathematics

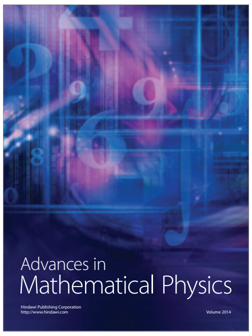

Algebra

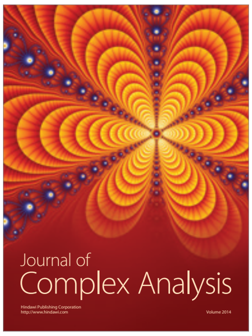

Journal of
Probability and Statistics

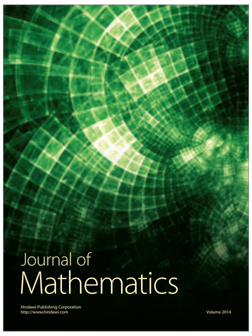
The Scientific
World Journal

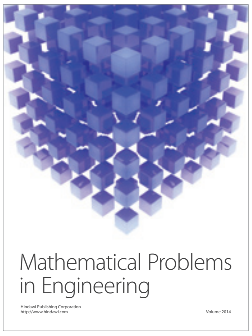

International Journal of
Differential Equations

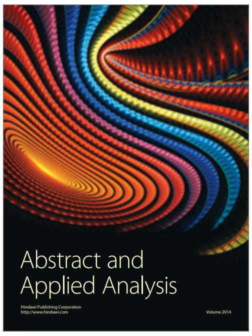
International Journal of
Combinatorics

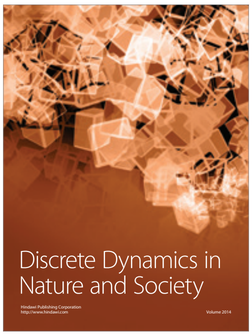

Advances in
Mathematical Physics

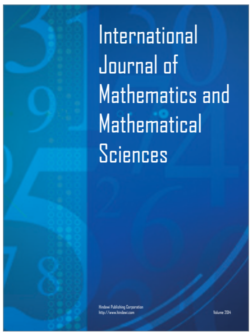

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

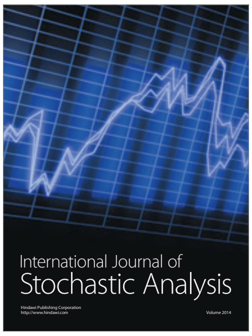

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization