Machine Scheduling with a Maintenance Interval and Job Delivery Coordination

Jueliang Hu[1], Taibo Luo[2,3], Xiaotong Su[1], Jianming Dong[1], Weitian Tong[3], Randy Goebel[3], Yinfeng Xu[2,4], and Guohui Lin[1,3](✉)

[1] Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou 310018, Zhejiang, China
[2] Business School, Sichuan University, Chengdu 610065, Sichuan, China
[3] Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Canada
guohui@ualberta.ca
[4] State Key Lab for Manufacturing Systems Engineering, Xi'an 710049, Shaanxi, China

Abstract. We investigate a scheduling problem with job delivery coordination in which the machine has a maintenance time interval. The goal is to minimize the makespan. In the problem, each job needs to be processed on the machine non-preemptively for a certain time, and then transported to a distribution center; transportation is by one vehicle with a limited physical capacity, and it takes constant time to deliver a shipment to the distribution center and return back to the machine. We present a 2-approximation algorithm for the problem, and show that the performance ratio is tight.

Keywords: Scheduling · Machine maintenance · Job delivery · Bin-packing · Approximation algorithm · Worst-case performance analysis

1 Introduction

We consider a scheduling problem that arises from supply chain management research at the operational level, with the goal to show that decision makers at different stages of a supply chain can make coordinated decisions at the detailed scheduling level, and achieve substantial efficiencies. This problem integrates production and delivery whereby the jobs are first processed in a manufacturing center, and then delivered to a distribution center. We use a machine to model the manufacturing center, which has a preventive maintenance time interval when it is unavailable for processing any jobs. Job delivery is performed by a single vehicle with a limited physical load capacity, between the manufacturing center and the distribution center. The goal is to minimize the makespan. A special case of this problem was first considered by Wang and Cheng [9], in which the jobs have uniform size.

J. Hu and T. Luo—Co-first authors.

J. Wang and C. Yap (Eds.): FAW 2015, LNCS 9130, pp. 104–114, 2015.
DOI: 10.1007/978-3-319-19647-3_10

Our target scheduling problem is formally described as follows. We are given a set of jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$, each of which needs to be processed in a manufacturing center (the machine) and then delivered to a distribution center (the customer). Each job J_i requires a non-preemptive processing time of p_i in the manufacturing center; when transported by the only vehicle to the distribution center, it occupies a fraction s_i of physical space on the vehicle. The vehicle has a normalized space capacity of 1, is initially at the manufacturing center, and needs to return to the manufacturing center after all jobs are delivered. It takes the vehicle T units of time to deliver a shipment and return back to the manufacturing center. The manufacturing center, modeled as a single machine, has a known maintenance time interval $[s, t]$, where $0 \leq s \leq t$, during which no jobs can be processed. The problem objective is to minimize the makespan, that is, the time the vehicle returning to the manufacturing center after all jobs are delivered.

Using the notation of Lee et al. [7] and following Wang and Cheng [9], the problem under study is denoted as $(1, h(1) \mid \textit{non-pmtn}, D, s_i \mid C_{\max})$. In this three-field notation, the first field denotes the machine environment, the second denotes the job characteristics, and the last denotes the performance measure to be optimized. In our case, "1" says that there is only a single machine to process the jobs and "$h(1)$" indicates that there is a hole (*i.e.* a maintenance interval) in the machine, "*non-pmtn*" states that each job needs a continuous processing or, if interrupted by the unavailable machine maintenance interval, it has to restart the processing after the machine becomes available,[1] "D" indicates the delivery requirement that jobs must be delivered to the distribution center after the processing is completed in the manufacturing center, "s_i" is the normalized physical size of job J_i on the single vehicle, and lastly, "$C_{\max}$" denotes the makespan, which is the time the vehicle returning to the manufacturing center after all jobs are delivered.

For the special case where all jobs have the same size, that is $s_i = \frac{1}{K}$ for some positive integer K, Wang and Cheng showed that $(1, h(1) \mid \textit{non-pmtn}, D, s_i = \frac{1}{K} \mid C_{\max})$ is NP-hard, and presented a $\frac{3}{2}$-approximation algorithm based on the *shortest processing time* (SPT) rule [9]. Essentially, the SPT rule sorts the jobs into a non-decreasing order of the processing time and the machine processes the jobs in this order. The intuition is to let the machine finish processing as many jobs as possible at any given time point, to optimally supply the transportation vehicle.

While the SPT rule alone works well in this special uniform-size case, it can be very bad in the general case where the jobs have different sizes. Indeed, for another extremely special case where all jobs have zero processing time, the problem $(1, h(1) \mid \textit{non-pmtn}, D, s_i \mid C_{\max})$ reduces to minimizing the number of shipments, or the classic *bin-packing* problem, which is NP-hard and APX-complete [2].

In this paper, we show that the *next-fit* (NF) algorithm [5] designed for the bin-packing problem can be employed for packing the jobs into a favorable

[1] In the literature, *non-resumable* (specified as "*nr-a*") has been used.

number of *batches*, where each batch is a shipment to be delivered by the single vehicle. This is followed by applying the SPT rule to sequence the batches with delivery coordination. We show that this algorithm has a worst-case performance guarantee of 2, and this ratio is tight. In the next section, we present the performance analysis in detail. We conclude the paper in the last section.

2 The Algorithm D-NF-SPT

In our target scheduling problem $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$ we assume the non-trivial case where $\sum_{i=1}^{n} p_i > s \geq \min_{i=1}^{n} p_i$, *i.e.* the machine maintenance interval does affect the schedule, since otherwise the problem reduces to the problem $(1 \mid D, s_i \mid C_{\max})$, which has been extensively investigated in the literature [1,4,6,8,10], and it admits a (best possible) 1.5-approximation algorithm [8]. In the 1.5-approximation algorithm, when the total size of the jobs is greater than 1 but less than or equal to 2, the jobs are packed by the NF algorithm; otherwise, the jobs are packed by the *modified first-fit decreasing* (MFFD) algorithm [3]. The resultant batches are then processed and delivered in the SPT order.

Recall that there are n jobs, and each job J_i, for $i = 1, 2, \ldots, n$, needs to be processed non-preemptively for p_i units of time on the machine, and then transported to the distribution center by a single vehicle. The machine has a known maintenance time interval $[s, t]$, during which no jobs can be processed. The job J_i has a physical size $s_i \in (0, 1]$, representing its fractional space requirement on the vehicle during the transportation. A shipment (*i.e.*, a batch, used interchangeably) can contain multiple jobs, as long as the total size of the jobs in the shipment is no greater than 1. The vehicle takes constant time T to deliver a shipment to the distribution center and return back to the machine. For ease of presentation, we use $\Delta = t - s$ to denote the length of the machine maintenance. As mentioned in the introduction, we have the following lemma due to the hardness results of the bin-packing problem.

Lemma 1. *The problem* $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$ *is NP-hard and APX-hard.* □

Let π denote a feasible schedule, in which the jobs are transported in k shipments denoted as $B_1, B_2, \ldots, B_k$ in order. We extend the notation to use $p(B_j)$ ($s(B_j)$, respectively) to denote the total processing time (size, respectively) of the jobs of B_j, for every j. Note that in general the jobs of B_j are not necessarily processed before all the jobs of B_{j+1} on the machine. Let α denote the smallest batch index such that $\sum_{j=1}^{\alpha} p(B_j) > s$. Clearly, $1 \leq \alpha \leq k$. We can assume, without loss of generality, that for every $j < \alpha$ the jobs of B_j are processed before all the jobs of B_{j+1} on the machine, and furthermore they are processed on the machine consecutively in an arbitrary order. We use δ to denote the length of the machine idling period due to the pending maintenance. It follows that the machine finishes processing all the jobs at time $\sum_{j=1}^{k} p(B_j) + \Delta + \delta$. On the other hand, the total transportation time for this schedule is kT. We assume that the vehicle does not idle if there are shipments ready to be transported.

Our algorithm D-NF-SPT can be described as follows (see Fig. 1). First (the D-step), all jobs are sorted into a non-increasing order of the ratio $\frac{s_i}{p_i}$, which we also call the *density*. Next (the NF-step), in this order, the jobs are formed into shipments (batches) by their physical sizes using the next-fit (NF) bin-packing algorithm. The NF algorithm assigns the job at the head of the order to the last (largest indexed) shipment if the job fits in, or else to a newly created shipment for the job. This way, every shipment contains a number of consecutive jobs. The achieved batch sequence is denoted as $\langle B'_1, B'_2, \ldots, B'_k \rangle$. The processing times of the shipments are then calculated, and the shipments are sorted into a non-decreasing order of the processing time (the SPT-step). The final batch sequence is denoted as $\langle B_1, B_2, \ldots, B_k \rangle$. According to this shipment order, a maximum number of batches are processed before time s; the other batches are processed starting time t (when the maintenance ends). For each shipment, its jobs are processed consecutively on the machine in an arbitrary order; and a shipment is transported to the distribution center after all its jobs are finished and the vehicle is available. We denote the achieved schedule as π, that is $\pi = \langle B_1, B_2, \ldots, B_k \rangle$ with $p(B_1) \le p(B_2) \le \ldots \le p(B_k)$. Let B_α denote the first shipment processed after time t;

$$\delta = s - \sum_{j=1}^{\alpha-1} p(B_j) \tag{1}$$

denotes the length of the machine idle time before the maintenance (see Fig. 2 for the configuration of π).

Algorithm D-NF-SPT:

Step 1. (The D-step) Sort the jobs into a non-increasing order of the ratio s_i/p_i;

Step 2. (The NF-step) Pack the jobs by size into a sequence of batches using the algorithm NF:

- 2.1. Place the current job into the last batch if it fits in;
- 2.2. Or else create a new batch for the current job;
- 2.3. The achieved batch sequence is denoted as $\langle B'_1, B'_2, \ldots, B'_k \rangle$;

Step 3. (The SPT-step) Sort the job batches into a non-decreasing order of the processing time:

- 3.1. The achieved batch sequence is denoted as $\langle B_1, B_2, \ldots, B_k \rangle$;

Step 4. Process the jobs in this batch order and deliver a finished batch as early as possible:

- 4.1. Let α denote the smallest batch index such that $\sum_{j=1}^{\alpha} p(B_j) > s$;
- 4.2. Batches $B_1, B_2, \ldots, B_{\alpha-1}$ are processed before time s;
- 4.3. Batches $B_\alpha, B_{\alpha+1}, \ldots, B_k$ are processed starting time t.

Fig. 1. A high-level description of the algorithm D-NF-SPT.

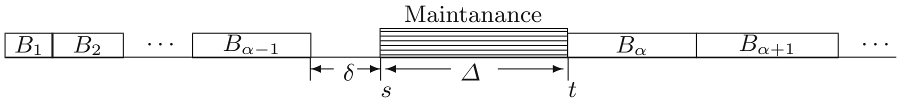

Fig. 2. A visual configuration of the schedule π produced by the D-NF-SPT algorithm.

We next prove some structural properties for the schedule π, and estimate its makespan denoted as $C_{\max}$. For ease of presentation, the finish processing time of the batch B_j on the machine is denoted as C_j, and let D_j denote the time at which the vehicle delivers the batch B_j to the distribution center and returns back to the machine. Clearly, $D_j - C_j \geq T$, for every j.

Lemma 2. *For the schedule π produced by the algorithm D-NF-SPT for the problem $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$, the makespan is*

$$C_{\max} = \begin{cases} \sum_{j=1}^{\alpha} p(B_j) + \Delta + \delta + (k - \alpha + 1)T, & \text{if } C_\alpha > D_{\alpha-1},\ C_k < D_{k-1}; \\ p(B_1) + kT, & \text{if } C_\alpha \leq D_{\alpha-1},\ C_k < D_{k-1}; \\ \sum_{j=1}^{k} p(B_j) + \Delta + \delta + T, & \text{if } C_k \geq D_{k-1}. \end{cases}$$

Proof. Recall that the machine processes the jobs of $B_1 \cup B_2 \cup \ldots \cup B_{\alpha-1}$ continuously before time s, and processes the jobs of $B_\alpha \cup B_{\alpha+1} \cup \ldots \cup B_k$ continuously after time t. Thus for the last job batch B_k, $C_k = \sum_{j=1}^{k} p(B_j) + \Delta + \delta$.

If the batch B_k has finished the processing while the vehicle is not ready for transporting it, *i.e.* $C_k < D_{k-1}$, we conclude that the vehicle does not idle during the time interval $[C_\alpha, C_k]$, where $C_\alpha = \sum_{j=1}^{\alpha} p(B_j) + \Delta + \delta$. This can be proven by a simple contradiction, as otherwise there would be a batch B_j for some $j > \alpha$, such that $C_j > D_{j-1}$. Then clearly $p(B_j) = C_j - C_{j-1} > D_{j-1} - C_{j-1} \geq T$. It follows that all the succeeding batches have a processing time greater than T. This indicates that for every successive batch, including B_k, the vehicle has to idle for a while before delivering it.

Using the same argument, if the vehicle idles inside the time interval $[C_1, C_\alpha]$ (note that the vehicle has to wait for the first batch B_1 to finish), then there must be $C_\alpha > D_{\alpha-1}$ and thus the vehicle must have delivered all the batches $B_1, B_2, \ldots, B_{\alpha-1}$ at time C_α. In this case, the makespan is $C_{\max} = \sum_{j=1}^{\alpha} p(B_j) + \Delta + \delta + (k - \alpha + 1)T$. If the vehicle does not idle before time C_α, that is, $\sum_{j=2}^{\alpha} p(B_j) + \Delta + \delta \leq (\alpha - 1)T$, then the makespan is $C_{\max} = p(B_1) + kT$.

If the job batch B_k has finished the processing and the vehicle is ready for transporting it, *i.e.* $C_k \geq D_{k-1}$, then the makespan is the finishing time of the batch B_k plus one shipment delivery time of the vehicle, which is $C_{\max} = \sum_{j=1}^{k} p(B_j) + \Delta + \delta + T$. □

From the proof of Lemma 2, we have the following corollary.

Corollary 1. *For the schedule π produced by the algorithm D-NF-SPT for the problem $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$, the vehicle idles inside the time interval $[C_1, C_\alpha]$ if and only if $C_\alpha > D_{\alpha-1}$.* □

Consider the associated instance I of the bin-packing problem to pack all the jobs of $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ by their size into the minimum number of batches (of capacity 1); let k^o denote this minimum number of batches. It is known that $k \leq 2k^o - 1$ [5], where k is the number of batches by the algorithm NF. The algorithm NF is one of the simplest approximation algorithms designed for the bin-packing problem, but not the best in terms of approximation ratio. Nevertheless, there are important properties of the packing result achieved by the algorithm NF, stated in the next two lemmas.

Lemma 3. *Consider the job batch sequence $\langle B'_1, B'_2, \ldots, B'_k \rangle$ produced by the algorithm D-NF-SPT in Step 2. Let $\mathcal{J}'$ be any subset of jobs, and assume all its jobs can be packed into k' batches. For any k_1, if $\sum_{j=1}^{k_1} s(B'_j) \leq s(\mathcal{J}')$, then $k_1 \leq 2k' - 1$.*

Proof. From the execution of the NF algorithm, we know that every two adjacent batches, B'_j and B'_{j+1}, have a total size strictly greater than 1. If k_1 is odd, then $\sum_{j=1}^{k_1} s(B'_j) > \frac{k_1-1}{2}$; otherwise, $\sum_{j=1}^{k_1} s(B'_j) > \frac{k_1}{2}$. On the other hand, every one of the k' batches has size at most 1, and thus $s(\mathcal{J}') \leq k'$. Putting together, we have

$$k' \geq s(\mathcal{J}') \geq \sum_{j=1}^{k_1} s(B'_j) > \frac{k_1 - 1}{2}.$$

That is, $k' > \frac{k_1-1}{2} + 1 = \frac{k_1+1}{2}$. This proves the lemma. □

Lemma 4. *Consider the job batch sequence $\langle B'_1, B'_2, \ldots, B'_k \rangle$ produced by the algorithm D-NF-SPT in Step 2. Let $\mathcal{J}'$ be any subset of jobs. For any k_1, if $\sum_{j=1}^{k_1} p(B'_j) > p(\mathcal{J}')$, then $\sum_{j=1}^{k_1} s(B'_j) > s(\mathcal{J}')$.*

Proof. Recall that in Step 1 of the algorithm D-NF-SPT, all the jobs of $\mathcal{J}$ are sorted by non-increasing density $\frac{s_i}{p_i}$. Assume to the contrary that $\sum_{j=1}^{k_1} p(B'_j) > p(\mathcal{J}')$ and $\sum_{j=1}^{k_1} s(B'_j) \leq s(\mathcal{J}')$. There must exist at least one job $J \in \mathcal{J}'$ but not in the first k_1 batches $B'_1, B'_2, \ldots, B'_{k_1}$, such that

$$\frac{s(J)}{p(J)} > \frac{\sum_{j=1}^{k_1} s(B'_j)}{\sum_{j=1}^{k_1} p(B'_j)} \geq \min_{J_i \in B'_{k_1}} \left\{ \frac{s_i}{p_i} \right\}.$$

However, this is a contradiction since such a job J must have been in one of the first k_1 batches $B'_1, B'_2, \ldots, B'_{k_1}$, from the execution of the NF algorithm. This proves the lemma. □

Let π^* denote an optimal schedule, in which there are k^* job batches B^*_1, B^*_2, $\ldots$, $B^*_{k^*}$, when finished, delivered in this order. We assume that the batch $B^*_{\alpha^*}$ is the first one in this order containing a job processed after time t; and use δ^* to denote the length of the machine idle time before the maintenance. It is important to note that we may assume without loss of generality that the jobs of B^*_j, for each $j < \alpha^*$, are processed continuously (in an arbitrary order), but no specific processing order for the jobs of $B^*_{\alpha^*}, B^*_{\alpha^*+1}, \ldots, B^*_{k^*}$.

The makespan of the optimal schedule π^* is denoted as $C^*_{\max}$. Again for ease of presentation, the finish processing time of the batch B^*_j on the machine is denoted as C^*_j, and let D^*_j denote the time at which the vehicle delivers the batch B^*_j to the distribution center and returns back to the machine. Clearly, $D^*_j - C^*_j \geq T$, for every j.

Lemma 5. *For the optimal schedule π^* for the problem $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$, the makespan is*

$$C^*_{\max} \geq \max\left\{ p(B^*_1) + k^*T,\ \sum_{j=1}^{k^*} p(B^*_j) + \Delta + \delta^* + T \right\}.$$

Proof. Since after the first batch B^*_1 is processed on the machine, the vehicle needs to deliver all the k^* batches; thus the makespan is at least $p(B^*_1) + k^*T$.

On the other hand, the finish processing time of the last batch $B^*_{k^*}$ is $C^*_{k^*} = \sum_{j=1}^{k^*} p(B^*_j) + \Delta + \delta^*$, and afterwards it has to be delivered; hence the makespan is at least $\sum_{j=1}^{k^*} p(B^*_j) + \Delta + \delta^* + T$. This completes the proof. □

Now we are ready to prove the main theorem.

Theorem 1. *The algorithm D-NF-SPT is an $O(n \log n)$-time 2-approximation for the problem $(1, h(1) \mid non\text{-}pmtn, D, s_i \mid C_{\max})$.*

Proof. First, if $\sum_{i=1}^{n} p_i \leq s$, *i.e.* all the jobs can be processed before the machine maintenance, the target problem reduces to the problem $(1 \mid D, s_i \mid C_{\max})$, which admits a 1.5-approximation algorithm [8]. We thus assume in the following that $\sum_{i=1}^{n} p_i > s$. Consequently, $1 \leq \alpha \leq k$ and $1 \leq \alpha^* \leq k^*$ (these four quantities are all well defined).

If in the schedule π produced by the algorithm D-NF-SPT, $C_k \geq D_{k-1}$, then by Lemma 2 the makespan is $C_{\max} = \sum_{j=1}^{k} p(B_j) + \Delta + \delta + T$. On the other hand, from Lemma 5 we have $C^*_{\max} \geq \sum_{j=1}^{k^*} p(B^*_j) + \Delta + \delta^* + T$. Clearly, $\delta \leq p(B_\alpha) \leq \sum_{j=1}^{k} p(B_j) = \sum_{j=1}^{k^*} p(B^*_j)$. It follows that

$$C_{\max} = \sum_{j=1}^{k} p(B_j) + \Delta + \delta + T \leq 2\sum_{j=1}^{k^*} p(B^*_j) + \Delta + T \leq 2C^*_{\max}.$$

That is, the makespan of the schedule π is no more than twice of the optimum.

If $k^* = 1$ in the optimal schedule π^*, that is, all the jobs can form into a single batch, then we also have $k = 1$ in the schedule π, and consequently $C_{\max} = p(B_1) + \Delta + \delta + T$. As in the previous paragraph the makespan of the schedule π is no more than twice that of the optimum.

In the following we consider $C_k < D_{k-1}$, $k \geq 2$ and $k^* \geq 2$, and we separate the discussion into two cases. Note that in the following $D_0 = 0$, meaning at the beginning the vehicle is ready.

Case 1. $C_\alpha \leq D_{\alpha-1}$.

From Corollary 1 and Lemma 2, we know that in the schedule π the vehicle does not idle inside the time interval $[C_1, C_\alpha]$ and the makespan is $C_{\max} = p(B_1) + kT$.

By letting $\mathcal{J}'$ be the whole set $\mathcal{J}$ of jobs in Lemma 3, we have $k \leq 2k' - 1 \leq 2k^* - 1$, since k' is the minimum number of batches for all the jobs of $\mathcal{J}$.

One can check that for every possible value of α, we always have $C_2 \leq D_1$ because there is no vehicle idling inside the time interval $[C_1, C_\alpha]$, and thus $p(B_1) \leq p(B_2) = C_2 - C_1 \leq D_1 - C_1 = T$. It follows from Lemma 5 that

$$C_{\max} = p(B_1) + kT \leq T + (2k^* - 1)T = 2k^*T \leq 2C^*_{\max}.$$

Case 2. $C_\alpha > D_{\alpha-1}$.

Note that we have $C_k < D_{k-1}$, and thus $\alpha \leq k - 1$. From Corollary 1 and Lemma 2, we know that in the schedule π, the vehicle idles inside the time interval $[C_1, C_\alpha]$ and the makespan is $C_{\max} = \sum_{j=1}^{\alpha} p(B_j) + \Delta + \delta + (k - \alpha + 1)T$.

If $\alpha > 2$ and the vehicle idles inside the time interval $[C_1, C_{\alpha-1}]$, then $p(B_{\alpha-1}) > T$. Consequently, $p(B_k) > T$ too, which contradicts $C_k < D_{k-1}$. In the remaining situation, either $\alpha = 1$ (*i.e.*, no jobs processed before the machine maintenance), or $2 \leq \alpha \leq k - 1$ and the vehicle idles only inside the time interval $[C_{\alpha-1}, C_\alpha]$. Thus we always have $p(B_\alpha) \leq p(B_{\alpha+1}) \leq T$ (again, as otherwise $C_k > D_{k-1}$, a contradiction).

Subcase 2.1. $\alpha^* = 1$. In this subcase, all the batches are finished after time t, and thus $C^*_{\max} \geq t + k^*T$. It follows from $p(B_\alpha) \leq T$ and $k \leq 2k^* - 1$ [5] that

$$C_{\max} = t + p(B_\alpha) + (k - \alpha + 1)T \leq t + T + 2k^*T - \alpha T = t + 2k^*T \leq 2C^*_{\max}.$$

Subcase 2.2. $\alpha^* \geq 2$. In this subcase, $C^*_{\max} \geq \max\{t + (k^* - \alpha^* + 1)T, p(B^*_1) + k^*T\}$.

Let $\mathcal{J}'$ denote the subset of jobs that are processed before time s in the optimal schedule π^*, and $\mathcal{J}'' = \mathcal{J} - \mathcal{J}'$. Clearly, $\sum_{j=1}^{\alpha} p(B_j) > p(\mathcal{J}')$ since not all the jobs of B_α can be processed before time s. On the other hand, the batch sequence $\langle B_1, B_2, \ldots, B_k\rangle$ is the rearrangement of the batch sequence $\langle B'_1, B'_2, \ldots, B'_k\rangle$ in the SPT order; therefore, $\sum_{j=1}^{\alpha} p(B_j) \leq \sum_{j=1}^{\alpha} p(B'_j)$. It follows that

$$\sum_{j=1}^{\alpha} p(B'_j) > p(\mathcal{J}').$$

By Lemma 4 we have

$$\sum_{j=1}^{\alpha} s(B'_j) > s(\mathcal{J}'),$$

and thus

$$\sum_{j=\alpha+1}^{k} s(B'_j) < s(\mathcal{J}'').$$

From Lemma 3 and that the jobs of $\mathcal{J}''$ are in $k^* - \alpha^* + 1$ batches, we conclude that

$$k - \alpha \le 2(k^* - \alpha^* + 1) - 1.$$

It follows from $p(B_\alpha) \le T$ that

$$\begin{aligned} C_{\max} &= t + p(B_\alpha) + (k - \alpha + 1)T \\ &\le t + T + 2(k^* - \alpha^* + 1)T \\ &= (t + (k^* - \alpha^* + 1)T) + (k^* - \alpha^* + 2)T \\ &\le 2C^*_{\max}. \end{aligned}$$

That is, in the remaining situation we also have $C_{\max} \le 2C^*_{\max}$. Hence the algorithm D-NF-SPT is a 2-approximation.

The running time of the algorithm D-NF-SPT in $O(n \log n)$, where n is the number of jobs, is clearly seen, because the job sorting by density and the later job batch sorting by processing time take an $O(n \log n)$-time, and the algorithm NF takes only an $O(n)$-time. This proves the theorem. □

2.1 A Tight Instance

In this instance I there are $2n$ jobs, $\mathcal{J} = \{J_1, J_2, \ldots, J_{2n}\}$, with n being even. The processing time and the size of the job J_i is (p_i, s_i), and here $J_{2i-1} = (i\epsilon, \frac{1}{2})$ and $J_{2i} = ((2i+1)\epsilon^2, \epsilon)$ for every $i = 1, 2, \ldots, n$. The positive constant ϵ is small such that $\epsilon < \frac{1}{2n+1}$. The machine maintenance time interval is $[s, t]$ where $s = \frac{1}{2}n(n-1)\epsilon + (n-1)(n+1)\epsilon^2 + \frac{1}{2}(2n-1)\epsilon^2$ and $t = s + \frac{1}{2}(2n-1)\epsilon^2$, *i.e.* $\Delta = \frac{1}{2}(2n-1)\epsilon^2$. The one shipment delivery time is $T = 1$.

Clearly, $\frac{s_i}{p_i} = \frac{1}{(i+1)\epsilon}$ for every $i = 1, 2, \ldots, 2n$ and therefore the job order after Step 1 of the algorithm D-NF-SPT is $\langle J_1, J_2, \ldots, J_{2n}\rangle$. Using this job order, the algorithm NF packs the jobs into a sequence of n batches $B'_j = \{J_{2j-1}, J_{2j}\}$, $j = 1, 2, \ldots, n$. Clearly, $s(B'_j) = \frac{1}{2} + \epsilon$ for all j, and $p(B'_j) = j\epsilon + (2j+1)\epsilon^2$. Therefore, $B_j = B'_j$ for every j, and the final batch order is $\langle B_1, B_2, \ldots, B_n\rangle$. Note that

$$s = \frac{1}{2}n(n-1)\epsilon + (n-1)(n+1)\epsilon^2 + \frac{1}{2}(2n-1)\epsilon^2 = \sum_{j=1}^{n-1} p(B_j) + \frac{1}{2}(2n-1)\epsilon^2.$$

Since $p(B_j) = j\epsilon + (2j+1)\epsilon^2 < T$ for every j, $C_{n-1} < s < t < C_{n-1} + p(B_n)$ and $(2n-1)\epsilon^2 + p(B_n) < T$, for the achieved schedule π its makespan is

$$C_{\max} = p(B_1) + nT = \epsilon + 3\epsilon^2 + nT. \tag{2}$$

Consider a feasible schedule in which there are $\frac{n}{2} + 1$ batches where $B^*_j = \{J_{4j-3}, J_{4j-1}\}$ for each $j = 1, 2, \ldots, \frac{n}{2}$, and $B^*_{\frac{n}{2}+1} = \{J_2, J_4, \ldots, J_{2n}\}$. Clearly, $s(B^*_j) = 1$ for all $1 \le j \le \frac{n}{2}$, $s(B^*_{\frac{n}{2}+1}) = n\epsilon$, $p(B^*_j) = (4j-1)\epsilon$ for all $1 \le j \le \frac{n}{2}$, and $p(B^*_{\frac{n}{2}+1}) = n(n+2)\epsilon^2$. Since $\sum_{j=1}^{\frac{n}{2}-1} p(B^*_j) < s$, all the jobs of the batches

$B_1^*, B_2^*, \ldots, B_{\frac{n}{2}-1}^*$ are processed before time s in this feasible schedule. Due to $\sum_{j=1}^{\frac{n}{2}+1} p(B_j^*) + (2n-1)\epsilon^2 < \frac{n}{2} - 1$, no matter when the jobs of the batches $B_{\frac{n}{2}}^*$ and $B_{\frac{n}{2}+1}^*$ are processed, the vehicle has not delivered the batch $B_{\frac{n}{2}-1}^*$ and comes back to the machine. It follows that the makespan of this feasible schedule is at most $p(B_1^*) + (\frac{n}{2}+1)T = 3\epsilon + (\frac{n}{2}+1)T$. Therefore, the makespan of an optimal schedule for the instance I is also

$$C_{\max}^* \leq 3\epsilon + \left(\frac{n}{2} + 1\right) T. \tag{3}$$

Consequently, putting Eqs. (2) and (3) together gives

$$\frac{C_{\max}}{C_{\max}^*} \geq \frac{\epsilon + 3\epsilon^2 + nT}{3\epsilon + (\frac{n}{2}+1)T} \to 2, \text{ when } n \to +\infty.$$

3 Conclusions

We have investigated the single scheduling problem with job delivery coordination, in which the machine has an unavailable maintenance interval. A good schedule needs not only to well organize jobs into a smaller number of shipments to save delivery time, but also must wisely exploit the machine time period before maintenance. The first consideration is addressed by employing a good approximation algorithm for the bin-packing problem, where the item size is the job physical size and the bin has a size that is the vehicle capacity. Nevertheless, the second consideration implies that the machine should perhaps process first those *jobs* of shorter processing times. We realized that this is *not the same as* the machine processing first those *batches* of shorter processing time. We thus propose to sort the jobs in a non-increasing order of the density s_i/p_i, and call the *next-fit* (NF) bin-packing algorithm to pack the jobs into batches. Two key properties of the packing results achieved by the algorithm NF lead to the desired performance analysis.

We also showed that the worst-case performance ratio 2 of the algorithm D-NF-SPT is tight. It would be really interesting to see whether the problem admits a better approximation algorithm, for example, by distinguishing the machine availability before and after the maintenance. From the practical point of view, it is worth investigating the problem where the machine has multiple maintenance periods, whether they occur on a regular basis, or irregularly but known in advance.

Acknowledgements. Hu is supported by the National Natural Science Foundation of China (NNSF) Grants No. 11271324 and 11471286. Luo is supported by a sabbatical research grant of Lin; his work was mostly done during his visit to the UofA. Dong is supported by the Zhejiang Provincial Natural Science Foundation Grants No. LY13A010015 and the Science Foundation of Zhejiang Sci-Tech University (ZSTU) Grants No. 13062171-Y. Tong is supported by an Alberta Innovates Technology Futures (AITF) Graduate Student Scholarship. Goebel is supported by AITF and the Natural

Sciences and Engineering Research Council of Canada (NSERC). Xu is supported by the NNSF Grants No. 61221063 and 71371129, and the Program for Changjiang Scholars and Innovative Research Team in University Grant No. IRT1173. Lin is supported by NSERC and the Science Foundation of Zhejiang Sci-Tech University (ZSTU) Grants No. 14062170-Y; his work was mostly done during his sabbatical leave at the ZSTU.

References

1. Chang, Y.-C., Lee, C.-Y.: Machine scheduling with job delivery coordination. Eur. J. Oper. Res. **158**, 470–487 (2004)
2. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco (1979)
3. Garey, M.R., Johnson, D.S.: A 71/60 theorem for bin-packing. J. Complex. **1**, 65–106 (1985)
4. He, Y., Zhong, W., Gu, H.: Improved algorithms for two single machine scheduling problems. Theoret. Comput. Sci. **363**, 257–265 (2006)
5. Johnson, D.S.; Near-optimal allocation algorithms. Ph.D. thesis, Massachusetts Institute of Technology (1973)
6. Lee, C.-Y., Chen, Z.-L.: Machine scheduling with transportation considerations. J. Sched. **4**, 3–24 (2001)
7. Lee, C.-Y., Lei, L., Pinedo, M.: Current trends in deterministic scheduling. Ann. Oper. Res. **70**, 1–41 (1997)
8. Lu, L., Yuan, J.: Single machine scheduling with job delivery to minimize makespan. Asia-Pac. J. Oper. Res. **25**, 1–10 (2008)
9. Wang, X., Cheng, T.C.E.: Machine scheduling with an availability constraint and job delivery coordination. Naval Res. Logistics **54**, 11–20 (2007)
10. Zhong, W., Dósa, G., Tan, Z.: On the machine scheduling problem with job delivery coordination. Eur. J. Oper. Res. **182**, 1057–1072 (2007)