Single bounded parallel-batch machine scheduling with an unavailability constraint and job delivery ⋆

Jing Fan[1], C. T. Ng[2], T. C. E. Cheng[2], and Hui Shi[3]

[1] Shanghai Polytechnic University, Shanghai 201209, China
[2] Logistics Research Centre, Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong SAR, China
[3] Shanghai General Hospital, School of Medicine, Shanghai Jiaotong University, Shanghai 200080, China sspu_fj@163.com

Abstract. We consider a scheduling problem where a manufacturer processes a set of jobs for a customer and delivers the completed jobs to the customer. The job sizes and processing times are given. The objective is to minimize the maximum delivery time to the customer. In the production stage, one machine with an unavailability period is used to process the jobs. The machine has a fixed capacity and the jobs are processed in batches under the condition that the total size of the jobs in a batch cannot exceed the machine capacity. The processing time of a batch is the maximum processing time of the jobs contained in the batch. In addition, each batch is non-resumable, i.e., if the processing of a batch cannot be completed before the unavailability period, the batch needs to be processed anew after the unavailability interval. In the distribution stage, the manufacturer assigns a vehicle with a fixed capacity to deliver the completed jobs. The total size of the completed jobs in one delivery cannot exceed the vehicle capacity. We first consider the case where the jobs have the same size and arbitrary processing times, for which we provide a 3/2-approximation algorithm and show that the worst-case ratio is tight. We then consider the case where the jobs have the same processing time and arbitrary sizes, for which we provide a 5/3-approximation algorithm and show that the worst-case ratio is tight.

Keywords: parallel-batch; production and delivery; unavailability constraint; approximation algorithm

⋆ This research was supported in part by the National Natural Science Foundation of China (No.11601316). The first author was also supported in part by the key discipline "Applied Mathematics" of Shanghai Polytechnic University (No.XXKPY1604), Research Center of Resource Recycling Science and Engineering, and Gaoyuan Discipline of Shanghai − Environmental Science and Engineering (Resource Recycling Science and Engineering) of Shanghai Polytechnic University.

1 Introduction

Although production scheduling integrated with logistics is more difficult to deal with than production scheduling alone from the theoretical research perspective, there is an abundance of literature on the former because such problems are more practical in the realistic manufacturing environment.

Potts [1] was probably the first researcher that considered scheduling with job delivery. Hall and Potts [2] studied integrated scheduling that involves a supplier, a manufacturer, and a customer. For integrated scheduling with one machine and one customer, Chen and Vairaktarakis [3] presented a polynomial-time dynamic programming solution algorithm to minimize the maximum arrival time, under the condition that the last completed job delivered to the customer and the total distribution cost do not exceed given bounds. Different from [3] in which the number of vehicles has no limit, Lee and Chen [4] considered the problem with a limited number of vehicles and showed that it is polynomially solvable by dynamic programming. Studying the same problem in [4] where the jobs have different sizes, Chang and Lee [5] showed that the problem is strongly NP-hard and provided a heuristic with a worst-case performance bound of 5/3. He et al. [6] and Zhong et al. [7] studied the same problem in [5], and proposed improved approximation algorithms with worst-case performance bounds of 53/35 and 3/2, respectively.

However, due to the occurrence of breakdowns or the necessity for maintenance and repair, the machines may become unavailable during the production stage. There are many studies on integrated scheduling under the constraint of machine unavailability. When job processing is interrupted by machine unavailability, the interrupted job is often assumed to be non-resumable, which means that the job needs to processed anew as defined in [8]. For the problem with delivery using a capacitated vehicle to minimize the maximum arrival time, Wang and Cheng [9] showed that it is NP-hard, and proposed a 3/2-approximation algorithm and showed that the worst-case ratio is tight. More details on this research stream can be found in Chen [10] and Ma et al. [11].

In most models of scheduling with delivery, the machine processes one job at a time. However, it is noted that batch-processing machines with limited capacities are also widely used in real production. When a bounded parallel-batch machine is used for production, the jobs are processed in batches under the condition that the total size of the jobs in a batch cannot exceed the machine capacity, and the processing time of a batch is equal to the longest processing time of the jobs in it. Li et al. [12] and Gong et al. [13] considered several parallel-batch machine scheduling problems with job delivery. Lu and Yuan [14] considered unbounded parallel-batch scheduling with job delivery to minimize the makespan. They provided a polynomial-time algorithm to solve the case where the jobs have identical sizes, and a heuristic with a worst-case performance ratio of 7/4 for the case where the jobs have non-identical sizes. Cheng et al. [15] considered integrated scheduling of production and distribution on parallel batch-processing machines. They presented a $(2-1/m)$-approximation algorithm for the case where the jobs have the same size and arbitrary processing times, and provided a 13/7-approximation algorithm for the case where the jobs have the same processing time and arbitrary sizes.

In this paper we consider scheduling with job delivery for a customer on a bounded parallel-batch machine with a machine unavailability period, where a capacitated vehicle is used to deliver the completed jobs to the customer. The objective is to minimize the time of the last completed job delivered to the customer. We consider two cases of the problem corresponding to different conditions on the job processing times and sizes, and design approximation algorithms for them.

2 Description of the problem

In our problem, there is a manufacturer that processes jobs on a bounded parallel-batch machine and delivers the completed jobs to a customer. Given a set of n jobs $J = \{J_1, J_2, \cdots, J_n\}$, where job J_j has the processing time p_j and size s_j for $j = 1, 2, ..., n$. In the production stage, the machine has a capacity B, i.e., it can simultaneously process at most B jobs as a batch. The processing time of a batch is the maximum processing time of the jobs contained in the batch. Due to reasons such as maintenance, breakdown etc, the machine has an unavailability period $[t_1, t_2]$. Let I be the length of the unavailability period, i.e., $I = t_2 - t_1$. The processing of batches is non-resumable, i.e., if there is at least one job in a batch that is interrupted by the unavailable period $[t_1, t_2]$, the whole batch needs to be processed anew after t_2. In the delivery stage, there is a vehicle with a capacity c to deliver the completed jobs to the customer. The transport time between the machine and the customer is T. Let D_j be the delivery time of job J_j, i.e., the arrival time of the batch containing job J_j to the customer. The objective is to minimize the maximum delivery time of all the jobs, denoted by D_{max}.

Chen [10] proposed a five-field notation to denote an integrated scheduling problem as $\alpha|\beta|\pi|\delta|\gamma$, where α represents the facility configuration of the manufacturer; β represents the production constraints; π represents the vehicle configuration and is often denoted by (v_1, v_2), where v_1 represents the number of vehicles and v_2 represents the vehicle capacity; δ represents the number of customers; and γ represents the scheduling objective to be minimized. Using the above notation, we denote the two cases of the problem under study as follows:

$$(\mathcal{P}1) : 1, h_1|nr - a, p - batch, s_j = 1, p_j|V(1, c)|1|D_{max},$$
$$(\mathcal{P}2) : 1, h_1|nr - a, p - batch, s_j, p_j = 1|V(1, c)|1|D_{max}.$$

For problem $(\mathcal{P}1)$, each job has a unit size but an arbitrary processing time. On the contrary, for problem $(\mathcal{P}2)$, each job has an arbitrary size but a unit processing time.

We organize the rest of the paper as follows: In Section 3 we show that problem $(\mathcal{P}1)$ is NP-hard and propose an approximation algorithm for it. In Section 4 we prove that problem $(\mathcal{P}2)$ is strongly NP-hard and present an approximation algorithm for it.

3 Algorithm for $(\mathcal{P}1) : 1, h_1|nr-a, p-batch, s_j = 1, p_j|V(1,c)|1|D_{max}$

In this section we study problem $(\mathcal{P}1)$, where the jobs have the same size, i.e., $s_j = 1$. Similar to [15], we assume that the manufacturer uses appropriate equipment to improve the efficiency of the supply chain. Specifically, we let $c = \mu B$, where $\mu \geq 2$ and μ is a positive integer.

We first analyze the computational complexity of $(\mathcal{P}1)$.

Theorem 1. *$(\mathcal{P}1) : 1, h_1|nr-a, p-batch, s_j = 1, p_j|V(1,c)|1|D_{max}$ is NP-hard.*

Proof. Consider the special case $(\mathcal{P}1')$, where $B = 1$, i.e., each batch contains at most one job. It is clear that $(\mathcal{P}1')$ is equivalent to $1, h_1|nr-a|V(1,c)|1|D'_{max}$, where D'_{max} is the maximum of the return time D'_j of the vehicle after delivering the completed jobs to the customer. Obviously, $D'_j = D_j + T$ for job J_j. Given that Wang and Cheng [9] have shown that problem $(\mathcal{P}1')$ is NP-hard, we obtain the conclusion.

Next, we derive some properties of the optimal solution for problem $(\mathcal{P}1)$.

Lemma 1. *There exists an optimal schedule σ^* possessing the following properties:*

(1) Let N^* to be the number of batches, then $N^* = \lceil n/B \rceil$;

(2) The batches are processed consecutively before and after the unavailability period;

(3) The batch that becomes available earlier is delivered earlier.

(4) The first delivery includes b^* batches, and each of the last a^* deliveries includes μ jobs, where a^* and b^* are two positive integers satisfying $N^* = a^*\mu + b^*$ and $0 < b^* \leq \mu$, respectively.

The lemma can be proved similar to the proof in [9]. Because every job has the same size 1, we construct the following algorithm including the same number of batches as the optimal schedule.

Algorithm A1

Step 1: Sequence all the jobs in non-increasing order of their processing times.

Step 2: Create the first batch H_{N^*} and put the first B jobs in H_{N^*}. Then create batch H_{N^*-1} and put the next B jobs in it. Repeat the assignment until there are y jobs left, where $0 < y \leq B$. Put them in batch H_1. The obtained batch set is $\{H_1, H_2, ..., H_{N^*}\}$. The batches are in non-decreasing order of their batch processing times.

Step 3: Regard batch H_j as job $\widetilde{J}_j$ for $j = 1, 2, ..., N^*$, whose processing time is the maximum processing time of the jobs in H_j and the size is 1. Take the obtained job set $\{\widetilde{J}_1, \widetilde{J}_2, ..., \widetilde{J}_N\}$ as the job set of problem $(\mathcal{P}1') : 1, h_1|nr-a|V(1,c)|1|D'_{max}$. Use the approximation algorithm proposed in Wang and Cheng [9] to obtain the schedule σ.

Furthermore, we can obtain the following lemma.

Lemma 2. *Sorting all the batches of* σ *and* σ^* *in non-decreasing order, we have* $P(H_j) \leq P(H_j^*)$ *for* $j = 1, ..., N^*$*, where* $P(H_j)$ *and* $P(H_j^*)$ *are the processing times of the* j*-th batch in schedules* σ *and* σ^**, respectively,*

The proof is similar to Cheng et al. [15]. Given that the worst-case ratio of the approximation algorithm Wang and Cheng [9] proposed for $(\mathcal{P}1')$ is $3/2$, we derive the worst-case ratio of algorithm $A1$ for $(\mathcal{P}1)$ as follows:

Theorem 2. *Solving* $(\mathcal{P}1)$*, algorithm* $A1$ *has the worst-case ratio of* $3/2$*, which is tight.*

Proof. It suffices to prove that the objective value produced by the optimal schedule for $(\mathcal{P}1')$, denoted by $\sigma^*(\mathcal{P}1')$, is not greater than that for $(\mathcal{P}1)$, denoted by $\sigma^*(\mathcal{P}1)$. Otherwise, we construct a new schedule $\widehat{\sigma}(\mathcal{P}1)$ by replacing the corresponding batches of $\sigma^*(\mathcal{P}1)$ with the batches of $\sigma^*(\mathcal{P}1')$ to process in non-decreasing order of the batches in the two schedules, and delivering the batches as $\sigma^*(\mathcal{P}1)$. Because the completion times of the batches in $\widehat{\sigma}(\mathcal{P}1)$ are no later than those in $\sigma^*(\mathcal{P}1)$ by Lemma 2, so are the delivery times. Hence, the new schedule $\widehat{\sigma}(\mathcal{P}1)$ is no worse than $\sigma^*(\mathcal{P}1)$.

Next, consider the following instance: $n = 6$, $B = 2$, $\mu = 2$, $[t_1, t_2] = [2, 2+\epsilon]$, $p_1 = p_2 = \epsilon$, $p_3 = p_4 = 1$, and $p_5 = p_6 = 1$. The delivery time is $T = \epsilon$. The schedule produced by algorithm $A1$ is as follows: The first delivery including J_1 and J_2 is delivered at ϵ, and the second delivery including J_3, J_4, J_5, and J_6 is delivered at $3 + \epsilon$. Hence, $D_{max} = 3 + 2\epsilon$. However, in the optimal schedule, there are two deliveries: the first delivery consisting of J_3 and J_4, and the second delivery consisting of the others. The optimal objective function value is $D^*_{max} = 2 + 3\epsilon$. So, we have $\frac{D_{max}}{D^*_{max}} = \frac{3}{2}$ if ϵ is sufficiently small.

4 Algorithm for $(\mathcal{P}2) : 1, h_1|nr - a, p - batch, s_j, p_j = 1|V(1, c)|1|D_{max}$

In $(\mathcal{P}2) : 1, h_1|nr - a, p - batch, s_j, p_j = 1|V(1, c)|1|D_{max}$, all the jobs have a unit processing time but arbitrary sizes. First we analyze the computational complexity of $(\mathcal{P}2)$.

Theorem 3. $(\mathcal{P}2) : 1, h_1|nr - a, p - batch, s_j, p_j = 1|V(1, c)|1|D_{max}$ *is strongly* NP*-hard.*

Proof. Consider the special case $(\mathcal{P}2')$, where $t_1 = t_2$ and $T = 0$, i.e., there is no unavailability interval on the machine and no delivery is needed. Since each job has a processing time of 1, every batch has a processing time of 1. Hence, $(\mathcal{P}2')$ is equivalent to minimizing the number of batches, i.e., the bin-packing problem, which is a well-known strongly NP-hard problem. Therefore, $(\mathcal{P}2)$, as well as $(\mathcal{P}2')$, is strongly NP-hard.

Obviously, the optimal schedule for $(\mathcal{P}2)$ possesses properties $(2) - (4)$ in Lemma 1.

In this section we use the same notation and the corresponding meanings, such as N and N^*, as those used in Section 3. Next, we propose the following approximation algorithm for $(\mathcal{P}2)$.

Algorithm A2

Step 1: Sort the jobs in non-increasing order of their sizes. Re-label them as job $J_1, J_2, \cdots, J_n$.

Step 2: Use the First Fit Decreasing (FFD) rule to assign the jobs into batches. Create the first empty batch H_1 and put job J_1 in it. Check the following jobs one by one as to whether it can be put in the batch. If so, put the job in H_1 and delete it from the job list. If not, go on to check the next job. When all the jobs have been assigned, create the second batch H_2 and assign the remaining jobs in the job list. Repeat the assignment until there is no job in the job list. The obtained batch set is $\{H_1, ..., H_N\}$.

Step 3: Assign the batches in an arbitrary order to the machine for processing.

Step 4: Deliver the first completed b batches in D_1. For the following batches, deliver μB batches immediately in each delivery. If the vehicle is available when the μB batches are completed, deliver them immediately. If the vehicle is not available at the time, wait until the vehicle returns to the manufacturer and deliver the batches. When the last μB batches are delivered to the customer, production and distribution are finished, and the obtained deliveries are $D_1, ..., D_{a+1}$.

To analyze the performance of algorithm A2, recall that for the bin-packing problem, the number of bins obtained by FFD is no more than the sum of 6/9 and 11/9 times the optimal number of bins. In algorithm $A2$, Steps 1 and 2 assign the jobs to batches by the FFD rule, so we have the following results on N and a.

Lemma 3. *([16]) $N \le \frac{11}{9}N^* + \frac{6}{9}$, where N^* is the optimal number of batches.*

Lemma 4. $a^* \le a < \frac{11}{9}a^* + \frac{14}{9}$.

Proof. It is clear that $a^* \le a$.

By Lemma 3, we have $N \le \frac{11}{9}N^* + \frac{6}{9} \le \frac{11}{9}a^*\mu + \frac{11b^*+6}{9}$ and $\frac{N}{\mu} \le \frac{11}{9}a^* + \frac{11b^*+6}{9\mu}$. Since $0 < b^* \le \mu$ and $\mu \ge 2$,

$$a \le \frac{N}{\mu} \le \frac{11}{9}a^* + \frac{11b^*+6}{9\mu} \le \frac{11}{9}a^* + \frac{11}{9} + \frac{6}{9\mu} \le \frac{11}{9}a^* + \frac{14}{9}.$$

But if $a = \frac{11}{9}a^* + \frac{14}{9}$, by $N \le \frac{11}{9}N^* + \frac{6}{9}$, $N = a\mu + b$, and $N^* = a^*\mu + b^*$, we have $\frac{11}{9}a^*\mu + \frac{14}{9}\mu + b \le \frac{11}{9}(a^*\mu + b^*) + \frac{6}{9}$, i.e.,

$$\frac{14}{9}\mu < \frac{11}{9}b^* + \frac{6}{9} \le \frac{11}{9}\mu + \frac{6}{9}.$$

So, we deduce that $\mu < 2$, which contradicts the assumption $\mu \ge 2$.

As a result, we can easily obtain the maximal value of a when $a^* \leq 4$ in Table 1.

Table 1. The maximal values of a^* when $a^* \leq 4$.

a^*	1	2	3	4
Maximal value of a	2	3	5	6

For convenience, we use L and L^* to denote the numbers of deliveries in schedules π and π^*, respectively, which means $L = a + 1$ and $L^* = a^* + 1$. Meanwhile, we use $C(D_j)$ and $C(D_j^*)$ to denote the completion times of deliveries D_j and D_j^* in π and π^*, and D_{max} and D_{max}^* to denote the objective values of π and π^*, respectively. Because every job has a unit processing time, we can easily obtain the following relationships between π and π^*.

Lemma 5. *(1) $\lambda = \lambda^*$, where λ and λ^* are the numbers of batches completed before the unavailability interval $[t_1, t_2]$ in π and π^*, respectively;*

(2) $\delta \leq \delta^*$, where δ and δ^* are the idle times on the machine before $[t_1, t_2]$ in π and π^*, respectively.

The results are obvious and we omit the proof.

Lemma 6. *$k - 1 \leq l \leq k + 1$, where l and k are the first deliveries completed after the unavailability interval $[t_1, t_2]$ in π and π^*, respectively.*

Proof. Because k is the first delivery completed after t_2, there are at most $k\mu$ batches completed in the total k deliveries in π^*. We prove the result by contradiction.

If $l \geq k + 2$, there are at least $(k + 1)$ deliveries before t_1 in π, i.e., there are at least $k\mu$ batches completed before t_1. So it is a contradiction.

If $l \leq k - 2$, there are $(k - 1)$ deliveries before t_1 in π^*, i.e., there are no fewer than $(k - 2)\mu$ batches completed before t_1, which is a contradiction to the situation that at most $(k - 2)\mu$ batches completed after t_2 in π.

In the following we analyze the worst-case ratios of algorithm $A2$ for $(\mathcal{P}2)$ according to $a^* \geq 5$ and $a^* \leq 4$, respectively.

Lemma 7. *When $a^* \geq 5$, the worst-case ratio of algorithm $A2$ is $\frac{5}{3}$.*

Proof. We prove the result in three cases.

Case 1: $C(D_1^*) \geq t_2$, which means $k = 1$ and $l = 1$ or 2.

• If $D_{max}^* = C(D_{L^*}^*) + T = N^* + I + \delta^* + T$, then we deduce that $\mu \geq 2T$. Moreover, $D_{max} = C(D_L) + T = N + I + \delta + T$, so

$$\frac{D_{max} - D_{max}^*}{D_{max}^*} \leq \frac{N - N^*}{N^*} = \frac{N}{N^*} - 1.$$

Since $a^* \geq 5$ and $\mu \geq 2$, $N^* \geq 10$. Similar to [15], we can find positive integers $\alpha \geq 1$ and $1 \leq \beta \leq 9$ such that $N^* = 9\alpha + \beta$, and we can obtain an upper bound on $\frac{N}{N^*}$ from Table 2 as follows:

$$\frac{N}{N^*} \leq \frac{9\alpha+2}{11\alpha+3} \leq \frac{9}{11} + \frac{5}{9(9\alpha+2)} \leq \frac{14}{11} < \frac{5}{3}. \tag{1}$$

Table 2. Upper bounds on $\frac{N}{N^*}$.

N^*	9α+1	9α+2	9α+3	9α+4	9α+5	9α+6	9α+7	9α+8	9α+9
Maximal value of N	11α+1	11α+3	11α+4	11α+5	11α+6	11α+8	11α+9	11α+10	11α+11
Upper bound on $\frac{N}{N^*}$	$\frac{11}{9}$	$\frac{14}{11}$	$\frac{5}{4}$	$\frac{16}{13}$	$\frac{11}{9}$	$\frac{19}{15}$	$\frac{5}{4}$	$\frac{21}{17}$	$\frac{11}{9}$

• If $D^*_{max} = C(D^*_1) + (2a^*+1)T = b^* + I + \delta^* + (2a^*+1)T$, then $\mu < 2T$ and the objective value of schedule π is

$$D_{max} = \begin{cases} C(D_1) + (2a+1)T, & \text{for } \ l = 1 \text{ or } 2, \\ C(D_l) + (2(a-l)+3)T, & \text{for } \ l = 2. \end{cases}$$

For the first case, $D_{max} \leq b + I + \delta + (2a+1)T$. By Lemma 5 and $b - b^* \leq \mu$, we have

$$\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{(b-b^*) + 2(a-a^*)T}{C(D^*_1) + (2a^*+1)T} < \frac{a - a^* + 1}{a^* + \frac{1}{2}}.$$

Moreover, because of $a^* \geq 5$ and Lemma 4, we have

$$\frac{a - a^* + 1}{a^* + \frac{1}{2}} \leq \frac{2}{9} + \frac{22}{9(a^* + \frac{1}{2})} \leq \frac{2}{3}. \tag{2}$$

For the second case, we have $b < b^*$; otherwise, $C(D_1) > t_2$, which contradicts $l = 2$. Hence,

$$\begin{aligned} \frac{D_{max}-D^*_{max}}{D^*_{max}} &\leq \frac{(b-b^*)+(l-1)\mu+2(a-a^*-l+1)T}{C(D^*_1)+(2a^*+1)T} \\ &= \frac{(b-b^*)+2(a-a^*)T+\mu-2T}{C(D^*_1)+(2a^*+1)T} \\ &\leq \frac{a-a^*}{a^*+\frac{1}{2}} < \frac{2}{3}. \end{aligned}$$

Case 2: $C(D^*_k) > t_2$ for $1 < k < L^*$, where D^*_k is the first delivery completed after the unavailability interval $[t_1, t_2]$ in π^*.

• If $D^*_{max} = C(D^*_{L^*}) + T = N^* + I + \delta^* + T$ and $D_{max} = C(D_L) + T = N + I + \delta + T$, we obtain the same result as (1).

• If $D^*_{max} = C(D^*_1) + (2a^*+1)T = b^* + I + \delta^* + (2a^*+1)T$, then $\mu < 2T$ and $2(k-1)T > (k-1)\mu + I + \delta^*$. Therefore, the objective vale of schedule π is

$$D_{max} = \begin{cases} C(D_1) + (2a+1)T, & \text{for } \ l = k-1, \ k \text{ or } k+1, \\ C(D_l) + (2(a-l)+3)T, & \text{for } \ l = k-1. \end{cases}$$

For the first case, we obtain the same result as (2). For the second case, we have

$$\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{(b - b^*) + (l - 1)\mu + I + \delta + 2(a - a^* - l + 1)T}{b^* + (2a^* + 1)T}.$$

Because of $b - b^* \leq \mu$ and $l = k - 1$, $(b - b^*) + (l - 1)\mu + I + \delta + 2(a - a^* - l + 1)T \leq 2(a - a^* + 1)T$. Moreover, $\frac{D_{max}-D^*_{max}}{D^*_{max}} \leq \frac{2}{3}$.

• If $D^*_{max} = C(D^*_k) + (2(a^* - k) + 3)T = b^* + (k - 1)\mu + I + \delta^* + (2(a^* - k) + 3)T$, then $\mu < 2T$ and $2(k - 1)T \leq (k - 1)\mu + I + \delta^*$. Hence, the objective value of schedule π is

$$D_{max} = \begin{cases} C(D_1) + (2a + 1)T, & \text{for } \ l = k + 1, \\ C(D_l) + (2(a - l) + 3)T, & \text{for } \ l = k - 1, \ k \text{ or } k + 1. \end{cases}$$

For the first case, we have

$$\begin{aligned} \tfrac{D_{max}-D^*_{max}}{D^*_{max}} &\leq \tfrac{(b-b^*)+2(a-a^*)T-((k-1)\mu+I+\delta^*+(2-2k)T)}{b^*+(k-1)\mu+I+\delta^*+(2(a^*-k)+3)T} \\ &\leq \tfrac{(b-b^*)+2(a-a^*)T}{b^*+2(k-1)T+(2(a^*-k)+3)T} \\ &\leq \tfrac{2(a-a^*+1)T}{2(k-1)T+(2(a^*-k)+3)T} \\ &= \tfrac{a-a^*+1}{a^*+\frac{1}{2}} \leq \tfrac{2}{3}. \end{aligned}$$

For the second case, we have

$$\begin{aligned} \tfrac{D_{max}-D^*_{max}}{D^*_{max}} &\leq \tfrac{(b-b^*)+2(a-a^*)T+(l-k)(\mu-T))}{b^*+(k-1)\mu+I+\delta^*+(2(a^*-k)+3)T} \\ &\leq \tfrac{(b-b^*)+2(a-a^*)T+(l-k)(\mu-T))}{(2a^*+1)T}. \end{aligned}$$

When $l = k-1$ or k, we easily obtain $\frac{D_{max}-D^*_{max}}{D^*_{max}} \leq \frac{a-a^*+1}{a^*+\frac{1}{2}} \leq \frac{2}{3}$. When $l = k+1$, we have $b \leq b^*$; otherwise, $C(D_{l-1}) > t_2$, which contradicts the definition of l. Hence, we obtain the same result.

Case 3: $C(D^*_{L^*}) \geq t_2$, i.e., the last delivery $D^*_{L^*}$ is completed after the unavailability interval $[t_1, t_2]$ in π^*. Obviously, $L^* = a^* + 1$.

• If $D^*_{max} = C(D^*_{L^*}) + T = b^* + (L^* - 1)\mu + I + \delta^* + T = b^* + a^*\mu + I + \delta^* + T$, then we deduce that $2a^*T \leq a^*\mu + I + \delta^*$. Moreover,

$$D_{max} = \begin{cases} C(D_L) + T, & \text{for } \ l = L^* - 1, \ L^* \text{ or } L^* + 1, \\ C(D_l) + (2(a - l) + 3)T, & \text{for } \ l = L^* - 1, \ L^*, \\ C(D_1) + (2a + 1)T, & \text{for } \ l = L^* + 1. \end{cases}$$

For the first case, we obtain the same result as (1). For the second case, we have $\mu < 2T$, $D_{max} = b + (l - 1)\mu + I + \delta + (2(a - l) + 3)T$, and

$$\begin{aligned} \tfrac{D_{max}-D^*_{max}}{D^*_{max}} &\leq \tfrac{(b-b^*)+(l-L^*)\mu+2(a-l+1)T}{b^*+(L^*-1)\mu+I+\delta^*+T} \\ &\leq \tfrac{(b-b^*)+(l-L^*)\mu+2(a-l+)T}{2(L^*-\frac{1}{2})T}. \end{aligned}$$

When $l = L^* - 1$, $(b - b^*) + (l - L^*)\mu + 2(a - l + 1)T \leq \mu - \mu + 2(a - a^* + 1)T = 2(a - a^* + 1)T$. When $l = L^*$, $(b - b^*) + (l - L^*)\mu + 2(a - l + 1)T \leq \mu + 0 + 2(a - a^*)T \leq 2(a - a^* + \frac{1}{2})T < 2(a - a^* + 1)T$. Therefore,

$$\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{a - a^* + 1}{a^* + \frac{1}{2}} \leq \frac{2}{3}.$$

For the third case, since $\mu < 2T$ and $D_{max} = C(D_1) + (2a + 1)T = b + (2a + 1)T$, we have

$$\begin{aligned} \frac{D_{max} - D^*_{max}}{D^*_{max}} &= \frac{(b - b^*) - (a^*\mu + I + \delta^*) + 2aT}{b^* + a^*\mu + I + \delta^* + T} \\ &\leq \frac{(b - b^*) + 2(a - a^*)T}{2(a^* + \frac{1}{2})T} \\ &\leq \frac{a - a^* + 1}{a^* + \frac{1}{2}} \leq \frac{2}{3}. \end{aligned}$$

• If $D^*_{max} = C(D^*_1) + (2a^* + 1)T = b^* + (2a^* + 1)T$, then $\mu < 2T$ and $2(L^* - 1)T > (L^* - 1)\mu + I + \delta^*$, i.e., $2a^*T > a^*\mu + I + \delta^*$. We obtain

$$D_{max} = \begin{cases} C(D_1) + (2a + 1)T, & \text{for } \ l = L^* - 1,\ L^* \text{ or } L^* + 1, \\ C(D_l) + (2(a - l) + 3)T, & \text{for } \ l = L^* - 1. \end{cases}$$

For the first case, we achieve the same result as (2). For the second case, since $l = L^* - 1$, we have $2(L^* - 2)T \leq (L^* - 2)\mu + I + \delta^*$ and $D_{max} = b + (l - 1)\mu + I + \delta + (2(a - l) + 3)T \leq b + 2(L^* - 2)T + 2(a - L^* + 1)T + 3T = b + (2a + 1)T = C(D_1) + (2a + 1)T$. Hence, we obtain the inequalities in (2).

Lemma 8. *When $a^* \leq 4$, the worst-case ratio of algorithm A2 is $\frac{5}{3}$.*

Proof. We prove the result in two cases.

Case 1: $b \leq b^*$. In fact, this inequality holds for $a^* \leq 4$ and the values of a satisfy Lemma 4 except $a^* = 3$ and $a = 4$. Now we show it by contradiction. If $b > b^*$, then $a\mu + b^* < a\mu + b \leq \frac{11}{9}(a^*\mu + b^*) + \frac{6}{9} = \frac{11}{9}a^*\mu + \frac{11}{9}b^* + \frac{6}{9}$, i.e., $(a - \frac{11}{9}a^*)\mu < \frac{2}{9}b^* + \frac{6}{9} \leq \frac{2}{9}\mu + \frac{6}{9}$. So

$$(a - \frac{11}{9}a^* - \frac{2}{9})\mu < \frac{6}{9}.$$

Using the corresponding data in Table 1, we obtain $\mu < 2$, contradicting $\mu \geq 2$.

Most parts of the remaining proof are similar to Lemma 7. Here we discuss two different situations.

The first situation is that $D^*_{max} = C(D^*_{L^*}) + T = N^* + I + \delta^* + T = a^*\mu + b^* + I + \delta^* + T$ and $D_{max} = C(D_L) + T = N + I + \delta + T = a\mu + bI + \delta + T$. Hence, $\frac{D_{max} - D^*_{max}}{D^*_{max}} = \frac{(b - b^*) + (a - a^*)\mu}{b^* + a^*\mu + I + \delta^* + T} \leq \frac{a - a^*}{a^*}$. For $a^* = 2, 3, 4$ and the corresponding maximal value of a in Table 1, we obtain $\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{2}{3}$. For $a^* = 1$ and $a = 2$, we have $\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{N - N^*}{N^*} = \frac{N}{N^*} - 1$. Given the upper bounds on $\frac{N}{N^*}$ for $N^* \leq 9$ in Table 3, we have $\frac{N}{N^*} \leq \frac{3}{2} < \frac{5}{3}$.

Table 3. The upper bounds on $\frac{N}{N^*}$.

N^*	1	2	3	4	5	6	7	8	9
Maximal value of N	1	3	4	5	6	8	9	10	11
Upper bound on $\frac{N}{N^*}$	1	$\frac{3}{2}$	$\frac{4}{3}$	$\frac{5}{4}$	$\frac{6}{5}$	$\frac{4}{3}$	$\frac{9}{7}$	$\frac{5}{4}$	$\frac{11}{9}$

The second situation corresponds to other combinations of D^*_{max} and D_{max}. We always obtain

$$\frac{D_{max} - D^*_{max}}{D^*_{max}} \leq \frac{(b - b^*) + 2(a - a^*)T}{2(a^* + \frac{1}{2})T} \leq \frac{a - a^*}{a^* + \frac{1}{2}} = \frac{a + \frac{1}{2}}{a^* + \frac{1}{2}} - 1.$$

The upper bounds on $\frac{a+\frac{1}{2}}{a^*+\frac{1}{2}}$ is $\frac{5}{3}$, which we deduce from Table 4.

Table 4. The upper bound on $\frac{a+\frac{1}{2}}{a^*+\frac{1}{2}}$ when $a^* \leq 4$.

a^*	1	2	3	4
Upper bound on $\frac{a+\frac{1}{2}}{a^*+\frac{1}{2}}$	$\frac{5}{3}$	$\frac{7}{5}$	$\frac{11}{7}$	$\frac{13}{9}$

Case 2: $b > b^*$ for $a^* = 3$ and $a = 4$. Note that $k - 1 \leq l \leq k$, i.e., $l \neq k + 1$. Most parts of the remaining proof are similar to Lemma 7. Here we discuss two different situations.

The first situation is that $D^*_{max} = C(D^*_{L^*}) + T = a^*\mu + b^* + I + \delta^* + T$ and $D_{max} = C(D_L) + T = a\mu + b + I + \delta + T$. Since $b - b^* \leq \mu$, $\frac{D_{max}-D^*_{max}}{D^*_{max}} = \frac{(b-b^*)+(a-a^*)\mu}{b^*+a^*\mu+I+\delta^*+T} \leq \frac{a-a^*+1}{a^*} = \frac{2}{3}$.

The second situation is that $D^*_{max} = C(D^*_1) + (2a^* + 1)T = b^* + (2a^* + 1)T$ and $D_{max} = C(D_1) + (2a + 1)T = b + I + \delta + (2a + 1)T$ for $k = 2$ and $l = 1$. Given that $\mu+I+\delta \leq 2T$ in this situation, $\frac{D_{max}-D^*_{max}}{D^*_{max}} = \frac{(b-b^*)+I+\delta+2(a-a^*)T}{b^*+(2a^*+1)T} \leq \frac{\mu+I+\delta+2(a-a^*)T}{b^*+(2a^*+1)T} < \frac{4}{7} < \frac{2}{3}$.

From Lemma 7 and Lemma 8, we derive the performance of $A2$ as follows:

Theorem 4. *Solving* $(\mathcal{P}2) : 1, h_1|nr - a, p - batch, s_j, p_j = 1|V(1, v)|1|D_{max}$, *algorithm $A2$ has the worst case ratio of $\frac{5}{3}$, which is tight.*

Proof. It is obvious that the worst-case ratio of algorithm $A1$ is $\frac{5}{3}$ by Lemma 7 and Lemma 8.

Next, consider the following instance: $n = 12$, $B = 7$, $\mu = 2$, $[t_1, t_2] = [1, 1+\epsilon]$, $s_1 = s_2 = s_3 = s_4 = 3$, and $s_5 = s_6 = ... = s_{12} = 2$. The delivery time is $T > \frac{2+\epsilon}{2}$. The schedule produced by algorithm $A2$ is as follows: the first delivery including J_1 and J_2 is delivered at time 1; the second delivery including J_3, ..., J_7 and the third delivery including J_8, ..., J_{12} are delivered at $1+2T$ and $1+4T$, respectively. Hence, $D_{max} = 1 + 5T$. However, in the optimal schedule, there are two deliveries: the first delivery consisting of J_1, J_2, J_5, J_6, J_7, and J_8, and the second delivery consisting of the remaining jobs. The optimal objective value is $D^*_{max} = 2 + 3T$. When $T \to +\infty$, $\frac{D_{max}}{D^*_{max}} = \frac{1+5T}{2+3T} \to \frac{5}{3}$.

References

1. Potts, C.N.: Analysis of a heuristic for one machine sequencing with release dates and delivery times. Operations Research 28, 1436-1441 (1980).
2. Hall, N.G., Potts, C.N.: Supply chain scheduling: batching and delivery. Operatioans Reseatch 51(4), 566-584 (2003).
3. Chen, Z.L., Vairaktarakis, G.L.: Integrated scheduling of production and distribution operations. Management Science 51(4), 614-628 (2005).
4. Lee, C.Y., Chen Z.L.: Machine scheduling with transportation considerations. Journal of Scheduling 4, 3-24 (2001).
5. Chang. Y.C., Lee C.Y.: Machine scheduling with job delivery coordination. European Journal of Operational Research 158, 470-487 (2004).
6. He, Y., Zhong W.Y., Gu H.K.: Improved algorithms for two single machine scheduling problems. Theoretical Computer Science 363, 257-265 (2006).
7. Zhong, W., Dosa G., Tan Z.Y.: On the machine scheduling problem with job delivery coordination. European Journal of Operational Research 182(3), 1057-1072 (2007).
8. Lee, C. Y.: Machine scheduling with an availability constraints. Journal of Global Optimization 9, 363-382 (1996).
9. Wang, X., Cheng T.C.E.: Machine scheduling with an availability constraint and job delivery coordination. Naval Research Logistics 54, 11-20 (2007).
10. Chen Z.L.: Integrated production and outbound distribution scheduling: Review and extensions. Operations Research 58, 130-148 (2010).
11. Ma Y., Chu C.B., Zuo C.R.: A survey of scheduling with deterministic machine availability constraints. Computers & Industrial Engineering 58, 199-211 (2010).
12. Li S.S., Yuan J.J., Fan B.Q.: Unbounded parallel-batch scheduling with family jobs and delivery coordination. Information Processing Letters 111(12), 575-582 (2011).
13. Gong H., Chen D.H., Xu K.: Parallel-batch scheduling and transportation coordination with waiting time constraint. The Scientific World Journal 15, 1-8 (2014).
14. Lu L.F., Yuan J.J.: Unbounded parallel batch scheduling with job delivery to minimize makespan. Operations Research Letters 36, 477-480 (2008).
15. Cheng B.Y., Pei J., Li K., Pardalos P.M.: Integrated scheduling of production and distribution for manufacturers with parallel batching facilities. Optimization Letters 12, 1609-1623 (2018).
16. Dosa, G., Tan Z., Tuza, Z., Yan Y., Lnyi, C.S.: Improved bounds for batch scheduling with nonidentical job sizes. Naval Research Logistics 61(5), 351-358 (2014).