Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 8625849, 9 pages
https://doi.org/10.1155/2020/8625849

Research Article

Integrated Scheduling Problem on a Single Bounded Batch Machine with an Unavailability Constraint

Jing Fan

College of Arts and Science, Shanghai Polytechnic University, Shanghai 201209, China

Correspondence should be addressed to Jing Fan; sspu_fj@163.com

Received 28 July 2020; Revised 25 October 2020; Accepted 5 November 2020; Published 1 December 2020

Academic Editor: Chin-Chia Wu

We consider a scheduling problem where a set of jobs are first processed on a machine with an unavailability interval and, then, delivered to the customer directly. We focus on an integrated schedule of production and distribution such that the sum of the maximum delivery time and total delivery cost is optimized. We study two classes of processing machines in the production part. In the first class, the serial-batch machine, the processing time of a batch is the sum of the processing times of its jobs. In the second class, the parallel-batch machine, the processing time of a batch is the maximum processing time of the jobs contained in the batch. 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. Two patterns of job's processing, i.e., resumable and non-resumable, are considered if it is interrupted by the unavailability interval on the machine. In the distribution part, there are sufficient vehicles 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 show that these four problems are NP-hard in the strong sense in which the jobs have the same processing times and arbitrary sizes, and we propose an approximation algorithm for solving these four problems. Moreover, we show that the performance ratio of the algorithm is 2 for the serial-batch machine setting, and the error bound is 71/99 for the parallel-batch machine setting. We also evaluate the performance of the approximation algorithm by the computational results.

1. Introduction

Joint consideration of production and distribution is so beneficial in making higher level decisions for the manufacturers in the realistic supply chain environment that many researchers kept on studying on various models of such integrated scheduling problems. Potts [1] was probably the first researcher who considered scheduling with job delivery. Hall and Potts [2] studied integrated scheduling that involves a supplier, a manufacturer, and a customer. Up to now, there are an increasing number of literature and a variety of new models on scheduling problems with job delivery. For one machine, Chen and Vairaktarakis [3] studied the problem to minimize the weighted sum of maximum delivery time and total transportation cost with a relative preference, in which sufficient vehicles without capacity limit were used to deliver completed jobs to customers. They proposed two optimal algorithms for one customer and multiple customers. Chen [4] and Wang et al. [5] have surveyed the integrated scheduling problems of production and distribution operation.

In most models of scheduling with delivery, one machine processes one job at a time. However, a new kind of integrated scheduling problems concerned with batch processing has already been investigated by many researchers. Batch scheduling is motivated by many industrial manufacturing processes. For example, in the burn-in stage of semiconductor manufacturing, burn-in ovens can handle up to multiple jobs simultaneously. Generally, batch-processing machine has two versions: serial-batch and parallel-batch, according to [6]. When processed on the serial-batch machine, jobs may be batched and one job in a batch is processed at a time so that the processing time of a batch is the sum of the processing times of its jobs. Learning effect of jobs is always accompanied with the serial-batching scheduling problems. Lee et al. [7] and Pei et al. [8–10] investigated the serial-batching scheduling problems with deteriorating jobs or learning effects to minimize the makespan. There is no setup time in [7], but there are independent setup times in [8–10]. Pei et al. [11] considered the coordinated scheduling problem of production and transportation with deteriorating jobs. The batches including completed jobs would be delivered to a customer by a single vehicle with only one batch in one delivery. They analysed some useful properties and presented a heuristic for the general case and two optimal algorithms for two special cases. Lu et al. [12] studied the integrated production and delivery scheduling on a serial-batch machine to minimize the makespan and considered four different problems from the situations whether split is allowed in the production or delivery of the jobs.

When processed on the parallel-batch machine, several jobs can be processed as a batch simultaneously on a machine at one time such that the processing time of a batch is the maximum processing time of the jobs contained in the batch. The bounded parallel-batch-processing machine setting is introduced by Lee et al. [13], which is always encountered in burn-in operations in the semiconductor industry and heat treatment operations in the metalworking industries. Uzsoy [14] showed the problem is NP-hard to minimize makespan on a single bounded parallel-batch machine and provided some approximation algorithms. And Brucker et al. [15] discussed two variants: the unbounded model and the bounded model. Lee and Lee [16] developed a heuristic by iterative decomposition of a mixed integer programming model. Li et al. [17], Gong et al. [18], Lu and Yuan [19], and Cheng et al. [20] considered some integrated scheduling of production and distribution on parallel-batch-processing machines. In [20], the authors focused on finding a schedule of the jobs and a delivery plan so that the maximum delivery time of jobs is minimized. They showed that the problem can be solved optimally in O ($n\log n$) time if the jobs have identical sizes and proposed two approximation algorithms with an asymptotical performance ratio of 11/9 if the jobs have identical processing times and 2 if the jobs have arbitrary processing times and arbitrary sizes.

Up to now, most works have studied the machine setting which is always available. But, the machines can become unavailable during the production stage due to the occurrence of breakdowns or the necessity for maintenance and repair, which is called as the scheduling under the constraint of machine unavailability. When job processing is interrupted by machine unavailability, the interrupted job may be resumable or nonresumable once the machine is available again. In the resumable case, the interrupted job can be processed continuously, but in the nonresumable case, the interrupted job needs to be processed anew. For the integrated scheduling problem with a capacitated vehicle to minimize the maximum delivery time, Wang and Cheng [21] showed that it is NP-hard and proposed a 3/2-approximation algorithm and provided the instance with the worst-case ratio 3/2. More details on this research stream can be found in the work of Wang et al. [5] and Ma et al. [22]. There are a few literatures concerned with the integrated scheduling problem on the batch-processing machine with the unavailable constraints. Pei et al. [23] considered a single-machine serial-batching scheduling problem with a machine availability constraint, position-dependent processing time, and time-dependent setup time. Fan et al. [24] studied the integrated scheduling problem to minimize the maximum delivery time to the customer on a single bounded parallel-batch machine with an unavailability interval. For the nonresumable jobs, two approximation algorithms with the worst-case ratios 3/2 and 5/3 are explored in two cases where the jobs have the same size and arbitrary processing times, and the jobs have the same processing time and arbitrary sizes, respectively.

In this paper, we consider an integrated scheduling problem for a customer on a bounded batch machine with a machine unavailability period, where the jobs have identical processing times and sufficient capacitated vehicles are used to deliver the completed jobs to the customer. The objective is to minimize the sum of the delivery time when the last completed job is delivered to the customer and the total delivery cost. There are four classes of problems corresponding to different conditions during the processing of the interrupted job after the unavailability period and the type of machine, i.e., serial-batching and parallel-batching. In the practical aluminum-making process, serial-batching is common. Cylindrical aluminum ingots are processed one after another in the same batch on an extrusion machine, which has a schedule to be checked. Also, in the burn-in stage of semiconductor manufacturing, the burn-in oven can handle up to multiple jobs simultaneously, which is the form of parallel-batching. The oven must be stopped to accept inspection as a plan. All completed jobs have to be transported to the customer as soon as possible. Therefore, how to schedule the production and the delivery effectively is worth to study in such complicated settings.

2. Problem Formulation

A set of n jobs $\{J_1, J_2, \ldots, J_n\}$ is given, where job J_j has the processing time p_j and size s_j. Each job has an arbitrary size but an identical processing time $p_j = p$ for $j = 1, 2, \ldots, n$. In the production part, the machine has a capacity U, i.e., it can simultaneously process several jobs as a batch if their total size does not exceed the machine capacity. We discuss two types of batches, that is, serial-batching and parallel-batching. For the serial-batching, the processing time of a batch is the sum of the processing times of its jobs. For the parallel-batching, the processing time of a batch is the maximum processing time of the jobs contained in the batch. Meanwhile, the machine has an unavailability period $[\eta_1, \eta_2]$ because of maintenance and breakdown. Let η be the length of the unavailability period, i.e., $\eta = \eta_2 - \eta_1$. If there is, at least, one job in a batch that is interrupted by the unavailable period $[\eta_1, \eta_2]$, the interrupted job may be processed continuously just after η_2, in which case we call it resumable and denoted by $r - a$, or needs to be processed anew after η_2, defined as nonresumable and denoted by $nr - a$. In the latter case, the processing time of the batch containing this job is not affected for the parallel-batching mao the sum of η and the total processing time of the jobs and the idle time of the machine for the serial-batching machine. In the delivery part, there are sufficient vehicles to deliver the completed jobs to the customer and each of them has the same capacity V. Similar to [20], we assume that $V = xU$, where $x \geq 2$ and x is a positive integer. The transport price of one trip between the machine and the customer is denoted by c and the total delivery cost is denoted by TC. 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. Let $D_{\max}$ be the maximum delivery time of all the jobs. Our goal is to minimize the sum of maximum delivery time and total delivery cost. Using the five-field notation proposed by Chen [4] to denote an integrated scheduling problem, we denote the four classes of the problem in this paper as follows:

$(P1)$: $1, h_1|r-a, s-\text{batch}, s_j, p_j = p|V(\infty, c)$, direct$|1|D_{\max} + \text{TC}$

$(P2)$: $1, h_1|nr-a, s-\text{batch}, s_j, p_j = p|V(\infty, c)$, direct$|1|D_{\max} + \text{TC}$

$(P3)$: $1, h_1|r-a, p-\text{batch}, s_j, p_j = p|V(\infty, c)$, direct$|1|D_{\max} + \text{TC}$

$(P4)$: $1, h_1|nr-a, p-\text{batch}, s_j, p_j = p|V(\infty, c)$, direct$|1|D_{\max} + \text{TC}$

Here, h_1 denotes one unavailability interval and $V(\infty, c)$ presents the situation of the vehicle. Also, because there is one customer, the vehicle should transport jobs directly from the manufacturer to the customer. In the problems $(P1)$ and $(P2)$, *s-batch* means serial-batching. On the contrary, in the problems $(P3)$ and $(P4)$, *p-batch* means parallel-batching.

We organize the rest of the paper as follows. In Section 3, we show that all problems are NP-hard in the strong sense and propose some elementary properties. In Section 4, we present an approximation algorithm for these four problems and prove the different error bounds for these four problems. In the last section, we conclude our results and discuss the direction of the future research.

3. Properties of Four Problems

In this section, we analyse the computational complexity of the problems $(P1)$–$(P4)$.

Theorem 1. *Problems $(P1)$–$(P4)$ are all strongly NP-hard.*

Proof. Consider the special case (P') of problems $(P1)$ and $(P2)$, in which $p = 0$, $\eta_1 = \eta_2$, and $c = 0$, i.e., the processing time of each job is zero, and there is no unavailability interval on the machine and no delivery in problem (P'). Hence, the problem (P') is equivalent to minimize the number of batches, i.e., the bin-packing problem, which is a well-known strongly NP-hard problem. Therefore, problems $(P1)$ and $(P2)$ are strongly NP-hard.

Similarly, we construct the special case (P'') of problems $(P3)$ and $(P4)$, in which $\eta_1 = \eta_2$ and $c = 0$. Since each job has a processing time of p, every batch has a processing time of p. Hence, the problem (P'') is also equivalent to the bin-packing problem. Therefore, problems $(P3)$ and $(P4)$ are both strongly NP-hard.

Associating with the bin-packing problem, we study some properties of the optimal solutions for these four problems. □

Lemma 1. *There exists an optimal schedule γ_i^* of the problem (Pi) for $i = 1, 2, 3, 4$, satisfying the following properties:*

(1) Let X^* to be the number of batches in the optimal schedule γ_i^*; then, $X^* = \lceil n/U \rceil$

(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 β^* batches, and each of the last α^* deliveries x batches, where α^* and β^* are two positive integers satisfying $X^* = \alpha^* x + \beta^*$ and $0 < \beta^* \leq x$

This lemma can be proved similar with the proof in [21].

4. Algorithm and Analyses

To minimize the objective function value, we must assign many enough jobs in every batch to reduce the number of batches and deliveries. Therefore, associating with the algorithm of the bin-packing problem, we present the following algorithm to solve four scheduling problems.

4.1. Algorithm H

Step 1: reindex all jobs as $\{J_1, J_2, \ldots, J_n\}$ so that $s_1 \geq s_2 \geq \ldots \geq s_n$.

Step 2: use the First Fit Decreasing (FFD) rule to assign the jobs into batches, in which jobs are to be processed together on the machine. Build the first empty batch B_1 and put the job into it one by one in the sequence in Step 1 if the current total sizes of the batch B_1 are no more than U. When all the jobs have been checked and there are still jobs need to be assigned, build the second batch B_2 and assign the remaining jobs as the sequence in Step 1. Repeat the assignment until there is no job left. Let X be the number of batches produced in Step 2.

Step 3: assign the batches in a nondecreasing order of their processing times to process continuously from time 0 except the unavailability interval.

Step 4: Let. $X = \alpha x + \beta$ and $0 < \beta \leq x$ deliver the first completed β batches in D_1. Also, deliver x batches immediately in the following every delivery. When the last x batches are delivered to the customer, the algorithm finishes.

Recalling the FFD rule for the bin-packing problem, we can easily obtain Lemma 2.

Lemma 2 (see [25]). $X \leq 11/9X^{*} + 6/9$.

Based on Lemma 2, we can explore the corresponding values of X and X^{*} if $X^{*} \leq 9$ (see Table 1).

In order to list the corresponding relationship of values between X and X^{*} if $X^{*} \geq 10$, we use two positive integers $a \geq 1$ and $1 \leq b \leq 9$ such that $X^{*} = 9a + b$ similar to that in [26]. Hence, we can obtain Table 2.

Table 1: The corresponding values of X and X^{*} if $X^{*} \leq 9$.

Variables	Values								
X^{*}	1	2	3	4	5	6	7	8	9
X	1	2, 3	3, 4	4, 5	5, 6	6, 7, 8	7, 8, 9	8, 9, 10	9, 10, 11

Table 2: The maximal values of X and the upper bound of X/X^{*} if $X^{*} \geq 10$.

Variables	Values				
X^{*}	$9a+1$	$9a+2$	$9a+3$	$9a+4$	$9a+5$
The maximal value of X	$11a+1$	$11a+3$	$11a+4$	$11a+5$	$11a+6$
Upper bound of X/X^{*}	11/9	14/11	5/4	16/13	11/9
X^{*}	$9a+6$	$9a+7$	$9a$+8	$9a+9$	
The maximal value of X	$11a+8$	$11a+9$	$11a$+10	$11a+11$	
Upper bound of X/X^{*}	19/15	5/4	21/17	11/9	

Furthermore, we analyse the relationship of the objective function values produced by algorithm H and the optimal schedule in the problems $(P1)$–$(P4)$, respectively. For convenience, we use γ_i to denote the schedule obtained by algorithm H for the problem (Pi).

Lemma 3

(1) $d(P2) - d^{*}(P2) \leq np$, where $d(P2)$ and $d^{*}(P2)$ are the idle times on the machine before the unavailability interval in schedules γ_2, γ_2^{*} for the problem $(P2)$, respectively

(2) $d(P4) \leq d^{*}(P4)$, where $d(P4)$ and $d^{*}(P4)$ are the idle times on the machine before the unavailability interval in schedules γ_4, γ_4^{*} for the problem $(P4)$, respectively

Proof (1) According to Step 3 of Algorithm H, the batches are processed in a nondecreasing order of their processing times. Let n_b, n_b^{*} be the number of jobs completed before the beginning time of $[\eta_1, \eta_2]$ in schedules γ_2, γ_2^{*}, respectively. Because the length of the available time before η_1 on the machine is the fixed, we have $n_b p + d(P2) = n_b^{*} p + d^{*}(P2)$. Hence, $d(P2) - d^{*}(P2) \leq (n_b^{*} - n_b)p \leq n_b * p \leq np$.

(2) Because the processing time of each batch is equal to p no matter how many jobs are in it and the starting time of processing is zero by the algorithm H, it is obvious that the idle time in γ_4 must be no more than that in γ_4^{*}, i.e., $d(P4) \leq d^{*}(P4)$.

For the problem (Pi), we use $D_{\max}(Pi)$, $D *_{\max}(Pi)$, $\mathrm{TC}(Pi)$, and $\mathrm{TC}^{*}(Pi)$ to denote the maximum delivery time, the total delivery cost in γ_i, and γ_i^{*} $(i = 1, 2, 3, 4)$, respectively. We can easily conclude every value of $D_{\max}(Pi)$, $D^{*}_{\max}(Pi)$, $\mathrm{TC}(Pi)$, and $\mathrm{TC}^{*}(Pi)$ for $i = 1, 2, 3, 4$. □

Lemma 4 *(1) In the problem $(P1)$, $D_{\max}(P1) = \eta + np$, $TC(P1) = (\alpha + 1)c$ and $D *_{\max}(P1) = \eta + np$, $TC^{*}(P1) = (\alpha^{*} + 1)c$*

(2) In the problem $(P2)$, $D_{\max}(P2) = \eta + np + d(P2)$, $TC(P2) = (\alpha + 1)c$ and $D *_{\max}(P2) = \eta + np + d^{*}(P2)$, $TC^{*}(P2) = (\alpha^{*} + 1)c$

(3) In the problem $(P3)$, $D_{\max}(P3) = \eta + Xp$, $TC(P3) = (\alpha + 1)c$ and $D^{*}_{\max}(P3) = \eta + X^{*}p$, $TC^{*}(P3) = (\alpha^{*} + 1)c$

(4) In the problem $(P4)$, $D_{\max}(P4) = \eta + Xp + d(P4)$, $TC(P4) = (\alpha + 1)c$ and $D *_{\max}(P4) = \eta + X^{*}p + d^{*}(P4)$, $TC^{*}(P4) = (\alpha^{*} + 1)c$

Proof. Because the processes of batches are resumable in the problem $(P1)$, the difference between the result of the algorithm H and the optimal solution is only the number of deliveries. In the problem $(P2)$, the idle time of the machine is also different between the result of the algorithm H and the optimal solution. Moreover, in the problem $(P3)$, the total processing time X_p is needed in the solution of the algorithm H, but X_p^{*} is needed in the optimal solution. Also, the difference of the idle time of the machine is obvious in the objective function value of the algorithm H and the optimal schedule in the problem $(P4)$.

It is notable that the completion time of the last batch is larger than the ending time of the unavailability interval in the optimal schedule $\gamma_i *$ $(i = 1, 2, 3, 4)$. Otherwise, the performance ratio will approach infinity.

From the objectives of every problem, we find that the relationship between α and α^{*} is critical. Hence, we explore the following lemma. □

Lemma 5 (See [24]). $\alpha^{*} \leq \alpha < 11/9\alpha^{*} + 14/9$.

Proof. $\alpha^{*} \leq \alpha$ is obvious. Due to $x \geq 2$, Lemma 2, and the explanation of $X^{*} = \alpha^{*}x + \beta^{*}$, $X = \alpha x + \beta$, we can deduce

$$\alpha \leq \frac{X}{x} \leq \frac{1}{x}\left(\frac{11}{9}X^{*} + \frac{6}{9}\right) = \frac{1}{x}\left(\frac{11}{9}\left(\alpha^{*}x + \beta^{*}\right) + \frac{6}{9}\right) \leq \frac{11}{9}\alpha^{*} + \frac{14}{9}. \quad (1)$$

Moreover, if $\alpha = 11/9\alpha^{*} + 14/9$, we have

$$\frac{11}{9}X^* + \frac{6}{9} = \frac{11}{9}\left(\alpha^* x + \beta^*\right) + \frac{6}{9} \le \frac{11}{9}\alpha^* x + \left(\frac{11}{9}x + \frac{6}{9}\right) \le \frac{11}{9}\alpha^* x + \frac{14}{9}x = \alpha x < X, \tag{2}$$

which contradicts Lemma 2.

Consequently, we can easily obtain the corresponding values of α and α^* if $\alpha^* \le 4$ (see Table 3).

To simplify the expression the objective function value of each problem, we use $Z(Pi)$ and $Z^*(Pi)$ to denote the objective function values obtained by algorithm H and the optimal schedule, respectively. □

Theorem 2. *Algorithm H is a 2-approximation algorithm for solving the problem (P1) and the problem (P2), and the worst-case ratio is tight.*

Proof. According to Lemma 4, it is obvious that

$$\frac{Z(P1) - Z^*(P1)}{Z^*(P1)} = \frac{(\alpha - \alpha^*)c}{\eta + np + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)c}{\eta + np + \alpha^* c} \le \frac{\alpha - \alpha^*}{\alpha^*}. \tag{3}$$

From Table 3, we have that the algorithm reaches the worst-case ratio when $\alpha = 2$ and $\alpha^* = 1$.

For the problem $(P2)$, because of (1) in Lemma 3, we have

$$\frac{Z(P2) - Z^*(P2)}{Z^*(P2)} = \frac{(\alpha - \alpha^*)c + (\mathrm{d}(P2) - \mathrm{d}^*(P2))}{\eta + np + \mathrm{d}^*(P2) + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)c + np}{np + \alpha^* c} \le 1. \tag{4}$$

Table 3: The corresponding values of α and α^* if $\alpha^* \le 4$.

Variables	Values			
α^*	1	2	3	4
α	1, 2	2, 3	3, 4, 5	4, 5, 6

Similarly, when $\alpha = 2$ and $\alpha^* = 1$, the algorithm reaches the worst-case ratio.

Consider the following instance: $n = 6$, $p = 2$, $U = 7$, $x = 2$, $[\eta_1, \eta_2] = [2, 2 + \varepsilon]$, and $s_1 = s_2 = 3$, $s_3 = s_4 = s_5 = s_6 = 2$. The schedule produced by algorithm H is as follows: the first delivery including job J_6 as the first batch is delivered at time 2; the second delivery including two batches, which consist of jobs J_1 and J_2 and jobs J_3, J_4, and J_5 respectively, is delivered at time $12 + \varepsilon$. Hence, $Z(P1) = Z(P2) = 12 + \varepsilon + 2c$. However, in the optimal schedule, there is only one delivery, including two batches which consist of jobs J_1, J_3, and J_4 and jobs J_2, J_5, and J_6 respectively. This delivery occurs at time $12 + \varepsilon$, and the optimal objective value is $Z^*(P1) = Z^*(P2) = 12 + \varepsilon + c$. If c is sufficiently large, we have $Z(P1)/Z^*(P1) = Z(P2)/Z^*(P2) \longrightarrow 2$.

For the problem $(P3)$ and the problem $(P4)$, we obtain the error bound of algorithm H in the following theorem. □

Theorem 3. *Algorithm H has an error bound 71/99 when solving the problem (P3) and the problem (P4).*

Proof. Because of (2) of Lemma 3, we have

$$\begin{aligned}\frac{Z(P3) - Z^*(P3)}{Z^*(P3)} &= \frac{(X - X^*)p + (\alpha - \alpha^*)c}{\eta + X^* p + (\alpha^* + 1)c} \le \frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c}, \\ \frac{Z(P4) - Z^*(P4)}{Z^*(P4)} &= \frac{(X - X^*)p + d(P4) - d^*(P4) + (\alpha - \alpha^*)c}{\eta + X^* p + (\alpha^* + 1)c} \le \frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c},\end{aligned} \tag{5}$$

We only need analyse the upper bound of $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c$. Moreover, we prove the result in two cases according to $\alpha^* \le 4$ and $\alpha^* \ge 5$, respectively.

Case 1: $\alpha^* \le 4$. We discuss this case in two subcases.

Subcase 1.1: when we consider the situation that $\beta^* \le \beta$, we can obtain the following inequality for $\alpha = \alpha^* + k$ $(k = 0, 1, 2)$ according to Table 3:

$$(\alpha^* + k)x + \beta^* \le (\alpha^* + k)x + \beta = \alpha x + \beta = X \le \frac{11}{9}X^* + \frac{6}{9} = \frac{11}{9}\left(\alpha^* x + \beta^*\right) + \frac{6}{9}. \tag{6}$$

$$\left(k - \frac{2}{9}\alpha^* - \frac{2}{9}\right)x \le \frac{6}{9}. \tag{7}$$

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} = \frac{(\alpha - \alpha^*)xp + (\beta - \beta^*)p + (\alpha - \alpha^*)c}{(\alpha^* x + \beta^*)p + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)xp + xp + (\alpha - \alpha^*)c}{\alpha^* xp + (\alpha^* + 1)c}. \tag{8}$$

For $k = 0$, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le X - X^*/X^* \le 3/11 < 71/99$ according to Tables 1 and 2.

For $k = 1$, $\alpha^* = 4$, we have $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le xp + xp + c/\alpha^* xp + (\alpha^* + 1)c \le 2xp + c/4xp + 5c < 1/2 < 71/99$.

Subcase 1.2: when we consider the situation that $\beta^* > \beta$, we have the following inequality from Table 3 except for $\alpha^* = 1$ and $\alpha = 2$:

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} = \frac{(\alpha - \alpha^*)xp + (\beta - \beta^*)p + (\alpha - \alpha^*)c}{(\alpha^* x + \beta^*)p + (\alpha^* + 1)c} \le \frac{(\alpha - \alpha^*)xp + (\alpha - \alpha^*)c}{\alpha^* xp + (\alpha^* + 1)c} \le \frac{\alpha - \alpha^*}{\alpha^*} \le \frac{2}{3} < \frac{71}{99}. \tag{9}$$

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} \le \frac{X - X^*}{X^*} + \frac{\alpha - \alpha^*}{\alpha^* + 1}. \tag{10}$$

Moreover, associating with $\beta^* \le x$, we have

When k=0, k=1 with $\alpha^* = 4$, it is easy to deduce that inequality (7) holds when, $k = 1$ with $\alpha^* = 4$.

For $\alpha^* = 1$ and $\alpha = 2$, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha * + 1)c = (X - X^*)p + c/X^* p + 2c$. Moreover, we have $X/X^* \le 3/2$ according to Tables 1 and 2. Hence, $(X - X^*)p + (\alpha - \alpha^*)c/X^* p + (\alpha^* + 1)c \le 1/2 < 71/99$.

Case 2: $\alpha^* \ge 5$. We can easily obtain the inequality (10).

Because $X/X^* \le 14/11$ according to Table 2 and $\alpha - \alpha^*/\alpha^* + 1 < 2/9\alpha^* + 14/9/\alpha^* + 1 = 2/9 + 4/3 \cdot 1/\alpha^* + 1 \le 2/9 + 4/3 \cdot 1/6 = 4/9$ according to Lemma 5, we obtain

$$\frac{(X - X^*)p + (\alpha - \alpha^*)c}{X^* p + (\alpha^* + 1)c} \le \frac{3}{11} + \frac{4}{9} = \frac{71}{99}, \tag{11}$$

and $Z(P3) - Z^*(P3)/Z^*(P3) \le 71/99$ and $Z(P4) - Z^*(P4)/Z^*(P4) \le 71/99$. □

5. Computational Results

Since the four problems are all strongly NP-hard and the bin-packing problem is contained in these problems, solving them will be costly and time consuming. But, in order to evaluate the effectiveness of the algorithm H, we consider four numerical simulations for the small size problems. The optimal schedule is obtained by processing the batches according to the branch-and-bound algorithm for the bin-packing problem in [27] and delivering as (4) in Lemma 1.

The approximation algorithm and the branch-and-bound algorithm are coded in VC++, and the computational experiments are run on a personal computer. We utilize three different job numbers, $n = 30$ and 50. Let the processing time of each job be 1, and the sizes of all jobs are generated randomly from a discrete uniform distribution [1, 10]. Let the capacity of the machine $U = 3$, 7 and the capacity of the vehicle $V = xU = 2U, 10U$. We set two levels of the length of the unavailability period $\eta = 0.2$, 1.2 and two levels of the beginning time of the unavailability interval $\eta_1 = 3$, 10 in the experiments. The cost of each delivery batch is set as $c = 0.5$, 2. For each combination of n, η, η_1, and c, 20 random instances are used, and the computational results are summarized in from Tables 4 to 7, which reports the average number of seconds of CPU time of the optimal solutions and the solutions of algorithm H for each combination of parameters. The percentage error of the solution produced by the algorithm H is calculated as $\rho = Z(Pi) - Z^*(Pi)/Z^*(Pi)$.

It is observed from Tables 4–7 that the average CPU times increase in polynomial times for the algorithm H when the number of jobs increase and increase in exponential times for the branch-and-bound algorithm. It is found that the average worst-case error bounds of the optimal solution of $(P2)$ and $(P4)$ are generally more than $(P1)$ and $(P3)$ because of the nonresumable conditions. Finally, the most important thing is that these four tables show that the approximation algorithm performs efficiently and the actual error bounds of four problems are lower than the theoretical error bounds.

Table 4: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P1$).

n	U	x	η	η_1	c	Average CPU times of the algorithm H	Average CPU times of B&B algorithm	Average value of ρ
50	3	2	0.2	3	0.5	0.0603	15.0346	1.012
					2	0.0615	15.0191	1.013
			0.2	10	0.5	0.0618	15.0233	1.011
					2	0.0612	15.0212	1.011
			1.2	3	0.5	0.0609	15.0305	1.013
					2	0.0620	15.0145	1.012
			1.2	10	0.5	0.0605	15.0221	1.012
					2	0.0618	15.0314	1.013
50	7	10	0.2	3	0.5	0.0607	15.0324	1.011
					2	0.0608	15.0219	1.012
			0.2	10	0.5	0.0621	15.0309	1.012
					2	0.0624	15.0242	1.013
			1.2	3	0.5	0.0619	15.0287	1.011
					2	0.0618	15.0237	1.012
			1.2	10	0.5	0.0620	15.0228	1.010
					2	0.0621	15.0158	1.011

Table 5: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P2$).

n	U	x	η	η_1	c	Average CPU times of the algorithm H	Average CPU times of B&B algorithm	Average value of ρ
30	3	2	0.2	3	0.5	0.0512	3.5346	1.130
					2	0.0526	3.6121	1.126
			0.2	10	0.5	0.0518	3.5193	1.213
					2	0.0572	3.4023	1.204
			1.2	3	0.5	0.0501	3.5012	1.127
					2	0.0513	3.3986	1.123
			1.2	10	0.5	0.0542	3.4591	1.113
					2	0.0537	3.5029	1.138
30	7	10	0.2	3	0.5	0.0527	3.3092	1.214
					2	0.0528	3.4290	1.209
			0.2	10	0.5	0.0518	3.4232	1.120
					2	0.0519	3.5014	1.118
			1.2	3	0.5	0.0529	3.4879	1.104
					2	0.0608	3.3871	1.106
			1.2	10	0.5	0.0538	3.4381	1.169
					2	0.0521	3.3080	1.184

Table 6: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P3$).

n	U	x	η	η_1	c	Average CPU times of the algorithm H	Average CPU times of B&B algorithm	Average value of ρ
50	3	10	0.2	3	0.5	0.0601	15.0412	1.103
					2	0.0612	15.0381	1.104
			0.2	10	0.5	0.0623	15.0315	1.113
					2	0.0625	15.0406	1.118
			1.2	3	0.5	0.0614	15.0337	1.217
					2	0.0623	15.0341	1.212
			1.2	10	0.5	0.0625	15.0502	1.204
					2	0.0631	15.0513	1.210
50	7	2	0.2	3	0.5	0.0605	15.0372	1.132
					2	0.0611	15.0381	1.179
			0.2	10	0.5	0.0608	15.0401	1.124
					2	0.0621	15.0427	1.119
			1.2	3	0.5	0.0620	15.0381	1.221
					2	0.0619	15.0348	1.220
			1.2	10	0.5	0.0618	15.0329	1.217
					2	0.0623	15.0414	1.215

Table 7: The results of the algorithm H and branch-and-bound (B&B) algorithm for ($P4$).

n	U	x	η	η_1	c	Average CPU times of the algorithm H	Average CPU times of B&B algorithm	Average value of ρ
30	3	10	0.2	3	0.5	0.0525	3.5281	1.219
					2	0.0527	3.5356	1.193
			0.2	10	0.5	0.0541	3.5027	1.201
					2	0.0547	3.5301	1.208
			1.2	3	0.5	0.0535	3.5531	1.120
					2	0.0539	3.5890	1.119
			1.2	10	0.5	0.0541	3.6211	1.301
					2	0.0545	3.6236	1.271
30	7	3	0.2	3	0.5	0.0529	3.7210	1.271
					2	0.0532	3.8104	1.229
			0.2	10	0.5	0.0537	3.7147	1.201
					2	0.0539	3.7818	1.203
			1.2	3	0.5	0.0531	3.8201	1.308
					2	0.0610	3.9032	1.320
			1.2	10	0.5	0.0549	3.7138	1.301
					2	0.0556	3.9102	1.284

6. Conclusions

We consider four integrated scheduling problems to minimize the sum of the maximum delivery time and the total delivery cost, in which a set of jobs are first processed on a single batch machine with an unavailability interval and, then, delivered to the customer directly, and the jobs have the same processing time and arbitrary sizes. We show that these four problems are all strongly NP-hard and propose an approximation algorithm for them. Moreover, we obtain that the worst-case error bound of the algorithm for the serial-batch machine setting and the parallel-batch machine setting, respectively. We also provide computational results to evaluate the performance of the heuristics.

The direction of future research may focus on other settings of jobs and vehicles, such as jobs with arbitrary processing times and arbitrary sizes or one vehicle with limited capacity.

Data Availability

Radom data used to support this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (116013 16), the key discipline "Applied Mathematics" of Shanghai Polytechnic University (XXKP Y1604), the subproject of "first class postgraduate education leading plan," 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.

References

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

[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] Z.-L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.

[5] D. Y. Wang, O. Grunder, and A. E. Moudni, "Integrated scheduling of production and distribution operations: a review," *International Journal of Industrial and Systems Engineering*, vol. 19, no. 1, pp. 94–122, 2015.

[6] H. S. Mirsanei, B. Karimi, and F. Jolai, "Flow shop scheduling with two batch processing machines and nonidentical job sizes," *The International Journal of Advanced Manufacturing Technology*, vol. 45, no. 5-6, pp. 553–572, 2009.

[7] W. C. Lee, C. C. Wu, and P. H. Hsu, "A single-machine learning effect scheduling problem with release times," *Omega*, vol. 38, no. 1, pp. 3–11, 2010.

[8] J. Pei, X. Liu, P. M. Pardalos, W. Fan, and S. Yang, "Single machine serial-batching scheduling with independent setup time and deteriorating job processing times," *Optimization Letters*, vol. 9, no. 1, pp. 91–104, 2015.

[9] J. Pei, X. Liu, P. M. Pardalos, A. Migdalas, and S. Yang, "Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine," *Journal of Global Optimization*, vol. 67, no. 1, pp. 251–262, 2017.

[10] J. Pei, X. Liu, P. M. Pardalos et al., "Scheduling deteriorating jobs on a single serial batching machine with multiple job types and sequence-dependent setup times," *Annals of Operations Research*, vol. 249, no. 1, pp. 175–195, 2017.

[11] J. Pei, P. M. Pardalos, X. Liu, W. Fan, and S. Yang, "Serial batching scheduling of deteriorating jobs in a two-stage supply chain to minimize the makespan," *European Journal of Operational Research*, vol. 244, no. 1, pp. 13–25, 2015.

[12] L. Lu, L. Zhang, and L. Wan, "Integrated production and delivery scheduling on a serial batch machine to minimize the makespan," *Theoretical Computer Science*, vol. 572, pp. 50–57, 2015.

[13] C.-Y. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Operations Research*, vol. 40, no. 4, pp. 764–775, 1992.

[14] R. Uzsoy, "Scheduling a single batch processing machine with non-identical job sizes," *International Journal of Production Research*, vol. 32, no. 7, pp. 1615–1635, 1994.

[15] P. Brucker, A. Gladky, H. Hoogeveen et al., "Scheduling a batching machine," *Journal of Scheduling*, vol. 1, no. 1, pp. 31–54, 1998.

[16] Y. H. Lee and Y. H. Lee, "Minimising makespan heuristics for scheduling a single batch machine processing machine with non-identical job sizes," *International Journal of Production Research*, vol. 51, no. 12, pp. 3488–3500, 2013.

[17] S. Li, J. Yuan, and B. Fan, "Unbounded parallel-batch scheduling with family jobs and delivery coordination," *Information Processing Letters*, vol. 111, no. 12, pp. 575–582, 2011.

[18] H. Gong, D. Chen, and K. Xu, "Parallel-batch scheduling and transportation coordination with waiting time constraint," *The Scientific World Journal*, vol. 2014, Article ID 356364, 8 pages, 2014.

[19] L. Lu and J. Yuan, "Unbounded parallel batch scheduling with job delivery to minimize makespan," *Operations Research Letters*, vol. 36, no. 4, pp. 477–480, 2008.

[20] B.-Y. Cheng, J. Y.-T. Leung, K. Li, and S.-L. Yang, "Single batch machine scheduling with deliveries," *Naval Research Logistics (NRL)*, vol. 62, no. 6, pp. 470–482, 2015.

[21] X. Wang and T. C. E. Cheng, "Machine scheduling with an availability constraint and job delivery coordination," *Naval Research Logistics*, vol. 54, no. 1, pp. 11–20, 2007.

[22] Y. Ma, C. Chu, and C. Zuo, "A survey of scheduling with deterministic machine availability constraints," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 199–211, 2010.

[23] J. Pei, X. Liu, P. M. Pardalos, K. Li, W. Fan, and A. Migdalas, "Single-machine serial-batching scheduling with a machine availability constraint, position-dependent processing time, and time-dependent set-up time," *Optimization Letters*, vol. 11, no. 7, pp. 1257–1271, 2017.

[24] J. Fan, C. T. Ng, T. C. E. Cheng et al., "Single bounded parallel-batch machine scheduling with an unavailability constraint and job delivery," in *Algorithmic Aspects in Information and Management*Springer, Berlin, Germany, 2020.

[25] J. Pei, B. Cheng, X. Liu, P. M. Pardalos, and M. Kong, "Single-machine and parallel-machine serial-batching scheduling problems with position-based learning effect and linear setup time," *Annals of Operations Research*, vol. 272, no. 1, pp. 217–241, 2019.

[26] G. Dosa, Z. Tan, Z. Tuza, Y. Yan, and C. S. Lányi, "Improved bounds for batch scheduling with nonidentical job sizes," *Naval Research Logistics (NRL)*, vol. 61, no. 5, pp. 351–358, 2014.

[27] J. M. Valério, "Exact solution of bin-packing problems using column generation and branch-and-bound," *Annals of Operations Research*, vol. 86, pp. 629–659, 1999.