European Journal of Operational Research 000 (2016) 1–9

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Integrated scheduling on a batch machine to minimize production, inventory and distribution costs

Ba-Yi Cheng[a,b], Joseph Y-T. Leung[a,b,c,*], Kai Li[a,b]

[a] School of Management, Hefei University of Technology, Hefei 230009, PR China
[b] Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei 230009, PR China
[c] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07012, USA

ARTICLE INFO

Article history:
Received 1 March 2016
Accepted 4 September 2016
Available online xxx

Keywords:
Batch-processing machines
Production
Inventory
Distribution
Approximation algorithms

ABSTRACT

We consider the problem of scheduling a set of jobs on a single batch-processing machine. Each job has a size and a processing time. The jobs are batched together and scheduled on the batch-processing machine, provided that the total size does not exceed the machine capacity. The processing time of the batch is the longest processing time among all the jobs in the batch. There is a single vehicle to deliver the final products to the customer. If the vehicle has not returned, completed batches will be put into the inventory. In this paper, we consider the problem of minimizing the production, delivery and inventory costs. We show that if the jobs have the same size, there is an $O(n\log n)$-time algorithm to find an optimal solution. If the jobs have the same processing time, there is a fast approximation algorithm with an absolute worst-case ratio less than 1.783 and an asymptotic worst-case ratio equal to 11/9. When the jobs have arbitrary sizes and arbitrary processing times, there is a fast approximation algorithm with absolute and asymptotic worst-case ratios less than or equal to 2, respectively.

1. Introduction

Manufacturers are now facing intense competition in terms of price. In order to lower the price, each manufacturer must reduce its total cost which includes the production cost, delivery cost and inventory cost. Imagine a manufacturer receiving an order to produce certain goods for a customer. First, the products must be produced by one or more machines. After the products are produced, they must be delivered by one or more vehicles to the customer. If no vehicle is available, the completed product must be stored in a warehouse until a vehicle becomes available. Thus, there are three kinds of cost associated with this process: production cost, delivery cost and inventory cost. From the manufacturer point of view, he is interested in minimizing the total cost.

In this paper, we consider the problem of scheduling a set of jobs on a single batch-processing machine with a fixed capacity. A batch-processing machine can process several jobs together as a batch, provided that the total size of all the jobs in the batch does not exceed the capacity of the machine. Each job has a size and a processing time. The processing time of a batch is the longest processing time among all the jobs in the batch. The production cost is directly proportional to the total time taken to process all the jobs. There is a single vehicle to deliver the products to the customer. The travel time from the manufacturer to the customer is known and is fixed. The delivery cost will be proportional to the number of runs the vehicle makes. If the vehicle is not available (because it is making a delivery to the customer) when the batches are completed, the batches will be stored in a warehouse which incurs an inventory cost. Our problem is to find an integrated schedule that minimizes the total cost.

Batch-processing machines occur quite often in many industries such as semi-conductor companies, metal-working companies, porcelain companies and food-making companies. In all of these companies, the production cost is usually the highest since the processing time of a batch is usually quite long. Moreover, the production process consumes a lot of energy. The production cost is usually proportional to the total time taken to produce the products. The cost of each delivery can be calculated since we know the exact route from the manufacturer to the customer. For a given route, the delivery cost includes fuel cost, road fees and labor cost for driver and workers. The delivery cost is usually proportional to the number of deliveries, and it is much lower than the production cost. Completed batches that are not immediately delivered will be stored in a warehouse which will incur an inventory cost. The inventory cost is usually proportional to the time taken by the batches staying in the warehouse. Among the three costs,

* Corresponding author at: Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07012, USA. Fax: +1 973 596 5777.
E-mail addresses: leung@cis.njit.edu, leung5of6@gmail.com (J.Y-T. Leung).

http://dx.doi.org/10.1016/j.ejor.2016.09.009
0377-2217/

production cost is the most expensive one, followed by the delivery cost and finally followed by the inventory cost. Thus, in our schedule, we give the highest priority to the production cost, then the delivery cost and finally the inventory cost.

This type of scheduling problem integrates production cost with distribution and inventory costs. Unfortunately, even minimizing the production cost on a single batch-processing machine is intractable. Uzsoy (1994) showed that the problem of minimizing the makespan of a set of jobs with arbitrary sizes on a single batch-processing machine is strongly NP-hard. Approximation algorithms have been proposed to solve the single-machine scheduling problem (Cheng, Cai, Yang, & Hu, 2014a; Leung, Ng, & Cheng, 2008; Li, Li, Wang, & Liu, 2005; Zhang, Cai, Lee, & Wong, 2001). Tang, Meng, Chen, and Liu (2016) considered the batch scheduling problem arising in steel production and proposed a branch and price solution. They showed that benefits can be increased when the batch scheduling is done well. Giglio (2015) considered the single-machine problem with sequence-dependent batch setup and controllable processing times. Two methods are compared including a mathematical programming and optimal control strategies. Intelligent algorithms such as genetic algorithms (Peres & Monch, 2013; Zegordi, Abadi, & Nia, 2010) and simulated annealing algorithms (Melouk, Demodaran, & Chang, 2004) have also been used. Multi-machine problems have also been studied. Dosa, Tan, Tura, Yan, and Lanyi (2014) presented improved bounds for the single-machine and parallel-machine problems. The bounds are respectively 1.7 and 2. Liu, Ng, and Cheng (2009) and Jula and Leachman (2010) proposed algorithms for parallel batch-processing machines. Zhang, Cai, and Wong (2003) studied the online problem and provided an online algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2}$. Sung and Min (2001) and Cheng, Yang, Hu, and Li (2014b) considered the two-machine flowshop problem. Oulamara, Kovalyov, and Finke (2005) considered the flowshop problem with unbounded batch-processing machines and provided approximation algorithms with one, two and three batches. Current research has concentrated on scheduling problems that minimizes the production cost. Little attention has been paid to the integrated scheduling that minimizes the production and delivery costs. We note that in recent years integrated scheduling with the classical scheduling model has aroused interests of many researchers. We will briefly review these research next.

Hall and Potts (2003) considered general integrated scheduling problem for the manufacturer that includes material supplying, production, and product distribution. They demonstrated that cooperation between the supplier and manufacturer can reduce total system cost in the supply chain by at least 20%. Other researchers studied two types of integrated scheduling problems. The first type is the three-stage problem that includes the supplier, manufacturer, and customers. Selvarajah and Steiner (2009) proposed a 3/2 approximation algorithm for the problem of minimizing the delivery and inventory holding costs. Sawik (2009) extended the problem to a long-term product case. The objective is to minimize the overall cost of the supply chain, including the inventory holding, production and delivery costs. Yeung, Choi, and Cheng (2011) and Osman and Demirli (2012) considered the three-stage problem with time windows and synchronized replenishment, respectively. The three-stage problems involve three parts of the supply chain and are complex to solve. Other researchers studied the two-stage problems. However, many of them are still NP-hard (Chen & Vairaktarakis, 2005). The research on two-stage problems can be divided into two parts; i.e. the scheduling between the supplier and manufacturer and the scheduling between the manufacturer and customer. Chen and Hall (2007) investigated the conflict between the optimal schedules of the supplier and manufacturer and they proposed a cooperation mechanism for the two sides. Agnetis, Hall, and Pacciarelli (2006) proposed an interchange cost that will be incurred when the orders of the jobs are different in the optimal schedules of the two sides. They also provided a cooperation scheme. Algorithms for solving the supplier-manufacturer problem include approximation algorithms (Averbakh & Xue, 2007; Selvarajah & Steiner, 2006) and genetic algorithm (Naso, Surico, Turchaiano, & Kaymak, 2007). Two-stage problems between the manufacturer and customer have also been considered. Chen and Pundoor (2006) considered time-sensitive products and Hall and Liu (2010) considered capacity allocation. Cheng, Leung, Li, and Yang (2015) considered the two-stage scheduling problem where the manufacturer uses batch-processing machines to process jobs and deliver products to the customer. The objective is to optimize the service span, that is, the period lasting from the beginning of the production to the end of distribution. Averbakh and Baysan (2012) considered a semi-online integrated scheduling problem and proposed a semi-online algorithm with competitive ratio $\frac{2D}{D+P}$, where D is the cost of a delivery and P is the lower bound of the processing times of jobs. Little research has been done on integrated scheduling of batch-processing machines. Our paper can be viewed as a two-stage problem between the manufacturer and customer.

The rest of the paper is organized as follows. In Section 2, we give the model and some notations that will be used throughout the paper. In Section 3, we consider the special case where the jobs have identical sizes. We propose an optimal algorithm for this case. In Section 4, we consider the special case where the jobs have identical processing times. We propose a fast approximation algorithm and show that it has an absolute worst-case ratio less than 1.783 and an asymptotic worst-case ratio equal to 11/9. In Section 5, we consider the general case where the jobs have arbitrary sizes and arbitrary processing times. We provide a fast approximation algorithm with absolute and asymptotic worst-case ratios less than or equal to 2, respectively. Finally, in Section 6, we conclude the paper and present some directions for future research.

2. Model and notations

There are n jobs to be processed. The job set is $J = \{1, 2, \ldots, n\}$. Each job j has a size s_j and a processing time p_j. The manufacturer has a single batch-processing machine with a capacity of D. The batch set is $B = \{B_1, B_2, \ldots, B_z\}$, where z denotes the number of batches. The total size of all the jobs in a batch cannot exceed D. The processing of batch B_i cannot be interrupted until all the jobs in B_i are completed. Therefore, the processing time of B_i is $P_i = \max\{p_j | j \in B_i\}$. The production cost is PC given by the following equation:

$$PC = \lambda_1 \sum_{i=1}^{z} P_i, \tag{1}$$

where $\lambda_1 > 0$ is a constant. The production cost is a linear function of the production time. The products will be delivered to the customer by a single vehicle. The capacity of the vehicle is $K = rD$, where r is a positive integer. We assume that each batch is packaged in a standard-size box or pallet. Therefore, each delivery consists of an integral number of batches. The delivery set is $\{d_1, d_2, \ldots, d_x\}$ and the number of deliveries is x. The travel time of the vehicle from the manufacturer to the customer (and from the customer to the manufacturer) is T. The delivery cost DC is given by the following equation:

$$DC = \lambda_2 x, \tag{2}$$

where $\lambda_2 > 0$ is a constant. The delivery cost is a linear function of the number of deliveries. After a batch is completed, the prod-

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

ucts are kept in the inventory if they cannot be delivered immediately. The inventory cost IC is determined by the waiting time of the batch; i.e. the time between the completion of the batch and the delivery of the batch. The inventory cost IC is given by the following equation:

$$IC = \lambda_3 \sum_{i=1}^{z} q_i, \tag{3}$$

where q_i denotes the waiting time of the batch B_i and $\lambda_3 > 0$ is a constant. The inventory cost is a linear function of the waiting times of the batches. The total cost TC is given by

$$TC = PC + DC + IC. \tag{4}$$

Since λ_1, λ_2 and λ_3 are constants, we can normalize them so that TC can be expressed as

$$TC = \sum_{i=1}^{z} P_i + \alpha x + \beta \sum_{i=1}^{z} q_i. \tag{5}$$

Since we assume that the production cost is much higher than the delivery cost which in turn is much higher than the inventory cost, we have $1 > > \alpha > > \beta$.

We use the five-field notation of Chen (2010) to denote the problem under investigation: $\gamma_1|\gamma_2|\gamma_3|\gamma_4|\gamma_5$. γ_1 represents the machine configuration and γ_2 represents the constraints in the production. γ_3 represents the vehicle configuration. γ_3 is usually in the form of (η_1, η_2), where η_1 represents the number of vehicles and η_2 denotes the vehicle capacity. γ_4 represents the number of customers and γ_5 represents the objective function. Using the above notation, we use $1|p-batch, D, s_j, p_j|1, K|1|TC$ to denote our problem. In the production part, there is one batch-processing machine with capacity D. Job j has a size s_j and a processing time p_j. In the distribution part, there is one vehicle with capacity K. The manufacturer provides products for one customer. The travel time from the manufacturer to the customer is a constant T. The objective is to minimize the total cost for the manufacturer. We first study the special case Π_1, where each job has a size of one unit. Then we study another special case Π_2, where each job has the same processing time p. Finally, we study the general problem where the jobs have arbitrary processing times and arbitrary sizes. The problems are denoted as follows.

Π_1: $1|p-batch, D, s_j = 1, p_j|1, K|1|TC$
Π_2: $1|p-batch, D, s_j, p_j = p|1, K|1|TC$
Π_3: $1|p-batch, D, s_j, p_j|1, K|1|TC$

We will be making certain assumptions about the parameters. First, we assume that $r \geq 4$. This is a reasonable assumption since the vehicle usually has a large capacity. Second, we assume that the optimal solution has at least four deliveries, since the cases of three or less deliveries are not interesting.

Before we leave this section, we will give the definitions of *absolute worst-case ratio* and *asymptotic worst-case ratio*. For a given instance $\mathcal{I}$ of the problem and an approximation algorithm A, let $A(\mathcal{I})$ and $OPT(\mathcal{I})$ denote the total cost obtained by algorithm A and an optimization algorithm, respectively, when applied to $\mathcal{I}$. Let $R_A(\mathcal{I}) \equiv A(\mathcal{I})/OPT(\mathcal{I})$. The *absolute* worst-case ratio R_A for algorithm A is defined as

$$R_A \equiv \inf\{a \geq 1 : R_A(\mathcal{I}) \leq a \text{ for all } \mathcal{I}\}.$$

The *asymptotic* worst-case ratio R_A^{∞} is defined as

$$R_A^{\infty} \equiv \inf\{a \geq 1 : \text{ for some } N > 0, R_A(\mathcal{I}) \leq a \text{ for all } \mathcal{I} \text{ with } OPT(\mathcal{I}) \geq N\}.$$

3. Identical job sizes

In this section, we investigate the problem Π_1, where all the jobs have a size of one unit but the processing times are arbitrary. We will give an $O(n\log n)$-time algorithm to solve this problem. For convenience in the presentation, we mark the variables of an optimal solution as X^*. For example, z^*, PC^*, B_i^* and x^* represent the number of batches, the production cost, the ith batch and the number of deliveries in the optimal solution, respectively. For each z^*, we can find k^* and y^* such that $z^* = k^*r + y^*$ where $k^* \geq 3$ and $1 \leq y^* \leq r$. Clearly, the minimum number of deliveries in an optimal solution is $k^* + 1$. Thus, $x^* = k^* + 1 \geq 4$, which is our assumption made in Section 2. Similarly, we use z, PC, B_i and x to denote the number of batches, the production cost, the ith batch and the number of deliveries in the solution obtained by our algorithm, respectively. For each z, we can find k and y such that $z = kr + y$ where $1 \leq y \leq r$. For a given set of z batches, the minimum number of deliveries is $k + 1$. Thus, $x = k + 1$.

Our algorithm first groups the jobs into batches and it works as follows. The jobs are sorted in non-increasing order of their processing times. Then the first D jobs are put in the first batch B_1. The next D jobs are put in the second batch B_2, and so on until all the jobs have been batched. (The last batch may have less than D jobs.) Let there be z batches formed. We next reverse this set of batches into non-decreasing order of their processing times B'_1, B'_2, ..., B'_z. We partition these z batches into $k+1$ deliveries d_1, d_2, ..., d_{k+1} as follows: the first $k+1$ batches are assigned to d_1, d_2, ..., d_{k+1}, one batch per delivery. The next $k+1$ batches are again assigned to d_1, d_2, ..., d_{k+1}, one batch per delivery. We continue this process until all the batches have been assigned. (If $y < r$, there are some deliveries with less than r batches.) The batches within each delivery are processed in non-increasing order of their processing times.

Our algorithm, to be called Algorithm A1, is shown below.

Algorithm A1.

Step 1. Sort the jobs in non-increasing order of their processing times.

Step 2. Assign jobs into batches. Put the first D jobs into the first batch, i.e. B_1. Put the next D jobs into B_2. Repeat this assignment until all jobs have been batched. The obtained batch set is $\{B_1, B_2, \ldots B_z\}$. Let $z = kr + y$, where $1 \leq y \leq r$.

Step 3. Make the production schedule for batches. Reverse the batches in non-decreasing order of their processing times; i.e. B_z, B_{z-1},..., B_2, B_1 and re-label them as $B'_1, B'_2, \ldots, B'_z$. Form $k+1$ deliveries using the following rule: First, assign B'_1 as the last batch processed in d_1. In the same way, assign $B'_2, \ldots, B'_{k+1}$ as the last batch processed in $d_2, \ldots, d_{k+1}$, respectively. Then, assign the next $k+1$ batches as the second last batches processed in $d_1, \ldots, d_{k+1}$, respectively. Repeat this assignment until all the batches have been assigned. Batches within a delivery are processed continuously in non-increasing order of their processing times.

Step 4. Make the inventory and distribution schedule. Let the first batch of the first delivery begin its processing at time 0. Let $t = \sum_{B'_i \in d_1} P_i$. The first delivery is made at time t. Suppose we have completed the delivery of d_{l-1}. For the next delivery d_l, if $\sum_{B'_i \in d_l} P_i \geq 2T$, then start the processing of the first batch in d_l immediately. In this case, the vehicle will return before all the batches in d_l have finished processing. If $\sum_{B'_i \in d_l} P_i < 2T$, then wait for $(2T - \sum_{B'_i \in d_l} P_i)$ time units before we start processing the first batch in d_l. In this way, the last batch in d_l is finished at the same time that the vehicle returns. In the production of the batches in d_l, $1 \leq l \leq k+1$, the completed batches are put in the inventory after completion except the last batch.

Note that in Step 4 we delay the processing of some batches to minimize the inventory cost. If $\sum_{B'_i \in d_l} P_i < 2T$, the vehicle will not be able to return to the manufacturer at the same time the last batch in the delivery is completed. Thus, all the batches in the

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

Table 1
An instance.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
p_j	6	14	17	3	10	9	7	9	2	6	12	14	19	8	20	6	17	3	6	14
j^{LPT}	15	13	3	17	2	12	20	11	5	6	8	14	7	1	10	16	19	4	18	9
p_j^{LPT}	20	19	17	17	14	14	14	12	10	9	9	8	7	6	6	6	6	3	3	2

Table 2
Production schedule.

Batches	Jobs in the batch	Batch size	P_i
B_1	15, 13	2	20
B_2	3, 17	2	17
B_3	2, 12	2	14
B_4	20, 11	2	14
B_5	5, 6	2	10
B_6	8, 14	2	9
B_7	7, 1	2	7
B_8	10, 16	2	6
B_9	19, 4	2	6
B_{10}	18, 9	2	3

Table 3
Inventory and distribution schedule.

Deliveries	Batches	ST	CT	DT	q_i	AT
d_1	B_2	0	17	29	12	41
	B_6	17	26	29	3	41
	B_{10}	26	29	29	0	41
d_2	B_1	29	49	65	16	77
	B_5	49	59	65	6	77
	B_9	59	65	65	0	77
d_3	B_4	69	83	89	6	101
	B_8	83	89	89	0	101
d_4	B_3	92	106	113	7	125
	B_7	106	113	113	0	125

delivery have to wait for the vehicle to return, increasing the inventory cost.

Before we prove the optimality of Algorithm A1, we give an example to illustrate the working of the algorithm. Suppose there are 20 jobs to be processed; i.e. $n = 20$. All the jobs have the same size of one unit. The processing times are shown in Table 1. Line 1 represents the job indexes, and line 2 shows the processing times. The machine capacity is $D = 2$ and the vehicle capacity is $K = 3D = 6$. The travel time from the manufacturer to the customer is $T = 12$. Since $1 \gg \alpha \gg \beta$, we select the coefficients as $\alpha = 0.2$ and $\beta = 0.05$.

First, we sort the jobs in non-increasing order of their processing times. The results are shown in line 3 of Table 1, and line 4 shows their processing times.

Next, the obtained batches are shown in Table 2. Column 1 shows the batches and we see that all the jobs are assigned into 10 batches; i.e. $B_1, \ldots, B_{10}$. Column 2 lists the jobs in each batch. Columns 3 and 4 show the total size of jobs in the batch and the processing time of the batch.

By Step 3 of Algorithm A1, there are four deliveries. The first delivery consists of B_2, B_6 and B_{10}. The second delivery consists of B_1, B_5 and B_9. The third delivery consists of B_4 and B_8. The last delivery consists of B_3 and B_7. The production, inventory and distribution method are shown in Table 3. In Table 3, columns 1 and 2 show the deliveries and batches in each delivery. Columns 3 and 4 show the starting time (ST) and the completion time (CT) of production. We see that there is a *break* between the processing of the batches in d_2 and d_3 as well as between d_3 and d_4. The total processing times of the batches in d_3 and d_4 are both less than $2T$. Column 5 shows the departure time (DT) of each delivery and column 6 shows the inventory time of each batch. We see that in the production of the batches of a delivery, only the last batch needs no inventory. Column 7 shows the arrival time (AT) of each batch, i.e. the time when the products are delivered to the customer. Using the above parameters, the total cost is $TC = \sum_{i=1}^{10} P_i + 0.2 * 4 + 0.05(\sum_{i=1}^{10} q_i) = 106 + 0.8 + 0.05 * (12 + 3 + 16 + 6 + 6 + 7) = 109.3$.

We now prove the optimality of Algorithm A1. The next lemma shows that Algorithm A1 produces a set of batches exactly the same as the optimal solution.

Lemma 1. *Algorithm A1 finds an optimal production cost; i.e.* $PC(A1) = PC^*$.

Proof. We will show that the set of batches produced by Algorithm A1 is exactly the same as the set of batches produced by an optimization algorithm. Let job w have the longest processing time among all the jobs in the job set. Without loss of generality, we may assume that job w is put in the first batch B_1^*. Since the processing time of a batch is determined by the longest processing time of all the jobs in the batch, P_1^* will not change if we put the next $D - 1$ longest jobs into B_1^*. This has the advantage of lowering the production cost of the optimal solution. Repeating this argument, we see that the next D longest jobs are put in B_2^*, and so on. But Algorithm A1 produces exactly the same batches. Therefore, $B_1 = B_1^*$, $B_2 = B_2^*$, ..., $B_z = B_{z^*}^*$. The lemma follows. □

Since $z^* = k^*r + y^*$, the minimum number of deliveries is $x^* = k^* + 1$. By Lemma 1, we have $z = z^*$, $k = k^*$ and $x = x^*$. Thus, we have the following lemma.

Lemma 2. *Algorithm A1 finds an optimal distribution cost; i.e.* $DC(A1) = DC^*$.

We now consider the inventory cost; i.e. the total waiting time of the batches. The next lemma shows that Algorithm A1 finds an optimal inventory cost.

Lemma 3. *Algorithm A1 finds an optimal inventory cost; i.e.* $IC(A1) = IC^*$.

Proof. Consider the batches produced by Algorithm A1 as tasks in the sense of classical scheduling theory, T_1, T_2, ..., T_z, where $T_i = B_i'$ for each $1 \le i \le z$. The processing time of T_j, denoted by $\hat{p}_j$, is exactly the processing time of batch B_i'. These z tasks will be scheduled on $x = k + 1$ identical and parallel machines. Each task has the same due date DD which is very large; i.e. $DD > \sum_{j=1}^{z} \hat{p}_j$. We want to schedule these z tasks on the x machines so that the total earliness, $\sum_{j=1}^{z} E(T_j)$, is minimized. For each $1 \le j \le z$, $E(T_j) = DD - C(T_j)$ if T_j completes on or before DD (Note that $C(T_j)$ is the completion time of T_j); otherwise, $E(T_j) = \infty$.

To solve this scheduling problem, it is clear that we should schedule the shortest x jobs to complete at time DD, one job per machine. Then we schedule the next x shortest jobs to finish before DD, one job per machine, and so on, until all the jobs have been assigned. This schedule will minimize the total earliness $\sum_{j=1}^{z} E(T_j)$. It is easy to see that the total earliness is the same as the total waiting time of all the batches in the x deliveries; i.e. $\sum_{j=1}^{z} E(T_j) = \sum_{i=1}^{z} q_i$. (Recall that we position the processing of the batches so that when the last batch finishes processing, the

vehicle is ready to deliver.) Therefore, Algorithm A1 finds an optimal inventory cost. □

We note that for a given set of batches, Steps 3 and 4 of Algorithm A1 yield the minimum inventory cost. Using the above three lemmas, we have the following theorem.

Theorem 1. *Algorithm A1 is an optimal algorithm for* Π_1, *with a running time of* $O(n\log n)$.

Proof. By Lemmas 1–3, Algorithm A1 finds PC^*, DC^* and IC^*. Therefore, it is an optimal algorithm. As for the running time, Step 1 takes $O(n\log n)$ time. Each of Steps 2 to 4 takes $O(n)$ time. Therefore, the overall running time is $O(n\log n)$. □

4. Identical processing times

In this section, we consider Π_2 where the jobs have identical processing times but arbitrary sizes. We first determine the computational complexity of Π_2. Consider a special case of Π_2 where $\alpha=\beta=0$. This implies that no inventory or distribution cost is needed. So the problem is just a production scheduling problem and the objective is to minimize the production cost. In this case, the production cost is $PC=\sum_{i=1}^{z}P_i=zp$ and hence the objective is to minimize the number of batches. This problem is equivalent to the bin-packing problem which is known to be strongly NP-hard. We have the following proposition.

Proposition 1. Π_2 *is strongly NP-hard.*

Our algorithm, to be called Algorithm A2, is shown below.

Algorithm A2.

Step 1. Sort the jobs in non-increasing order of their sizes. Assign the jobs into batches by the First-Fit rule. Let the obtained batch set be $\{B_1,\ldots,B_z\}$. Let $z=kr+y$, $1\le y\le r$.

Step 2. Make $k+1$ deliveries as follows: assign the first $k+1$ batches, i.e., $B_1,\ldots,B_{k+1}$, as the first batch processed in $d_1,\ldots,d_{k+1}$, respectively. Assign the next $k+1$ batches, $B_{k+2},\ldots,B_{2k+2}$, as the second batch in $d_1,\ldots,d_{k+1}$, respectively. Repeat the assignment until there are h batches left where $h\le k+1$. Assign the h batches as the last batch processed in $d_1,\ldots,d_h$.

Step 3. Make the production schedule. Process the batches in the deliveries in the order of $d_1,d_2,\ldots,d_{k+1}$. Start the processing of the first batch in d_1 at time zero. Let $t=|d_1|*p$. Make the first delivery at time t. Suppose we have completed the delivery d_{l-1}. For the next delivery d_l, if $|d_l|*p<2T$, then wait for $(2T-|d_l|*p)$ time units before we start processing the first batch in d_l; otherwise, start the processing of the first batch of d_l immediately. Repeat this step until all the deliveries are made. The batches in each delivery are processed consecutively.

Step 4. Make the inventory and distribution schedule. When all the batches in a delivery have been completed and the vehicle is available, deliver all the jobs in the batches to the customer. When a batch is completed and no vehicle is available, put the batch in the inventory.

Let π_2 denote the solution obtained by Algorithm A2 and π_2^* denote the optimal solution. Let $z=kr+y$ be the number of batches in π_2 and $z^*=k^*r+y^*$ be the number of batches in π_2^*. Then, $x=k+1$ and $x^*=k^*+1$. Dosa, Li, Han, and Tuza (2013) have studied the First-Fit-Decreasing (FFD) algorithm for the bin-packing problem and showed that $g\le\frac{11}{9}g^*+\frac{6}{9}$, where g is the number of bins obtained by the FFD algorithm and g^* is the optimal number of bins. Since Step 1 of Algorithm A2 is exactly the FFD algorithm, we have the following proposition.

Proposition 2. $z\le\frac{11}{9}z^*+\frac{6}{9}$.

The total processing time of the batches in π_2 *is* $\sum_{i=1}^{z}P_i$. *By the assumptions in the last paragraph of Section 2, we have* $k^*\ge 3$ *and* $r\ge 4$. *So there are at least four deliveries in* π_2^* *and the number of batches is* $z^*\ge 3r+1\ge 13$. *Now we consider the total cost of* π_2. *We first investigate the production cost* $PC(A2)$. *Then we investigate* $DC(A2)$ *and* $IC(A2)$.

Lemma 4. *The production cost generated by Algorithm A2 satisfies* $PC(A2)/PC^*\le 149/117<1.28$.

Proof. By Proposition 2, we have

$$\frac{PC(A2)}{PC^*}=\frac{\sum_{i=1}^{z}P_i}{\sum_{i=1}^{z^*}P_i^*}=\frac{zp}{z^*p}\le\frac{\frac{11}{9}z^*+\frac{6}{9}}{z^*}=\frac{11}{9}+\frac{6}{9z^*}\le\frac{11}{9}+\frac{2}{39}=\frac{149}{117}<1.28. \tag{6}$$

This completes the proof of Lemma 4. □

In the distribution part, the distribution cost is determined by the number of deliveries. In π_2, $x=k+1$ and in π_2^*, $x^*=k^*+1$. By (5), $DC(A2)=\alpha(k+1)$ and $DC^*=\alpha(k^*+1)$.

Lemma 5. *The distribution cost generated by Algorithm A2 satisfies* $DC(A2)/DC^*<53/36<1.48$.

Proof. By Proposition 2, we have $z=kr+y\le\frac{11}{9}(k^*r+y^*)+\frac{6}{9}$. Since $1\le y,\ y^*\le r$, we have $kr\le\frac{11}{9}(k^*r+y^*)+\frac{6}{9}-y\le\frac{11}{9}(k^*r+r)+\frac{6}{9}-y<\frac{11}{9}(k^*+1)r$. That is,

$$\frac{k}{k^*+1}<\frac{11}{9}. \tag{7}$$

Therefore, we have

$$\frac{DC(A2)}{DC^*}=\frac{\alpha(k+1)}{\alpha(k^*+1)}=\frac{k+1}{k^*+1}=\frac{k}{k^*+1}+\frac{1}{k^*+1}<\frac{11}{9}+\frac{1}{4}=\frac{53}{36}<1.48. \tag{8}$$

This completes the proof of Lemma 5. □

Completed batches that cannot be delivered immediately are put in the inventory. By Step 3 of Algorithm A2, the last batch in each delivery is completed at a time when a vehicle becomes available for delivery. Since Steps 2 and 3 of Algorithm A2 are similar to Steps 3 and 4 of Algorithm A1, for a given set of batches, Algorithm A2 yields a minimum inventory cost. Suppose there are i batches in a delivery, then the inventory times of $B_1,\ldots,B_i$ are $(i-1)p,(i-2)p,\ldots,p,0$, respectively. The inventory cost of the batches in the delivery will be $\sum_{j=0}^{i-1}jp=\frac{i(i-1)}{2}p$.

Now consider the inventory cost of π_2^*. We can derive a lower bound for IC^* as the following proposition shows.

Proposition 3. $IC^*\ge\frac{z^*}{2}(\frac{z^*}{k^*+1}-1)$.

Proof. The number of deliveries in π^* is $x^*=k^*+1$. Let u_l^* denote the number of batches in d_l^*, $1\le l\le k^*+1$. We have $z^*=\sum_{l=1}^{k^*+1}u_l^*$. The total inventory time of the batches in d_l^* is $\frac{u_l^*(u_l^*-1)p}{2}=\frac{p}{2}((u_l^*)^2-u_l^*)$. Therefore, we have

$$IC^*=\frac{p}{2}\sum_{l=1}^{k^*+1}\left[(u_l^*)^2-u_l^*\right]=\frac{p}{2}\left[(u_1^*)^2+\cdots+(u_{k^*+1}^*)^2-z^*\right]$$
$$\ge\frac{p}{2}\left[\left(\frac{z^*}{k^*+1}\right)^2+\cdots+\left(\frac{z^*}{k^*+1}\right)^2-z^*\right]=\frac{p}{2}\left[\frac{(z^*)^2}{k^*+1}-z^*\right]$$
$$=\frac{z^*p}{2}\left(\frac{z^*}{k^*+1}-1\right). \tag{9}$$

If $\frac{z^*}{k^*+1}$ is an integer, then $IC^*=\frac{z^*p}{2}(\frac{z^*}{k^*+1}-1)$ holds. If $\frac{z^*}{k^*+1}$ is not an integer, then $IC^*>\frac{z^*p}{2}(\frac{z^*}{k^*+1}-1)$. □

Lemma 6. *The inventory cost generated by Algorithm A2 satisfies* $IC(A2)/IC^*<1.783$.

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

Proof. The proof of this lemma is rather complicated. We will postpone its proof until the Appendix A. □

Now we have the following theorem on the performance of Algorithm A2.

Theorem 2. *The absolute worst-case ratio of Algorithm A2 is less than 1.783 and the asymptotic worst-case ratio is 11/9. The running time of Algorithm A2 is $O(n\log n)$.*

Proof. First we show the time complexity of A_2. Step 1 is equivalent to the FFD rule for the bin-packing problem, which takes $O(n\log n)$ time. Step 2 takes $O(n)$ time. Steps 3 and 4 take linear time. Thus, the overall running time of Algorithm A2 is $O(n\log n)$.

Now we investigate the absolute worst-case ratio R_{A2} and the asymptotic worst-case ratio R_{A2}^{∞}. By Lemmas 4–6, we have

$$R_{A2} = \frac{PC(A2)+IC(A2)+DC(A2)}{PC^*+IC^*+DC^*} < \frac{1.28\times PC^*+1.783\times IC^*+1.48\times DC^*}{PC^*+IC^*+DC^*} \le 1.783. \quad (10)$$

When the number of jobs $n\to\infty$, we also have $k^*\to\infty$ and $z^*\to\infty$. By (6), we see that

$$\lim_{n\to\infty}\frac{PC(A2)}{PC^*}=\lim_{z^*\to\infty}\frac{PC(A2)}{PC^*}\le\lim_{z^*\to\infty}\left(\frac{11}{9}+\frac{6}{9z^*}\right)=\frac{11}{9}. \quad (11)$$

By (8), the distribution cost satisfies

$$\lim_{n\to\infty}\frac{DC(A2)}{DC^*}\le\lim_{k^*\to\infty}\left(\frac{11}{9}+\frac{1}{k^*+1}\right)=\frac{11}{9}. \quad (12)$$

By (7), we have

$$\lim_{n\to\infty}\frac{k^*+1}{k+1}<\lim_{k^*\to\infty}\frac{k^*+1}{\frac{11}{9}k^*}=\frac{9}{11}. \quad (13)$$

By (32) in the Appendix A, we have

$$\begin{aligned}\lim_{n\to\infty}\frac{IC(A2)}{IC^*}&\le\lim_{n\to\infty}\frac{k^*+1}{k+1}\times\frac{z}{z^*}\times\frac{z-(k+1)}{z^*-(k^*+1)}\\&\le\lim_{k^*\to\infty}\frac{k^*}{k}\times\lim_{z^*\to\infty}\frac{\frac{11}{9}z^*+\frac{6}{9}}{z^*}\\&\quad\times\lim_{k^*\to\infty}\frac{kr+y-(k+1)}{k^*r+y^*-(k^*+1)}\\&=\frac{9}{11}\times\frac{11}{9}\times\lim_{k^*\to\infty}\frac{k(r-1)+y-1}{k^*(r-1)+y^*-1}\\&=\lim_{k^*\to\infty}\frac{k}{k^*}=\frac{11}{9}.\end{aligned} \quad (14)$$

Therefore, by (11), (12) and (14), we have

$$R_{A2}^{\infty}=\lim_{n\to\infty}\frac{PC(A2)+DC(A2)+IC(A2)}{PC^*+DC^*+IC^*}=\frac{11}{9}. \quad (15)$$

The worst-case example realizing 11/9 is from the worst-case example of the FFD algorithm for bin-packing. This completes the proof of Theorem 2. □

5. The general problem

In this section, we consider the general problem Π_3, where the jobs have arbitrary sizes and processing times. We propose Algorithm A3 as follows.

Algorithm A3.

Step 1. Order the jobs in non-increasing order of their processing times. Batch the jobs by the First-Fit rule. Let the batches be denoted by E_1, E_2, ..., E_z with processing times Q_1, Q_2, ..., Q_z, respectively.

Step 2. Assign jobs into batches and deliveries. Find positive integers k and y such that $z=kr+y$, where $1\le y\le r$. The number of deliveries is $x=k+1$. Put the obtained batches in reverse order and re-label them as $\{B_1,B_2,\ldots,B_z\}$, with processing times P_1, P_2, ..., P_z, respectively. Make x deliveries as follows: assign the first x batches; i.e. $B_1,\ldots,B_x$, as the last batch processed in $d_1,\ldots,d_x$ respectively. Assign the next x batches, $B_{x+1},\ldots,B_{2x}$ as the second last batch in $d_1,\ldots,d_x$, respectively. Repeat the assignment until there are h batches left, where $h\le x$. Then assign the last h batches as the first batch processed in $d_1,\ldots,d_h$.

Step 3. Make the production schedule. Process the batches in the deliveries in the order of $d_1,d_2,\ldots,d_x$. Start the processing of the first batch in d_1 at time zero. Let $t=\sum_{B_i\in d_1}P_i$. Make the first delivery at time t. Suppose we have made $l-1$ deliveries. For the lth delivery, if $\sum_{B_i\in d_l}P_i<2T$, then wait for $(2T-\sum_{B_i\in d_l}P_i)$ time units before we start processing the first batch of d_l. Otherwise, start processing the first batch of d_l immediately when the last batch of d_{l-1} has completed. Batches in each delivery are processed consecutively.

Step 4. Make the inventory and distribution schedule. When all the batches in a delivery have finished processing and a vehicle is available, deliver all the jobs in the batches to the customer. When a batch is completed and cannot be delivered immediately, put the batch in the inventory.

The batching algorithm in Step 1 of Algorithm A3 is the same as the First-Fit method used in bin packing. Xia and Tan (2010) have given tighter bounds for the First-Fit algorithm used in bin packing. They showed that $g\le\frac{17}{10}g^*+\frac{7}{10}$, where g and g^* are the numbers of bins used by the First-Fit algorithm and the optimization algorithm, respectively. Moreover, they showed that the absolute performance ratio of First-Fit is at most $\frac{12}{7}$. From the results of Xia and Tan (2010), we have the following proposition.

Proposition 4. $z\le\frac{17}{10}z^*+\frac{7}{10}$ *and* $z\le\frac{12}{7}z^*$.

We will be considering the production cost, the delivery cost and the inventory cost in that order. For the convenience in proving an upper bound of PC(A3)/PC, we add* dummy batches *with zero processing times to the end of the batch list generated in Step 1 of Algorithm A3 so that $z=2z^*$. This has no effect on the production cost since the added batches have zero processing times.*

After the execution of Step 1 of Algorithm A3, the batches are in non-increasing order of their processing times. We let E_i and Q_i denote the batches and their processing times, respectively, $1\le i\le z$. However, after the execution of Step 2, the order of the batches are reversed. We let B_i and P_i denote the batches and their processing times, respectively, after Step 2. Clearly, $B_i=E_{z-i+1}$ and $P_i=Q_{z-i+1}$. Next, we consider the properties of E_i and Q_i.

Lemma 7. *After the execution of Step 1 of Algorithm A3, the jobs are in non-increasing order of their processing times. Let the job list be denoted as $1,2,\ldots,j,\ldots,n$. In Step 1, if job e satisfies $\sum_{j=1}^{e-1}s_j\le fD$ and $\sum_{j=1}^{e}s_j>fD$, then all the jobs in $\{1,2,\ldots,e\}$ are assigned to the batches in $\{E_1,\ldots,E_{2f}\}$.*

Proof. We prove the lemma by contradiction. Suppose job e is assigned to the batch E_{2f+1}. Clearly, the sum of the sizes of the jobs in E_{2i-1} and E_{2i} is larger than D, for $1\le i\le f$. Otherwise, a job in E_{2i} can be put in E_{2i-1}. When job e is assigned, jobs $1,2,\ldots,e-1$ have all been assigned, and they are assigned to E_j, $1\le j\le 2f$. Thus, $\sum_{j=1}^{e-1}s_j>fD$, contradicting our assumption that $\sum_{j=1}^{e-1}s_j\le fD$. □

After Step 1 of Algorithm A3, the batches are in non-increasing order of their processing times; i.e. $Q_1\ge\cdots\ge Q_{2z-1}\ge Q_{2z}$. We also order the batches in the optimal solution in non-increasing order of their processing times; i.e. $Q_1^*\ge\cdots\ge Q_{z^*-1}^*\ge Q_{z^*}^*$.

Lemma 8. $Q_{2i}\le Q_{2i-1}\le Q_i^*$ *for* $1\le i\le z^*$.

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

Proof. By Lemma 7, we see that job e is assigned to a batch in $\{E_1, \ldots, E_{2f}\}$. So any job j with $j < e$ is also assigned to a batch in $\{E_1, \ldots, E_{2f}\}$. Therefore, any job j' assigned to the following batches, i.e., E_i $(i = 2f+1, 2f+2 \ldots)$, will satisfy $p_{j'} \le p_e$. Because the processing time of a batch is no more than the longest processing time of all the jobs in the batch, we have

$$Q_{2f+2} \le Q_{2f+1} \le p_e. \tag{16}$$

In the optimal solution, since $\sum_{j=1}^{e} s_j > fD$, the total size of the e jobs is larger than fD. Therefore, the jobs in $\{1, 2, \ldots, e\}$ cannot all be assigned to the first f batches. Therefore, we have

$$Q^*_{f+1} \ge p_e. \tag{17}$$

Combining (16) and (17), we have $Q_{2f} \le Q_{2f-1} \le Q^*_f$ for all $f \ge 1$. □

In Step 2 of Algorithm A3, the batches are reordered and now they are in non-decreasing order of their processing times. Denote the batches and their processing times as B_i and P_i, respectively. Without loss of generality, we may assume that the batches are also ordered in non-decreasing order of their processing times in an optimal solution. By Lemma 8, we have the following lemma.

Lemma 9. *The production cost generated by Algorithm A3 satisfies* $PC(A3)/PC^* \le 2$.

Proof. By Lemma 8, we have $Q_{2f} \le Q_{2f-1} \le Q^*_f$ for all $f \ge 1$. Consequently, we have $P_{2i-1} \le P_{2i} \le P^*_i$ for $i = 1, 2, \ldots, z^*$ and $\sum_{i=1}^{z} P_i \le 2\sum_{i=1}^{z^*} P^*_i$. Therefore, $PC(A3)/PC^* \le 2$. □

To prove an upper bound of $DC(A3)/DC^*$, we will keep the batches generated by Algorithm A3 unchanged. We have the following lemma on the distribution cost.

Lemma 10. *The distribution cost of Algorithm A3 satisfies* $\frac{DC(A3)}{DC^*} < 1.95$. *When n approaches* ∞, $\lim_{n\to\infty} \frac{DC(A3)}{DC^*} = 1.7$.

Proof. Let $z = kr + y$ and $z^* = k^*r + y^*$. The number of deliveries in the optimal solution is $x^* = k^* + 1$. By Proposition 4, we have $z = kr + y \le \frac{17}{10}(k^*r + y^*) + \frac{7}{10}$; i.e. $kr \le \frac{17}{10}(k^*r + y^*) + \frac{7}{10} - y$ and $k \le \frac{17}{10}(k^* + \frac{y^*}{r}) + \frac{7}{10r} - \frac{y}{r}$. So the number of deliveries satisfies

$$\begin{aligned} x = k + 1 &\le \frac{17}{10}\left(k^* + \frac{y^*}{r}\right) + \frac{7}{10r} - \frac{y}{r} + 1 \\ &= \frac{17}{10}(k^* + 1) + \frac{7 - 10y}{10r} + 1 \le \frac{17}{10}(k^* + 1) - \frac{3}{10r} + 1 \\ &< \frac{17}{10}(k^* + 1) + 1. \end{aligned} \tag{18}$$

Then

$$\begin{aligned} \frac{DC(A3)}{DC^*} = \frac{\alpha x}{\alpha x^*} = \frac{k+1}{k^*+1} &< \frac{\frac{17}{10}(k^*+1)+1}{k^*+1} = \frac{17}{10} + \frac{1}{k^*+1} \\ &\le \frac{17}{10} + \frac{1}{4} = 1.95. \end{aligned} \tag{19}$$

When n approaches ∞,

$$\lim_{n\to\infty} \frac{DC(A3)}{DC^*} = \lim_{k^*\to\infty} \frac{DC(A3)}{DC^*} = \lim_{k^*\to\infty} \left(\frac{17}{10} + \frac{1}{k^*+1}\right) = 1.7. \tag{20}$$

This completes the proof of Lemma 10. □

Now we consider the inventory cost. For the convenience in proving an upper bound for $IC(A3)/IC^*$, we assume that $z = 2z^*$ by adding some dummy batches at the end of the batch list generated in Step 1 of Algorithm A3. The dummy batches have processing times so as to satisfy $P^*_i \ge P_{2i} \ge P_{2i-1}$ for $1 \le i \le z^*$. For example, if $z < 2z^*$ and z is even, then we can set $P_{z+1} = P_{z+2} = P^*_{z/2+1}$, $P_{z+3} = P_{z+4} = P^*_{z/2+2}$, and so on. If $z < 2z^*$ and z is odd, then we can set $P_{z+1} = P_z$, $P_{z+2} = P_{z+3} = P^*_{(z+1)/2+1}$, and so on. Let us call this Algorithm A3'. The remaining steps of Algorithm 3' are the same as Algorithm A3. Clearly, $IC(A3) \le IC(A3')$, which implies that $IC(A3)/IC^* \le IC(A3')/IC^*$. Thus, an upper bound for $IC(A3')/IC^*$ will serve as an upper bound for $IC(A3)/IC^*$.

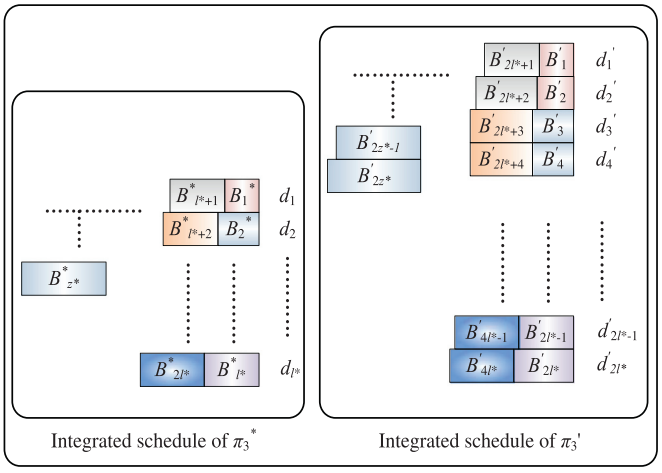

Fig. 1. Comparison of π^*_3 and π'_3.

Lemma 11. *The inventory cost generated by Algorithm A3 satisfies* $IC(A3)/IC^* \le 2$.

Proof. Let $\{B^*_1, B^*_2, \ldots, B^*_{z^*}\}$ denote the batch set of π^*_3, which is the optimal solution of Π_3. Let π'_3 denote the solution obtained by Algorithm 3'. We have $z' = 2z^*$ and $x' = 2x^*$. As discussed before, for a given set of batches, Steps 2 to 4 of Algorithm A3 gives an optimal inventory cost. Without loss of generality, we may assume that π^*_3 uses Steps 2–4 to generate the deliveries. As shown in Fig. 1, π'_3 is *twice* π^*_3. For each B^*_i in π^*_3, there are two corresponding batches B'_{2i-1} and B'_{2i} in π'_3. We have $P_{2i} \le P_{2i-1} \le P^*_i$. So, the inventory cost of π'_3 satisfies

$$IC(A3')/IC^* \le 2. \tag{21}$$

Thus, $IC(A3)/IC^* \le IC(A3')/IC^* \le 2$. □

By Lemmas 10 and 11, we have the following theorem on the performance of Algorithm A3.

Theorem 3. *The running time of Algorithm A3 is* $O(n\log n)$. *When solving* Π_3, $R_{A3} = TC(A3)/TC^* \le 2$ *and* $R^{\infty}_{A3} \le 2$.

6. Conclusions

In this paper, we have studied a class of integrated scheduling problems for manufacturers with batch-processing machines and arbitrary-size jobs. The objective is to minimize the total cost, which includes the production, delivery and inventory costs. We first consider the special case where the jobs have identical sizes. We show that the problem is solvable in polynomial time. Then we consider the special case where the jobs have identical processing times. We show that it is strongly NP-hard. We propose a fast approximation algorithm with absolute worst-case ratio less than 1.783 and asymptotic worst-case ratio equals to 11/9. Finally, for the general problem where the jobs have arbitrary sizes and arbitrary processing times, we provide a fast approximation algorithm with absolute worst-case ratio and asymptotic worst-case ratio less than or equal to 2, respectively.

From a computational complexity point of view, it is interesting to determine which factors make the problem intractable. Clearly,

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

the size of the job is the determining factor. When the sizes of the jobs are identical, the problem is solvable in polynomial time even when the processing times of the jobs are arbitrary. When the sizes of the jobs are arbitrary, the problem becomes intractable even when the processing times of the jobs are identical.

By the obtained results in this paper, we offer the following insights to managers of manufacturing companies.

1. The accurate integration of production, inventory and outbound distribution is feasible for manufacturing companies. The total cost of the three parts can be modeled and optimized. In order to obtain an effective integrated schedule, the key factors need to be analyzed in detail, including information about jobs, machines, inventory rules, vehicle configuration and distribution rules. The inventory part combines production and distribution, which makes the three parts an organic whole. Therefore, inventory part is especially important. If the integrated optimization model is given, effective methods can be designed to lower the total cost. It needs a good coordination among production, inventory and distribution units to achieve an optimal total cost.

2. The diversity of products aggravates the complexity of operations management for a manufacturing company. The diversity results from various demands of customers and it is unavoidable. For a decision maker, it is important to find an equilibrium between the functional efficiency and product diversity. In this paper, problem Π_1 is solvable in polynomial time and the unique difference between products in Π_1 is the processing time. Obviously, if all products are the same, i.e., they have the same size and processing time, then the integrated scheduling problem is easier to solve. Therefore, during the research and development phase of a new product, diversity should be carefully designed. Diversity is determined by product parameters. So, the parameters of products need an overall consideration, including the size, processing time, inventory and distribution requirements, etc.

3. There is a scale effect in the integrated scheduling. Observe the performance of $A2$ with the absolute worst case ratio being 1.783 but the asymptotic worst case ratio being only 1.223. This implies that when the number of products becomes sufficiently large, the performance becomes much better. Furthermore, observe the proof of Theorem 2. We find that the upper bound of performance is actually a decreasing function of the problem scale, which is determined by the number of products. Therefore, efficiency of integrated scheduling is influenced by the product demand. A better demand often leads to a better integrated scheduling efficiency. In other words, marketing is also important to the integrated scheduling of production and logistics.

For future research, there are many problems that are worthwhile to investigate. The first direction is the integrated scheduling problem involving the supplier and manufacturer. It is more difficult to find a cooperation scheme between the supplier and manufacturer, since the material supply and material inventory should be coordinated. The objective is to optimize the total cost of the supplier and manufacturer. The second direction for research is the integrated scheduling when the machine configuration is different. In practice, machine configurations such as parallel machines and flow shops are often encountered. The integrated scheduling problems with these machine configurations are also strongly NP-hard. Therefore, fast approximation algorithms with good performance ratios are much needed. Another interesting direction is the integrated scheduling problem with more than one customer. This type of scheduling problem may involve routing decisions.

Acknowledgments

We would like to thank the three anonymous referees whose suggestions have greatly improved the readability of the paper. This work is partly supported by the National Natural Science Foundation of China under Grants 71531008, 71671055, 71601065, 71131002 and 71471052. This work is also partly supported by the Fundamental Research Funds for the Central Universities under Grant JZ2016HGTB0727.

Appendix A. Proof of Lemma 6.

Proof. A lower bound for IC^* is shown in Proposition 3. Now we will derive an upper bound for $IC(A2)$. In π_2, the number of deliveries is $k+1$. In each delivery, the maximal number of batches is $\lceil \frac{z}{k+1} \rceil$. Therefore,

$$IC(A2) \le (k+1)\frac{\lceil \frac{z}{k+1} \rceil (\lceil \frac{z}{k+1} \rceil - 1)p}{2} = \frac{zp}{2}\left[\lceil \frac{z}{k+1} \rceil - 1\right]. \quad (22)$$

By (22) and (9), we have

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\le \frac{\frac{zp}{2}(\lceil \frac{z}{k+1} \rceil - 1)}{\frac{z^*p}{2}(\frac{z^*}{k^*+1} - 1)} \\ &\le \frac{z}{z^*} \times \frac{k^*+1}{k+1} \times \frac{(k+1)(\lceil \frac{z}{k+1} \rceil - 1)}{(k^*+1)(\frac{z^*}{k^*+1} - 1)} \\ &= \frac{k^*+1}{k+1} \times \frac{z}{z^*} \times \frac{z-(k+1)}{z^*-(k^*+1)}. \end{aligned} \quad (23)$$

We now consider various cases of k^*.

Case 1. $k^* = 3$.

In this case $13 \le 3r+1 \le z^* \le 4r$. So $z \le \frac{11}{9}z^* + \frac{6}{9} \le \frac{44r}{9} + \frac{r}{6} = \frac{91r}{18} < 6r$. That is, there are at most six deliveries in π and $k \le 5$. Since $z^* \ge 13$, $\frac{z}{z^*} \le \frac{11}{9} + \frac{6}{9z^*} \le \frac{149}{117}$. We now analyze the three subcases separately.

If $k = 3$. By (9), we have $IC^* \ge \frac{(3r+1)p}{2}(\frac{3r+1}{4} - 1)$. By (22), we have $IC(A2) \le 2rp(r-1)$. Thus, we have

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\le \frac{2rp(r-1)}{\frac{(3r+1)p}{2}(\frac{3r+1}{4} - 1)} = \frac{16r(r-1)}{(3r+1)(3r-3)} \\ &= \frac{16}{3} \times \frac{r}{3r+1} = \frac{16}{3} \times \frac{1}{3+\frac{1}{r}} < \frac{16}{3} \times \frac{1}{3} \\ &= \frac{16}{9} = 1.77\ldots \end{aligned} \quad (24)$$

If $k = 4$, by (23) we have

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\le \frac{4}{5} \times \frac{z}{z^*} \times \left(\frac{z}{z^*} + \frac{\frac{4z}{z^*} - 5}{z^* - 4}\right) \\ &\le \frac{4}{5} \times \frac{149}{117} \times \left(\frac{149}{117} + \frac{11}{1053}\right) < 1.34. \end{aligned} \quad (25)$$

If $k = 5$, then $\frac{4z}{z^*} - 6 \le \frac{4 \times \frac{91r}{18}}{3r} - 6 < 1$. So

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\le \frac{4}{6} \times \frac{z}{z^*} \times \left(\frac{z}{z^*} + \frac{\frac{4z}{z^*} - 6}{z^* - 4}\right) \\ &< \frac{2}{3} \times \frac{149}{117} \times \left(\frac{149}{117} + \frac{1}{9}\right) < 1.18. \end{aligned} \quad (26)$$

By (24)–(26), $IC(A2)/IC^* < 1.783$ when $k^* = 3$.

Case 2. $k^* \ge 4$.

Since $k^* \ge 4$ and $r \ge 4$, we have $z^* \ge 4r+1 \ge 17$. The worst case of $IC(A2)/IC^*$ occurs when π_2 has the maximum number of batches. By Proposition 2, $z \le \frac{11}{9}z^* + \frac{6}{9}$. Therefore, we consider the case where

$$z = \frac{11}{9}z^* + \frac{6}{9}. \quad (27)$$

Since $(k+1)r \ge z = \frac{11}{9}(k^*r + y^*) + \frac{6}{9}$, we have $k+1 \ge \frac{11}{9}k^* + \frac{11y^*+6}{9r}$. For $k^* \ge 4$,

$$\frac{k+1}{k^*+1} > \frac{\frac{11k^*}{9}}{k^*+1} = \frac{11}{9} - \frac{\frac{11}{9}}{k^*+1} \ge \frac{44}{45}. \quad (28)$$

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009

That is,

$$\frac{k^*+1}{k+1} < \frac{45}{44}. \tag{29}$$

Now we investigate an upper bound for z/z^*. Since $z^* \geq 17$, we can find positive integers u and i such that $z^* = 9u+i$ where $0 \leq i \leq 8$. We consider all the cases when $i = 0, \ldots, 8$, separately.

For $i = 0$, we have $z^* = 9u$ and $u \geq 2$ (since $z^* \geq 17$). By (27), $z = \frac{11}{9}(9u) + \frac{6}{9} = 11u + \frac{6}{9}$. Therefore, $\frac{z}{z^*} = \frac{11u+\frac{6}{9}}{9u} < \frac{11}{9} + \frac{2}{27u} \leq \frac{34}{27} < 1.26$.

For $i = 1$, we have $z^* = 9u+1$ and $u \geq 2$. By (27), we have $z = \frac{11}{9}(9u+1) + \frac{6}{9} = 11u + \frac{17}{9}$. So $\frac{z}{z^*} = \frac{11u+\frac{17}{9}}{9u+1} = \frac{11}{9} + \frac{\frac{6}{9}}{9u+1} \leq \frac{11}{9} + \frac{6}{9} \times \frac{1}{19} < 1.26$.

In a similar way, when $i = 2, 3, \ldots, 7$, $z^* = 9u+i$ and $u \geq 2$. The upper bounds of z/z^* for $i = 2, 3, \ldots, 7$ are $\frac{113}{90}, \frac{79}{63}, \frac{124}{99}, \frac{259}{207}, \frac{5}{4}$ and $\frac{281}{225}$, respectively. All of these ratios are less than 1.26.

For $i = 8$, we have $z^* = 9u+8$ and $u \geq 1$. By (27), we have $z = \frac{11}{9}(9u+8) + \frac{6}{9} = 11u + \frac{94}{9}$. So $\frac{z}{z^*} = \frac{11u+\frac{94}{9}}{9u+8} \leq \frac{193}{153} < 1.262$. Therefore, we have

$$\frac{z}{z^*} \leq 1.262. \tag{30}$$

Because $k^* \geq 4$ and $r \geq 4$, we have

$$\frac{z^*}{k^*+1} = \frac{k^*r+y^*}{k^*+1} \geq \frac{k^*r+1}{k^*+1} \geq \frac{4k^*+1}{k^*+1} \geq \frac{17}{5}. \tag{31}$$

By (23), we have

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\leq \frac{k^*+1}{k+1} \times \frac{z}{z^*} \times \frac{z-(k+1)}{z^*-(k^*+1)} \\ &\leq \frac{k^*+1}{k+1} \times \frac{z}{z^*} \times \left(\frac{z}{z^*} + \frac{\frac{z}{z^*}(k^*+1)-k-1)}{z^*-k^*-1}\right) \\ &= \frac{k^*+1}{k+1} \times \frac{z}{z^*} \times \left(\frac{z}{z^*} + \frac{\frac{z}{z^*} - \frac{k+1}{k^*+1}}{\frac{z^*}{k^*+1}-1}\right).\end{aligned} \tag{32}$$

By (29)–(31), we have

$$\begin{aligned}\frac{IC(A2)}{IC^*} &\leq \frac{k^*+1}{k+1} \times \frac{z}{z^*} \times \left(\frac{z}{z^*} + \frac{\frac{z}{z^*} - \frac{k+1}{k^*+1}}{\frac{z^*}{k^*+1}-1}\right) \\ &\leq \frac{45}{44} \times 1.262 \times \left(1.262 + \frac{1.262 - \frac{44}{45}}{\frac{17}{5}-1}\right) < 1.783.\end{aligned} \tag{33}$$

By the above discussions, we obtain $IC(A2)/IC^* < 1.783$ in all cases. □

References

Agnetis, A., Hall, N. G., & Pacciarelli, D. (2006). Supply chain scheduling: Sequence coordination. *Discrete Applied Mathematics, 154*, 2044–2063.

Averbakh, I., & Baysan, M. (2012). Semi-online two-level supply chain scheduling problems. *Journal of Scheduling, 15*(3), 381–390.

Averbakh, I., & Xue, Z. (2007). On-line supply chain scheduling problems with preemptions. *European Journal of Operational Research, 181*, 500–504.

Chen, Z. L. (2010). Integrated production and outbound distribution scheduling: Review and extension. *Operations Research, 58*, 130–148.

Chen, Z. L., & Hall, N. G. (2007). Supply chain scheduling: Conflict and cooperation in assembly systems. *Operations Research, 55*, 1072–1089.

Chen, Z. L., & Pundoor, G. (2006). Order assignment and scheduling in a supply chain. *Operations Research, 54*, 555–572.

Chen, Z. L., & Vairaktarakis, G. L. (2005). Integrated scheduling of production and distribution operations. *Management Science, 51*, 614–628.

Cheng, B. Y., Cai, J., Yang, S. L., & Hu, X. X. (2014a). Algorithms for scheduling incompatible job families on single batching machine with limited capacity. *Computers & Industrial Engineering, 75*, 116–120.

Cheng, B. Y., Leung, J. Y. T., Li, K., & Yang, S. L. (2015). Single batch machine scheduling with deliveries. *Naval Research Logistics, 62*, 470–482.

Cheng, B. Y., Yang, S. L., Hu, X. X., & Li, K. (2014b). Scheduling algorithm for flow shop with two batch-processing machines and arbitrary job sizes. *International Journal of Systems Science, 45*, 571–578.

Dosa, G., Li, R., Han, X., & Tuza, Z. (2013). Tight absolute bound for first fit decreasing bin packing: $FFD(I) \leq 11/9OPT(I) + 6/9$. *Theoretical Computer Science, 510*, 13–61.

Dosa, G., Tan, Z., Tura, Z., Yan, Y., & Lanyi, C. S. (2014). Improved bounds for batch scheduling with non-identical job sizes. *Naval Research Logistics, 61*(5), 351–358.

Giglio, D. (2015). Optimal control strategies for single-machine family scheduling with sequence-dependent batch setup and controllable processing times. *Journal of Scheduling, 18*(5), 525–543.

Hall, N. G., & Liu, Z. (2010). Capacity allocation in supply chains. *Operations Research, 58*, 1711–1725.

Hall, N. G., & Potts, C. N. (2003). Supply chain scheduling: Batching and delivery. *Operations Research, 51*, 566–584.

Jula, P., & Leachman, R. C. (2010). Coordinated multistage scheduling of parallel batch-processing machines under multi-resource constraints. *Operations Research, 58*, 933–947.

Leung, J. Y. T., Ng, C. T., & Cheng, T. C. E. (2008). Minimizing sum of completion times for batch scheduling of jobs with deteriorating processing times. *European Journal of Operational Research, 187*, 1090–1099.

Li, S. G., Li, G. J., Wang, X. L., & Liu, Q. M. (2005). Minimizing makespan on a single batching machine with release times and non-identical job sizes. *Operations Research Letters, 33*, 157–164.

Liu, L. L., Ng, C. T., & Cheng, T. C. E. (2009). Scheduling jobs with release dates on parallel batch processing machines. *Discrete Applied Mathematics, 157*, 1825–1830.

Melouk, S., Demodaran, P., & Chang, P. Y. (2004). Minimizing makespan for single machine batch-processing with arbitrary job sizes using simulated annealing. *International Journal of Production Economics, 87*, 141–147.

Naso, D., Surico, M., Turchaiano, B., & Kaymak, U. (2007). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research, 177*, 2069–2099.

Osman, H., & Demirli, K. (2012). Economic lot and delivery scheduling problems for multi-stage supply chains. *International Journal of Production Economics, 136*, 275–286.

Oulamara, A., Kovalyov, M. Y., & Finke, G. (2005). Scheduling a no-wait flow shop containing unbounded batching machines. *IIE Transactions, 37*(8), 685–696.

Peres, S. D., & Monch, L. (2013). Scheduling jobs on a single batch processing machine with incompatible job families and weighted number of tardy jobs objective. *Computers and Operations Research, 40*, 1224–1233.

Sawik, T. (2009). Coordinated supply chain scheduling. *International Journal of Production Economics, 120*, 437–451.

Selvarajah, E., & Steiner, G. (2006). Batch scheduling in a two-level supply chain – A focus on the supplier. *European Journal of Operational Research, 173*, 226–240.

Selvarajah, E., & Steiner, G. (2009). Approximation algorithms for the suppliers supply chain scheduling problem to minimize delivery and inventory holding costs. *Operations Research, 57*, 426–438.

Sung, C. S., & Min, J. I. (2001). Scheduling in a two-machine flowshop with batch processing machines for earliness/tardiness measure under a common due date. *European Journal of Operational Research, 131*, 95–106.

Tang, L. X., Meng, Y., Chen, Z. L., & Liu, J. Y. (2016). Coil batching to improve productivity and energy utilization in steel production. *Manufacturing & Service Operations Management, 18*(2), 262–279.

Uzsoy, R. (1994). Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research, 32*, 1615–1635.

Xia, B., & Tan, Z. (2010). Tighter bounds of the first fit algorithm for the bin-packing problem. *Discrete Applied Mathematics, 158*, 1668–1675.

Yeung, W. K., Choi, T. M., & Cheng, T. C. E. (2011). Supply chain scheduling and coordination with dual delivery models and inventory storage cost. *International Journal of Production Economics, 132*, 223–229.

Zegordi, S. H., Abadi, I. N. K., & Nia, M. A. B. (2010). A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. *Computers and Industrial Engineering, 58*, 373–381.

Zhang, G. C., Cai, X. Q., Lee, C. Y., & Wong, C. K. (2001). Minimizing makespan on a single batch-processing machine with nonidentical job sizes. *Naval Research Logistics, 48*, 226–240.

Zhang, G. C., Cai, X. Q., & Wong, C. K. (2003). Optimal on-line algorithms for scheduling on parallel batch processing machines. *IIE Transactions, 35*(2), 175–181.

Please cite this article as: B.-Y. Cheng et al., Integrated scheduling on a batch machine to minimize production, inventory and distribution costs, European Journal of Operational Research (2016), http://dx.doi.org/10.1016/j.ejor.2016.09.009