

Research Article

Batch Delivery Scheduling with Multiple Decentralized Manufacturers

Shi Li,¹ Xueling Zhong,² Hong Li,³ and Sujian Li³

¹ *Laboratoire Systèmes et Transports, Université de Technologie de Belfort-Montbéliard, 90000 Belfort, France*

² *Department of Computer, Guangdong University of Finance, Guangzhou 510520, China*

³ *School of Mechanical Engineering, University of Science and Technology, Beijing, China*

Correspondence should be addressed to Xueling Zhong; zhongxuel@hotmail.com

Received 13 April 2014; Accepted 16 July 2014; Published 31 August 2014

Academic Editor: Purushothaman Damodaran

Copyright © 2014 Shi Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses an integrated decision on production scheduling and delivery operations, which is one of the most important issues in supply chain scheduling. We study a model in which a set of jobs ordered by only one customer and a set of decentralized manufacturers located at different locations are considered. Specifically, each job must be assigned to one of the decentralized manufacturers to process on its single machine facility. Then, the job is delivered to the customer directly in batch without intermediate inventory. The objective is to find a joint schedule of production and distribution to optimize the customer service level and delivery cost. In our work, we discuss this problem considering two different situations in terms of the customer service level. In the first one, the customer service is measured by the maximum arrival time, while the customer service is measured by the total arrival time in the second one. For each situation, we develop a dynamic programming algorithm to solve, respectively. Moreover, we identify a special case for the latter situation by introducing its corresponding solutions.

1. Introduction

Supply chain scheduling has attracted a lot of attention in recent years. One of the important issues in this research area is the integration scheduling of production and delivery operations. In the make-to-order applications, manufacturers produce the ordered products and then deliver them to the customers directly without holding intermediate inventories. Nowadays, it is a quite common background in our social production activities, since it can improve the efficiency of certain production processes. In this context, the customer may cooperate with several manufacturers whose production capacity is of no significant difference. However, we sometimes have to face the fact that these manufacturers are located at decentralized places. In consequence, the transportation between each manufacturer and customer needs time and cost. This paper thus investigates a joint schedule of production and delivery in order to optimize both customer service level and delivery cost. Here, customer service level is measured by a function of the time when the

jobs are delivered to the customers, while the delivery cost consists of a fixed charge and a variable cost.

The rest of this paper is organized as follows. Section 2 reviews the related research work in the literature. In Section 3, we describe the problem addressed and introduce the notation used. Section 4 details the service measure form of maximum arrival time. Next, a dynamic programming algorithm is proposed to find an optimal solution. In Section 5, we first study the service measure form of total arrival time and then a dynamic programming algorithm is developed to solve it optimally as well. For this problem, we study a special case. Finally, our conclusions are given in Section 6.

2. Literature Review

A considerable amount of work has been done on batch delivery problems in the literature. Cheng and Kahlbacher [1] studied a single machine batch delivery scheduling to

minimize the sum of the total weighted earliness and delivery cost. In their work, the earliness of a job is defined as the difference between its delivery time and completion time. Cheng and Gordon [2] discussed a problem which involves a set of independent jobs that are to be partitioned into several batches and sequenced for processing by a single machine. Later, Cheng et al. [3] further show that this problem can be formulated as a classical parallel machine scheduling problem, and thus the complexity results and algorithms for the corresponding parallel machine scheduling problem can be easily extended to the problem. Cheng et al. [4] studied the single machine batch delivery problem to minimize the sum of the total weighted earliness and mean batch delivery time. Ji et al. [5] studied batch delivery scheduling on a single machine. The objective is to minimize the sum of the total weighted flow time and delivery cost. Wang and Cheng [6] studied parallel machine scheduling with batch delivery cost. The objective is to minimize the sum of the total flow time (i.e., equal weights) and delivery cost. Hall and Potts [7] studied a variety of problems where a supplier makes deliveries to several manufacturers, who also make deliveries to customers. The objective is to minimize the overall scheduling and delivery cost. Hall and Potts [8] considered several scheduling problems with batch delivery. The objective is to minimize the scheduling cost plus the delivery cost, and both single and parallel machine environments are considered. Note that delivery cost is defined as the nondecreasing function of the number of batch in [1–8].

In fact, most of the above research work considers customer service as their objective but without taking the transportation cost into account, or they assume that jobs can be delivered to customers instantaneously without any transportation delay. However, another set of batch delivery problems take transportation time and cost into account. Lee and Chen [9] incorporated transportation time and vehicle capacity requirements into machine scheduling models. They analyzed the complexity of various models in which jobs are first processed by machines and then delivered by one or more vehicles to a single customer. The vehicles have finite or infinite capacity and there is a transportation time for each customer of the delivery. Chang and Lee [10] extended Lee and Chen's work to the situation when each job occupies a different amount of space in the vehicle. Chen and Vairaktarakis [11] studied the problem of minimizing a convex combination of service level and distribution cost. The distribution cost of a delivery shipment consists of a fixed charge and a variable cost proportional to the total distance of the route taken by the shipment. Wang and Cheng [12] considered the identical parallel machine scheduling with job class setups and delivery to multiple customers. The delivery cost consists of a fixed charge and a variable cost related to the quantity of the delivery jobs. Li et al. [13] developed a single machine scheduling model that incorporated routing decisions of a delivery vehicle which serves customers at different locations. They considered the case where the single vehicle is infinite or bounded capacity. The objective is to minimize the sum of job arrival time. A similar problem setting of [13] is considered by Geismar et al. [14], who assume that a fixed limited number of trucks is available and the

objective is to minimize the time to produce and deliver all jobs to all customers. Chen and Lee [15] studies a general two-stage scheduling problem, in which jobs of different importance are processed by one first-stage processor and then, in the second stage, the completed jobs need to be batch delivered to various prespecified destinations in one of a number of available transportation modes. The objective is to minimize the sum of weighted job delivery time and total transportation cost.

It is worth mentioning that, different from all the above cases, Chen and Pundoor [16] studied a problem in which multiple decentralized manufacturers with nonidentical production capacity are located at different locations and have different production cost. Finished jobs are shipped to the single customer with fixed batch delivery cost, and each delivery batch has a capacity limit. The decision is to assign jobs to manufacturers to determine the processing sequences and to obtain a delivery schedule for the finished jobs.

In this paper, we focus on the integration of production scheduling and batch delivery model where multiple decentralized manufacturers are located at different locations, but only one customer is considered. More clearly, a set of jobs needs to be produced by manufacturers for a customer. After completing jobs, all the manufacturers must directly deliver them in batch to the customer by vehicle. We assume that each manufacturer has sufficient vehicles to use. A delivery batch incurs a fixed cost and a variable cost related to the number of jobs in that batch. The batch size is bounded by the capacity of the delivery vehicle. The objective is to determine the job assignment to multiple manufacturers and job processing sequence in each manufacturer together with the delivery schedule, so as to balance the customer service level and delivery cost. As introduced above, the problems discussed by Chen-Pundoor [16] are somewhat similar to ours. However, there are the following significant differences between two research works. (1) In our model a delivery batch of each manufacturer incurs not only a fixed cost but also a variable cost. Consequently, they make our problems more difficult to solve. (2) For the problem with the service level of the maximum arrival time, we propose a dynamic programming algorithm to solve it optimally, but Chen and Pundoor solved the similar problem by heuristics. (3) For the problem with the service level of the total arrival times, we propose a dynamic programming algorithm with time complexity of $O(mn^m)$ to solve it optimally, while Chen and Pundoor solved the problem with agreeable processing time and production cost by a dynamic programming algorithm with time complexity of $O(n^{m+1}mc^{2m})$. (4) We consider a special case of the problem with the service level of the total arrival time and solve it optimally by a polynomial algorithm. However, this point was not brought by Chen and Pundoor.

3. Problem Description

In this section, we describe our model formally. There are m manufacturers $M = \{M_1, M_2, \dots, M_m\}$, which are located at different locations. Each manufacturer has a single machine facility. A customer plans to order n job, denoted by $N = \{1, 2, \dots, n\}$. Each of these jobs should be assigned to one of

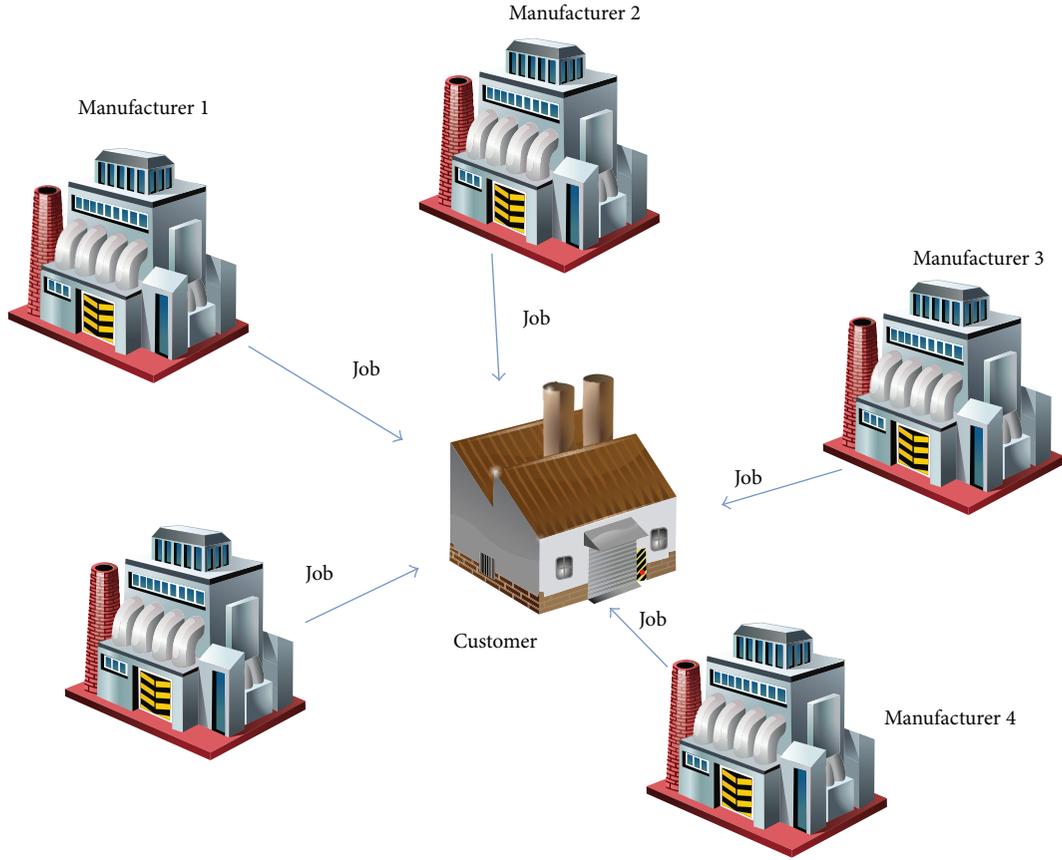


FIGURE 1: An example of the problem.

decentralized manufacturers to process on its single machine facility and then delivered to the customer directly in batch without intermediate inventory. More precisely, an example with 5 manufacturers and 1 customer is shown in Figure 1, where the lines represent the delivery of job. From this figure, as introduced earlier, we can see that there exist different manufactures located at different geographic areas but only one customer to consider.

Indeed, each job $j \in N$ has a processing time of p_j units. Job preemption is not allowed; that is, processing a job on a machine can not be interrupted until it is finished. All jobs and machines are available at time 0. All the finished jobs produced by each manufacturer need to deliver in batch to the customer by the vehicle. A vehicle used by each manufacture must depart only when all the assigned jobs have completed their processing. Suppose that there are enough homogeneous vehicles available to deliver jobs at each manufacturer location and each vehicle can be loaded at most c jobs in a batch. This indicates that the total number of jobs delivered in a batch is no greater than c . Let t_i denote the spending time when a vehicle travels from manufacturer i to the customer, for $M_i \in M$. Let A_j denote the time when the completed jobs arrive at the customer, for $j \in N$. With regard to the customer service level, we consider two particular forms:

- (i) maximum arrival time; that is, $A_{\max} = \max\{A_1, \dots, A_n\}$;
- (ii) total arrival times; that is, $\sum_{j \in N} A_j = A_1 + \dots + A_n$.

So, we consider this problem with two forms of customer service, level respectively. The batch delivery cost from manufacturer M_i to the customer needs total cost of $f_i + \gamma c_i$, where f_i is denoted by the fixed charge, $\gamma \leq c$ is denoted by the quantity of the delivered job in the batch, and c_i is denoted by the cost of each job delivered from M_i to the customer, for $M_i \in M$. Let B_{ik} denote the k th batch in M_i . Let $T = \sum_{i=1}^m (V_i f_i + \sum_{k=1}^{V_i} b_{ik} c_i) = \sum_{i=1}^m V_i f_i + n_i c_i$ be the total delivery cost, where V_i is denoted by the number of delivery batches in M_i , n_i is denoted by the number of jobs that assigned to process on the machine of M_i , and b_{ik} is denoted by the batch size of B_{ik} (the number of jobs in B_{ik}).

Adopting the extensive three-field notation, the problem can be denoted by $m \rightarrow 1, 1|c|Y+T$, where the first part $m \rightarrow 1$ of α field represents m manufacturers service for only single customer and the second part of α field represents that the processing facility in each manufacturer is single machine. The value of c under β represents the vehicle capacity. The objective functions that we consider under γ comprise a delivery cost T and a customer service cost Y , where $Y = A_{\max}$ or $Y = \sum_{j \in N} A_j$.

4. Maximum Arrival Time

In this section, we focus on problem $m \rightarrow 1, 1|c|A_{\max} + T$. We present the following lemma which establishes several properties for an optimal solution to the problem. The proof is trivial and omitted.

Lemma 1. *There exists an optimal solution to problem $m \rightarrow 1, 1|c|A_{\max} + T$ in which (1) there is no idle time at the beginning or between any two consecutively processed jobs on the machine of each manufacturer; (2) each batch delivery occurs when the last job of batch completes processing; (3) if manufacturer M_i has been assigned to n_i jobs, each of the first $\lceil n_i/c \rceil - 1$ batches includes c jobs and the last one includes $n_i - (\lceil n_i/c \rceil - 1)c$ jobs for $i = 1, 2, \dots, m$.*

First, we consider the complexity issues. On one hand, when both the delivery cost and delivery time are negligible, problem $m \rightarrow 1, 1|c|A_{\max} + T$ is reduced to the parallel machine scheduling problem with minimizing the makespan, denoted by $P||C_{\max}$. On the other hand, $P||C_{\max}$ is NP-hard in the ordinary sense when m is fixed and NP-hard in the strong sense when m is arbitrary. Thus, we have the the following theorem.

Theorem 2. *Even if $f_i = 0$, $c_i = 0$ and $t_i = 0$, for any $i = 1, 2, \dots, m$, problem $m \rightarrow 1, 1|c|A_{\max} + T$ is NP-hard in the ordinary sense when m is fixed and NP-hard in the strong sense when m is arbitrary.*

We now develop a dynamic programming algorithm to solve the problem. Let $P = \sum_{j=1}^n P_j$. The algorithm is formally described as follows.

Algorithm DPI

Step 1. Value function: define $H(q_1, \dots, q_m, g_1, \dots, g_m)$ as the minimum sum of the maximum arrival time and the total delivery cost, given that q_i jobs are assigned to manufacturer M_i , so far and the arrival time of the last job in these q_i jobs is equal to g_i exactly.

Step 2. Initialization: sequence the jobs arbitrarily.

For each $g_i = t_i, t_i + 1, \dots, t_i + P$, $q_i = 0, 1, \dots, n$, $i = 1, \dots, m$,

$$H(q_1, \dots, q_m, g_1, \dots, g_m) = \begin{cases} 0 & \text{if } q_1 = q_2 = \dots = q_m = 0, \\ g_1 = t_1, g_2 = t_2, \dots, g_m = t_m, & \\ \infty & \text{otherwise.} \end{cases} \quad (1)$$

Step 3. Recursive relations: for $g_i = t_i, t_i + 1, \dots, t_i + P$, $q_i = 0, 1, \dots, n$, $i = 1, \dots, m$,

$$H(q_1, \dots, q_m, g_1, \dots, g_m) = \min_{1 \leq i \leq m} \max \begin{cases} H(q_1, \dots, q_i - 1, \dots, q_m, \\ g_1, \dots, g_i - P, \dots, g_m) \\ + \left(\left\lceil \frac{q_i}{c} \right\rceil - \left\lceil \frac{q_{i-1}}{c} \right\rceil \right) f_i + c_i, \\ g_i + \sum_{\delta=1}^m \left(\left\lceil \frac{q_\delta}{c} \right\rceil f_\delta + q_\delta c_\delta \right). \end{cases} \quad (2)$$

Step 4. Optimal solution: $\min\{H(q_1, \dots, q_m, g_1, \dots, g_m) \mid \sum_{i=1}^m q_i = n, g_i = t_i, t_i + 1, \dots, t_i + P, q_i = 0, 1, \dots, n, i = 1, 2, \dots, m\}$.

Theorem 3. *Algorithm DPI solves problem $m \rightarrow 1, 1|c|A_{\max} + T$ optimally in $O(m(nP)^{m-1})$ time.*

Proof. Assume that the first $j - 1$ jobs have been assigned. When we assign the next job j to manufacturer M_i , if job j is the one with maximum arrival time, then the objective value of these first j jobs is equal to $g_i + \sum_{i=1}^m (\lceil q_i/c \rceil f_i + q_i c_i)$; otherwise, the objective value is the sum of $H(q_1, \dots, q_i - 1, \dots, q_m, g_1, \dots, g_i - P, \dots, g_m)$ and the delivery cost of job j ; that is, $(\lceil q_i/c \rceil - \lceil q_{i-1}/c \rceil) f_i + c_i$. This justifies (2). Since the value of optimal solution is determined by the minimum value with $q_1 + \dots + q_m = n$, we have justified that that Algorithm DPI solves the problem $m \rightarrow 1, 1|c|A_{\max} + T$ optimally.

We establish the running time complexity of Algorithm DPI as follows. Since $m - 1$ of the values q_1, \dots, q_m are independent and $m - 1$ of the values g_1, \dots, g_m are independent, the number of different states of the recursive relations is at most nP^{m-1} . For each state, the right-hand side of (2) can be calculated in $O(m)$ time. Thus, the overall time computational complexity of Algorithm DPI is $O(m(nP)^{m-1})$. \square

Theorem 3 implies that problem $m \rightarrow 1, 1|c|A_{\max} + T$ is NP-hard in the ordinary sense when m is fixed.

5. Total Arrival Times

5.1. General Case. In this subsection, we focus on problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$. When m is arbitrary, Chen and Pundoor proved the first problem in [16] is NP-hard in the strong sense by a reduction of the minimum cover problem, and when m is fixed, they pointed out that the computational complexity of the problem is open. If m is arbitrary, the computational complexity of problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$ is NP-hard in the strong sense. The proof is similar to the computational complexity proof of the first problem in [16]. If m is fixed, based on the ideas from Cheng et al. [17], for a parallel machine batching problem, we propose a polynomial algorithm for problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$. The following lemma establishes several properties for an optimal solution to this problem. The proof is trivial and omitted.

Lemma 4. *There exists an optimal solution to problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$ in which (1) there is no idle time at the*

beginning or between any two consecutively processed jobs on a machine in the manufacturer; (2) all jobs assigned to the same manufacturer are scheduled with an order of nondecreasing processing times; (3) each batch is delivered when the last job in the batch completes processing; (4) any two consecutive batches B_{ik} and B_{ik+1} satisfy $b_{ik} \geq b_{ik+1}$ in each manufacturer M_i .

Sort the n jobs from the single customer with an order of nondecreasing processing times such that $p_1 \leq p_2 \leq \dots \leq p_n$. We have the following property.

Lemma 5. *There exists an optimal solution to problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$ in which the jobs in each batch are sequenced continuously; that is, the jobs in a batch are $i, i + 1, \dots, j$, for some $i, j, 1 \leq i \leq j \leq n$.*

Proof. Given a schedule, the position weights on the machine of M_i are considered. Each job $j \in B_{ik}$ contributes its processing time p_j to the completion time of every jobs in $B_{ik} \cup B_{ik+1} \cup \dots \cup B_{iV_i}$. Hence, the position weight of each job in B_{ik} is equal to $w_{ik} = b_{ik} + b_{ik+1} + \dots + b_{iV_i}$. It is easy to see that the objective value of this schedule is equal to

$$\Delta = \sum_{i=1}^m \sum_{k=1}^{V_i} w_{ik} \sum_{l \in B_{ik}} p_l + (c_i + t_i) b_{ik} + V_i f_i. \quad (3)$$

If the number of batches at each manufacturer V_i ($i = 1, 2, \dots, m$) and the number of jobs in each batch b_{ik} ($i = 1, 2, \dots, m, k = 1, 2, \dots, V_i$) are fixed, then in order to minimize the value of Δ we should assign the job with the largest processing time to the position with the smallest position weight and the job with the second largest processing time to the position with the second smallest position weight and so on. Since the positions occupied by the jobs in the same batch have identical position weights, we get the desired result. \square

Based on Lemmas 4 and 5, we now develop a dynamic programming algorithm to solve problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$. The algorithm is formally described as follows:

Algorithm DP2

Step 1. Value function: define $H_j(q_1, q_2, \dots, q_m)$ as the minimum sum of the total arrival times and the delivery cost if the last $n - j + 1$ jobs have been considered and manufacturer M_i has been assigned q_i jobs among the considered jobs for $i = 1, 2, \dots, m$. Clearly, $\sum_{i=1}^m q_i = n - j + 1$.

Step 2. Initialization: set $j = n$. For each $\lambda = 0, 1, \dots, n, n + 1$, $q_i = 0, 1, \dots, n$, and $i = 1, \dots, m$,

$$H_\lambda(q_1, q_2, \dots, q_m) = \begin{cases} 0 & \text{if } \lambda = n + 1, q_1 = q_2 = \dots = q_m = 0, \\ \infty & \text{otherwise.} \end{cases} \quad (4)$$

Step 3. Recursive relations: for $q_i = 0, 1, \dots, n - j + 1$,

$$\begin{aligned} H_j(q_1, q_2, \dots, q_m) &= \min \{ H_u(q_1, \dots, q_i - (u - j), \dots, q_m) \\ &\quad + q_i P_{ju} + f_i + (u - j)(c_i + t_i) \mid j < u \\ &\leq \min \{ j + c, n + 1 \}, i = 1, \dots, m \}, \end{aligned} \quad (5)$$

where $P_{ju} = \sum_{l=j}^{u-1} p_l$. If $H_j(q_1, q_2, \dots, q_m) = H_\eta(q_1, \dots, q_\theta - (\eta - j), \dots, q_m) + q_\theta P_{j\eta} + f_\theta + (\eta - j)(c_\theta + t_\theta)$, then a batch containing jobs $j, \dots, \eta - 1$ is assigned to the beginning of the processed sequence at manufacturer M_θ .

If $j = 1$, then go to Step 4; otherwise, if $j \neq 1$, then set $j = j - 1$ and repeat Step 3.

Step 4. Optimal solution: $\min \{ H_1(q_1, q_2, \dots, q_m) \mid q_i = 0, 1, \dots, n, i = 1, \dots, m, \sum_{i=1}^m q_i = n \}$.

Theorem 6. *Algorithm DP2 solves problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$ optimally in $O(mn^m)$ time.*

Proof. By Lemma 5, we generate the batches containing consecutive jobs and construct the batch sequences in each manufacturers backwards in Algorithm DP2. Assume that manufacturer M_θ has been assigned α jobs in the current sequence and the batch containing jobs $j, \dots, \eta - 1 (\eta - j \leq c)$ is added to the beginning of the sequence. It is not difficult to calculate that the contribution of this batch to the objective value is $(\alpha + \eta - j)P_{j\eta} + f_\theta + (\eta - j)(c_\theta + t_\theta)$, where $P_{j\eta} = \sum_{l=j}^{\eta-1} p_l$ is the total processing time of the jobs in the batch. This contribution depends only on the batch and on the number of jobs assigned to manufacturer M_θ . Then, we can recursively compute the value of $H_j(q_1, q_2, \dots, q_m)$. Finally, the optimal solution can be obtained when $j = 1$. Thus, Algorithm DP2 solves the problem $m \rightarrow 1, 1|c| \sum_{j \in N} A_j + T$ optimally.

We establish the time complexity of Algorithm DP2 as follows. For each j , there are at most $(n - j + 2)^{m-1}$ different states of (q_1, q_2, \dots, q_m) . For each state, computing the right-hand side of (5) needs at most $O(c \cdot m)$ time. Therefore, the overall computational complexity of Algorithm DP2 is $O(n \cdot c \cdot m \cdot \sum_{j=1}^n (n - j + 2)^{m-1})$, equivalently $O(mn^m)$. \square

Theorem 6 shows that when the number of manufacturers m is fixed, Algorithm DP2 solves the problem optimally in polynomial time.

5.2. Special Case. In this subsection, we consider a special case with the objective function of $\sum_{j \in N} A_j + T$. In this special case, at most, one job can be delivered in a batch; that is, $c = 1$. Denote such a special case by $m \rightarrow 1, 1|c = 1| \sum_{j \in N} A_j + T$. It is easy to observe that when the delivery time and the delivery cost are negligible, problem $m \rightarrow 1, 1|c = 1| \sum_{j \in N} A_j + T$ is reduced to the classical parallel machine scheduling problem to minimize makespan, denoted by $P \parallel \sum C_i$. It is well known that $P \parallel \sum C_i$ is solved optimally by the generalized shortest-processing-time-first (SPT) rule, which can be described as follows: schedule the jobs with an order of nondecreasing

TABLE 1: The delivery time and delivery cost to the customer.

Manufacturers	1	2	3
t_i	1	4	2
$f_i + yc_i$	$2 + 6y$	$4 + 3y$	$5 + y$

TABLE 2: The processing time of jobs.

j	1	2	3	4	5	6
p_i	3	5	7	8	8	10

processing time, consider the jobs in this order and assign the current considering job to the earliest available machine. By Lemma 4, we develop an algorithm, denoted by SA, which is similar to SPT rule. Let G_i denote the current value of the objective function at manufacturer M_i when considering the first j jobs, g_i denote the total completion time of the jobs assigned to manufacturer M_i when considering the first j jobs, and G^* denote the optimal value of the objective function. Algorithm SA is shown in detail as follows.

Algorithm SA

Step 1. Sort and renumber the n jobs with an order of nondecreasing processing times such that $p_1 \leq p_2 \leq \dots \leq p_n$. Set $j = 1$. Set $G_i = 0$ and $g_i = 0$ for $i = 1, \dots, m$.

Step 2. WHILE $j \leq n$ DO
BEGIN

Find manufacturer M_u with $g_u + f_u + c_u + t_u = \min_{i=1,2,\dots,m} \{g_i + f_i + c_i + t_i\}$;

Assign job j to manufacturer M_u ; that is, set $g_u = g_u + p_j$;

Set $G_u = G_u + g_u + f_u + c_u + t_u$;

Set $j = j + 1$.

END

Step 3. Set $G^* = \sum_{i=1}^m G_i$.

Theorem 7. Algorithm SA solves problem $m \rightarrow 1, 1|c = 1| \sum_{j \in N} A_j + T$ optimally in $O(n \log n + mn)$ time.

The optimality of Algorithm SA can be easily justified. In terms of the time complexity, sorting the jobs with an order of nondecreasing processing times in Step 1 needs $O(n \log n)$ time, and executing Step 2 needs $O(mn)$ time since it takes $O(m)$ time to find the value of u . Hence, the overall running time complexity of algorithm SA is $O(n \log n + mn)$.

Now, we present a numerical example for this special case to demonstrate the optimality of the algorithm.

Example 8. Consider the instance with $n = 6$, $m = 3$, and $c = 1$. The delivery times and delivery cost to the customer are shown in Table 1. The processing times of jobs are given in Table 2.

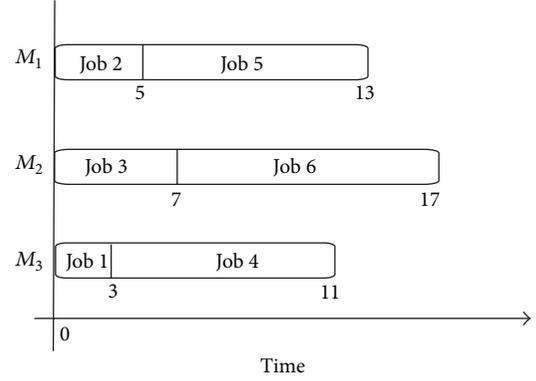


FIGURE 2: The result of Example 8.

Using Algorithm SA to solve the instance, we have the following results:

When $j = 1$, find $u = 3$, and so we have $g_3 = 3$ and $G_3 = 11$.

When $j = 2$, find $u = 1$, and so we have $g_1 = 5$ and $G_1 = 14$.

When $j = 3$, find $u = 2$, and so we have $g_2 = 7$ and $G_2 = 18$.

When $j = 4$, find $u = 3$, and so we have $g_3 = 11$ and $G_3 = 30$.

When $j = 5$, find $u = 1$, and so we have $g_1 = 13$ and $G_1 = 36$.

When $j = 6$, find $u = 2$, and so we have $g_2 = 17$ and $G_2 = 46$.

Therefore, the value of the optimal solution to Example 8 generated by Algorithm SA is equal to $G^* = G_1 + G_2 + G_3 = 35 + 46 + 30 = 112$. In such an optimal solution, the assignment of all the jobs to manufacturers M_1 , M_2 , and M_3 and the schedule on the machine of each manufacturer are illustrated in detail in Figure 2. In Figure 2, the horizontal axis represents time. In order to explain Figure 2, we see job 2. Job 2 is assigned to manufacturer M_2 ; it starts to process on the single machine of manufacturer M_2 at the point of time 0 and completes to process at the point of time 5 (shown in Figure 2). Once job 2 is completed to process, it is delivered as a batch to its customer immediately and needs 1 unit of delivery time and 8 units of delivery cost. Note that there does not show the delivery time and delivery cost of each job in Figure 2.

6. Conclusion

In this paper, we study a joint schedule of production and distribution to optimize the customer service level and delivery cost. Moreover, two different situations are considered by measuring the customer service in terms of: (i) total arrival times and (ii) maximum arrival time. To deal with these two situations, two dynamic programming algorithms are introduced, respectively, which are pseudopolynomial when

the number of manufacturers is fixed. We also identified a solvable special case for the latter situation. Compared with the related research work in the literature, we conclude our following contributions. (1) We extend the previous models with one manufacturer to a novel one with multiple decentralized manufacturers. (2) A delivery batch only incurs a fixed cost in the previous models. However, we consider here a model with a complicated cost structure in which a delivery batch of each manufacturer incurs not only a fixed cost but also a variable cost. (3) We propose the efficient dynamic programming algorithms to solve this problem optimally. However, heuristics was often developed for similar problems in the literature.

For future research, it will be worth considering the design of efficient and effective heuristics for the problems considered here. Our interesting research topic is also to study the extension of the current scheduling model with other objective functions, such as minimizing the total (weighted) tardiness or the number of the tardy jobs. Moreover, alternative machine configurations like flowshop facilities is a promising direction to study. At last, it might be worthy to extend the model with multiple decentralized manufacturers and multiple decentralized customers.

Conflict of Interests

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

Acknowledgments

The authors thank the AE and the anonymous referees for their helpful comments and suggestions. This research was supported in part by Humanities and Social Sciences Research Foundation of Ministry of Education of China (13YJC630239) and Guangdong Philosophical and Social Science Planning Fund Program (GD11XGL18).

References

- [1] T. C. E. Cheng and H. G. Kahlbacher, "Scheduling with delivery and earliness penalties," *Asia-Pacific Journal of Operational Research*, vol. 10, pp. 145–152, 1993.
- [2] T. C. E. Cheng and V. S. Gordon, "Batch delivery scheduling on a single machine," *Journal of the Operational Research Society*, vol. 45, no. 10, pp. 1211–1215, 1994.
- [3] T. C. E. Cheng, V. S. Gordon, and M. Y. Kovalyov, "Single machine scheduling with batch deliveries," *European Journal of Operational Research*, vol. 94, no. 2, pp. 277–283, 1996.
- [4] T. C. E. Cheng, M. Y. Kovalyov, and B. M. T. Lin, "Single machine scheduling to minimize batch delivery and job earliness penalties," *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 547–559, 1997.
- [5] M. Ji, Y. He, and T. C. E. Cheng, "Batch delivery scheduling with batch delivery cost on a single machine," *European Journal of Operational Research*, vol. 176, no. 2, pp. 745–755, 2007.
- [6] G. Wang and T. C. E. Cheng, "Parallel machine scheduling with batch delivery costs," *International Journal of Production Economics*, vol. 68, no. 2, pp. 177–183, 2000.
- [7] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.
- [8] N. G. Hall and C. N. Potts, "The coordination of scheduling and batch deliveries," *Annals of Operations Research*, vol. 135, pp. 41–64, 2005.
- [9] C. Y. Lee and Z. L. Chen, "Machine scheduling with transportation considerations," *Journal of Scheduling*, vol. 4, no. 1, pp. 3–24, 2001.
- [10] Y. Chang and C. Lee, "Machine scheduling with job delivery coordination," *European Journal of Operational Research*, vol. 158, no. 2, pp. 470–487, 2004.
- [11] Z.-L. Chen and G. L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.
- [12] X. L. Wang and T. C. E. Cheng, "Heuristics for parallel-machine scheduling with job class setups and delivery to multiple customers," *International Journal of Production Economics*, vol. 119, no. 1, pp. 199–206, 2009.
- [13] C. L. Li, G. Vairaktarakis, and C. Y. Lee, "Machine scheduling with deliveries to multiple customer locations," *European Journal of Operational Research*, vol. 164, no. 1, pp. 39–51, 2005.
- [14] H. N. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah, "The integrated production and transportation scheduling problem for a product with a short lifespan," *INFORMS Journal on Computing*, vol. 20, no. 1, pp. 21–33, 2008.
- [15] B. Chen and C. Lee, "Logistics scheduling with batching and transportation," *European Journal of Operational Research*, vol. 189, no. 3, pp. 871–876, 2008.
- [16] Z. L. Chen and G. Pundoor, "Order assignment and scheduling in a supply chain," *Operations Research*, vol. 54, no. 3, pp. 555–572, 2006.
- [17] T. C. E. Cheng, Z. L. Chen, M. Y. Kovalyov, and B. M. T. Lin, "Parallel-machine batching and scheduling to minimize total completion time," *IIE Transactions*, vol. 28, no. 11, pp. 953–956, 1996.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

