Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 804968, 7 pages
http://dx.doi.org/10.1155/2015/804968

Research Article

Integrated Production and Distribution Scheduling Problems Related to Fixed Delivery Departure Dates and Weights of Late Orders

Shanlin Li and Maoqin Li

Department of Mathematics, Taizhou University, Taizhou, Zhejiang 317000, China

Correspondence should be addressed to Shanlin Li; sxtyxdljz@163.com

Received 29 July 2014; Accepted 2 February 2015

Academic Editor: Dar-Li Yang

We consider an integrated production and distribution scheduling problem faced by a typical make-to-order manufacturer which relies on a third-party logistics (3PL) provider for finished product delivery to customers. In the beginning of a planning horizon, the manufacturer has received a set of orders to be processed on a single production line. Completed orders are delivered to customers by a finite number of vehicles provided by the 3PL company which follows a fixed daily or weekly shipping schedule such that the vehicles have fixed departure dates which are not part of the decisions. The problem is to find a feasible schedule that minimizes one of the following objective functions when processing times and weights are oppositely ordered: (1) the total weight of late orders and (2) the number of vehicles used subject to the condition that the total weight of late orders is minimum. We show that both problems are solvable in polynomial time.

1. Introduction

An increasing number of companies now adopt make-to-order business models in which products are custom-made and delivered to customers within a very short lead time directly from the factory. Consequently, there is little or no finished product inventory in the supply chain such that production and outbound distribution are intimately linked and must be scheduled jointly. A majority of the companies worldwide rely on the 3PL providers for their daily distribution and other logistics needs [1]. 3PL providers often follow a fixed daily or weekly schedule for serving their customers. For example, many package delivery service providers such as UPS and FedEx have daily fixed package pickup times; and most 3PL rail, ocean, and air freight service providers have a fixed weekly schedule for a specific origin-destination pair.

In this paper, we study integrated production and outbound distribution scheduling decisions commonly faced by many manufacturers that operate in a make-to-order mode and rely on a 3PL provider for finished product delivery to customers where the 3PL provider follows a fixed delivery schedule. Examples of such manufacturers include most high-end custom-made consumer electronics product manufacturers based in Asia that rely on air flights (which have fixed departure times) to deliver finished products to the US and European markets. The production and distribution scheduling problem faced by such a manufacturer can be described as follows. At the beginning of a planning horizon, the manufacturer has received a set $J = \{J_1, J_2, \dots, J_n\}$ of n independent orders from its customers to be processed on a single assembly line. Order J_i has a processing time p_i, a weight w_i and a desired due date d_i which is negotiated and agreed on by the manufacturer and the customer who placed the order. Finished orders are delivered by vehicles which have fixed departure times. In the planning horizon, there are z possible vehicle departure time instants $T_1, T_2, \dots, T_z$, whereby at time T_j, $1 \le j \le z$, there are v_j vehicles available for delivery. In the air flight case, each vehicle represents an air freight container. Based on a contractual agreement between the manufacturer and the 3PL provider, the manufacturer can use a certain number (e.g., v_j) of containers available on a given flight with departure time T_j. Usually the 3PL provider charges the manufacturer a fixed transportation cost for each air freight container used. Thus, the total transportation cost is represented by the total number of vehicles (i.e., total number of containers) used. Each order is packaged into a standard-size pallet for delivery convenience regardless of the order size. Each vehicle can deliver at most C orders (e.g., in the air flight case, each container can hold up to C pallets). The v_j vehicles can only deliver orders that are completed by time T_j. A feasible schedule is one in which each order has completed processing and delivered by one of the available vehicles. Without loss of generality, we may assume that $T_z \geq \sum_{i=1}^{n} p_i$; otherwise, there is at least one order that cannot be delivered and hence there is no feasible schedule.

In a given feasible schedule, if order J_i is delivered at time T_j and $T_j > d_i$, we define U_i to be 1; if order J_i is delivered at time T_j and $T_j \leq d_i$, we define U_i to be 0. We say in a given feasible schedule an order J_i is *early* if $U_i = 0$ and *late* if $U_i = 1$. The minimum total weight of late orders $\sum_{i=1}^{n} w_i U_i$ measures the delivery timeliness relative to the customers desired due dates and is one of the most commonly used measurements in practice. The problem of minimizing $\sum_{i=1}^{n} w_i U_i$ is NP-hard as it contains the NP-hard classical single-machine total weight of late orders scheduling problem [2] as a special case when the delivery part is not considered. In this paper, we consider the special case of the problem in which processing times and weights are oppositely ordered; that is, if $w_i < w_j$, then $p_i \geq p_j$ for all $J_i, J_j \in J$. That processing times and weights are oppositely ordered is the case in many practical applications. For example, when production resources are comparative shortage, one would prefer the order with shorter processing time. Our problem is to find a feasible schedule that minimizes one of the following objective functions: (1) $\sum_{i=1}^{n} w_i U_i$ and (2) the number of vehicles used subject to the condition that $\sum_{i=1}^{n} w_i U_i$ is minimum. We show that all two problems are solvable in polynomial time.

Research on integrated production and outbound distribution scheduling problems is relatively recent but has attracted a rapidly growing interest in the last several years [3]. In most of the problems considered in the literature, vehicle departure times are not fixed and need to be determined along with other decisions. Only a handful of problems considered in the literature involve fixed vehicle departure times. Such problems can be classified into two types based on vehicle availability. One type assumes that there are infinite number of vehicles available at each departure time, whereas the other type assumes that there are a limited number of vehicles available at each departure time. Stecke and Zhao [4], Melo and Wolsey [5], and Zhong et al. [6] all consider similar problems with an infinite number of vehicles where each order has a deadline which has to be satisfied and the objective is to minimize the total transportation cost. Their problems differ slightly in the structure of the transportation cost. Since the focus of this paper is on problems with a finite number of vehicles, we do not review these papers in detail.

Li et al. [7–9] and Zandieh and Molla-Alizadeh-Zavardehi [10] study several similar problems with a finite number of vehicles at each departure time which are all motivated by applications involving synchronizing assembly operations of consumer electronics products such as PCs and air transportation schedules. Orders may have different sizes and the capacity of a vehicle is measured by the total size (weight or volume) of orders that it can carry. There is an earliness or tardiness penalty if an order is delivered earlier or later than the due date. The objective is to minimize the total transportation cost and total weighted earliness and tardiness penalty. Li et al. [9] consider the case where all the orders are processed on a number of parallel production lines, whereas the other papers consider the case with a single production line. The problems are strongly NP-hard as they contain the strongly NP-hard classical single-machine total weighted tardiness scheduling problem [11] as a special case when the delivery part is not considered. These papers propose various heuristics for solving their problems. Wang et al. [12] study a problem with a finite number of vehicles which involves coordinating mail processing and distribution schedules at a mail processing and distribution center. The objective is to minimize the total unused vehicle capacity. The authors show that this problem is strongly NP-hard and propose dispatching rules and heuristics.

Fu et al. [13] consider a problem where there is a limit on the total delivery capacity at each departure time. Each order has a delivery departure deadline, a production window, a size, and a profit. The problem is to select a subset of orders to accept so as to maximize the total profit of the accepted orders under the constraint that each accepted order is processed within its production window, the delivery of this order is departed by its delivery departure deadline, and the total size of the orders delivered at each departure time does not exceed the available vehicle capacity limit. The problem is strongly NP-hard as it contains the bin packing problem as its special case when only the delivery part is considered. The authors propose a polynomial time approximation scheme for the problem.

Leung and Chen [14] discuss an integrated production and distribution scheduling problem. In the beginning of a planning horizon, the manufacturer has received a set of orders to be processed on a single production line. Completed orders are delivered to customers by a finite number of vehicles provided by the 3PL company which follows a fixed daily or weekly shipping schedule such that the vehicles have fixed departure dates. The problem is to find a feasible schedule that minimizes one of the following objective functions: (a) the maximum lateness of orders, (b) the number of vehicles used subject to the condition that the maximum lateness is minimum, and (c) the weighted sum of the maximum lateness and the number of vehicles used. They show that all three problems are solvable in polynomial time.

The remainder of this paper is organized as follows. In Section 2, we give a simple algorithm to check the feasibility of a given instance of the problem. In Sections 3 and 4, we give polynomial time algorithms to solve problems (1) and (2), respectively. We conclude the paper in Section 5.

2. Feasibility

Given an instance of the problem, we first need to determine whether there is any feasible schedule. The following Algorithm FD come from Leung and Chen [14] and are stated. The idea is to schedule the orders in Smallest-Processing-Time first (SPT) order. Let S be a SPT schedule. Let $S(T_i, T_j)$ denote the set of orders completed in the interval $(T_i, T_j]$ in S. Let T_{z+1} be any integer greater than T_z. We then assign the orders to the vehicles by the following algorithm.

Algorithm FD

Input. An SPT schedule S. For each departure time T_j, for $1 \leq j \leq z$, there are v_j vehicles available for delivery at T_j.

Output. “Yes” if it is possible to deliver all the orders in the schedule S, “No” otherwise.

Method

```
(1) $S(T_z, T_{z+1}) := \emptyset$; $T_0 := 0$.
(2) For $j = 1$ to $z$ do the following:
    (a) assign the orders in $S(T_{j-1}, T_j)$ to one of the $v_j$ vehicles available at time $T_j$;
        after an order is assigned to a vehicle, it is removed from $S(T_{j-1}, T_j)$;
    (b) if all of the $v_j$ vehicles are full and there is still at least one unassigned order in $S(T_{j-1}, T_j)$, then put all the unassigned orders into $S(T_j, T_{j+1})$;
(3) if $S(T_z, T_{z+1}) = \emptyset$ then return “Yes,” else return “No.”
```

If the algorithm returns “Yes,” then there is a feasible schedule; otherwise, there is no feasible schedule. The overall running time of the algorithm is $O(n \log n)$ time.

Let (S_1, S_2) be a SPT schedule of the instance of the problem. Clearly, there is no feasible schedule for the instance of the problem if $|S_2| = \sum_{j=k}^{z} v_j C$ and $\sum_{J_i \in S_1} p_i > T_{k-1}$, where $1 \leq k \leq z$. In the remainder of this paper, we will assume that there is a feasible schedule for the given instance.

3. Total Weight of Late Orders

In this section we give a polynomial-time algorithm to solve the total weight of late orders problem. For each $1 \leq i \leq n$, let $\overline{d_i} = \max\{T_m \mid T_m \leq d_i, 1 \leq m \leq z\}$ and for each $1 \leq j \leq z$, let $N_{0,j} = \{J_i \mid \overline{d_i} = T_j, 1 \leq i \leq n\}$. Clearly, an order $J_i \in N_{0,j}$ is early if and only if it is completed and delivered by time T_j for a given schedule. The following algorithm decides whether there is a feasible schedule that minimizes the total weight of late orders. To break ties when sequencing the orders in decreasing order of their weights, we employ the last-in first rule; that is, we arrange order J_j before order J_i if J_j is merged into a set of orders S and $w_j = w_i$, where $J_i \in S$.

Algorithm WF

Input. J and $N_{0,1}, \ldots, N_{0,z}$.

Output. A feasible schedule that minimizes the total weight of late orders.

Method

```
(1) $P = \sum_{i=1}^{n} p_i$; $t := 0$. $T_0 := 0$. $N_{0,0} := \emptyset$.
(2) For $j = z$ down to 1 do the following.
    (2.1) Let $R_{t,j}$ be all the orders in $N_{t,j}$, arranged in decreasing order of their weights.
    (2.2) If $v_j C < |R_{t,j}|$, then update $N_{t,j-1} := N_{t,j-1} \cup F_{t,j}$ and $R_{t,j} := R_{t,j} \setminus F_{t,j}$, where $F_{t,j}$ are the first $|R_{t,j}| - v_j C$ orders from $R_{t,j}$.
    (2.3) $p := \sum_{J_i \in R_{t,j}} p_i$.
    (2.4) Schedule the orders in $R_{t,j}$ from time $P - p$ to $P$. These orders will be delivered by the vehicles available at time $T_j$.
    (2.5) $P := P - p$.
    (2.6) If $P > T_{j-1}$, find $J_{l_t} \in N_{t,j_l}$ such that $w_{l_t} = \min\{w_i \mid J_i \in \bigcup_{i=0}^{j-1} N_{t,i}\}$, where $0 \leq j_l \leq j - 1$. Update $N_{t,0} := N_{t,0}, N_{t,1} := N_{t,1}, \ldots, N_{t,j_l-1} := N_{t,j_l-1}, N_{t,j_l} := N_{t,j_l} \setminus \{J_{l_t}\}, N_{t,j_l+1} := N_{t,j_l+1}, \ldots, N_{t,j-1} := N_{t,j-1}, N_{t,j} := R_{t,j}, \ldots, N_{t,z-1} := R_{t,z-1}, N_{t,z} := R_{t,z} \cup \{J_{l_t}\}$, and $t := t + 1$; return to (2).
(3) Stop. The schedule $(R_{t,1}, \ldots, R_{t,z})$ is an optimal feasible schedule, where the orders in $R_{t,j}$ will be delivered by the vehicles available at time $T_j$ and $\sum_{i=0}^{t-1} w_{l_i} = \min \sum w_j U_j$.
```

Algorithm WF consists of $t + 1$ iterations. For ease of presentation, we present the detailed output data obtained by the $t + 1$ iterations as follows:

$$\begin{aligned}&\left(R_{0,j_0}, \ldots, R_{0,z}\right), J_{l_0}, w_{l_0}; \ldots; \left(R_{t-1,j_{t-1}}, \ldots, R_{t-1,z}\right),\\ &J_{l_{t-1}}, w_{l_{t-1}}; \left(R_{t,1}, \ldots, R_{t,z}\right),\end{aligned} \tag{1}$$

where for $k = 0, 1, \ldots, t - 1$, $1 \leq j_k \leq z$, $P - \sum_{J_i \in \bigcup_{j=j_k}^{z} R_{k,j}} p_i > T_{j_k - 1}$, $J_{l_k} \in J \setminus \bigcup_{j=j_k}^{z} R_{k,j}$, and $w_{l_k} = \min\{w_i \mid J_i \in J \setminus \bigcup_{j=j_k}^{z} R_{k,j}\}$.

The following lemma describes some properties of the data obtained by Algorithm WF.

Lemma 1. *Let the data in (1) be obtained by Algorithm WF. All of the following hold:*

(i) $j_0 \geq j_1 \geq \cdots \geq j_t = 1$,

(ii) $w_{l_0} \leq w_{l_1} \leq \cdots \leq w_{l_{t-1}}$,

(iii) *for each* $0 \leq k \leq t - 1$ *and* $k + 1 \leq i \leq t$, $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{i,j}$ *and is a late order if the orders in* $R_{i,j}$ *are delivered at time* T_j.

Proof. Let $(R_{0,j_0}, \ldots, R_{0,z})$, J_{l_0}, and w_{l_0} be the output data obtained by the first iteration of Algorithm WF. Then, we have that $P - \sum_{J_i \in \bigcup_{j=m}^{z} R_{0,j}} p_i \leq T_{m-1}$, $|R_{0,m}| \leq v_m C$ for $m = z, z - 1, \ldots, j_0 + 1$, and $P - \sum_{J_i \in \bigcup_{j=j_0}^{z} R_{0,j}} p_i > T_{j_0 - 1}$, $|R_{0,j_0}| \leq v_{j_0} C$. We now run the second iteration on the data

$(N_{1,0}, N_{1,1}, \ldots, N_{1,z})$. It follows from Steps (2.1) and (2.2) that $R_{1,z} = R_{0,z} \cup \{J_{l_0}\}$ if $|R_{0,z}| < v_z C$ and $R_{1,z} = R_{0,z} \cup \{J_{l_0}\} \setminus \{J_l\}$ otherwise, where $w_l = \max\{w_i \mid J_i \in R_{0,z} \cup \{J_{l_0}\}\}$. Since order processing times and order weights are oppositely ordered, we have $p_l = \min\{p_i \mid J_i \in R_{0,z} \cup \{J_{l_0}\}\}$. In either case, $\sum_{J_i \in R_{1,z}} p_i \geq \sum_{J_i \in R_{0,z}} p_i$. This implies that $P - \sum_{J_i \in R_{1,z}} p_i \leq P - \sum_{J_i \in R_{0,z}} p_i \leq T_{z-1}$. Further, we have that for $m = z-1, \ldots, j_0$,

$$\bigcup_{j=m}^{z} R_{1,j} = \bigcup_{j=m}^{z} R_{0,j} \cup \left\{J_{l_0}\right\} \quad \text{if } \sum_{j=m}^{z} \left|R_{0,j}\right| < \sum_{j=m}^{z} v_j C, \tag{2}$$

$$\bigcup_{j=m}^{z} R_{1,j} = \bigcup_{j=m}^{z} R_{0,j} \cup \left\{J_{l_0}\right\} \setminus \{J_l\} \quad \text{if } \sum_{j=m}^{z} \left|R_{0,j}\right| = \sum_{j=m}^{z} v_j C, \tag{3}$$

where $p_l = \min\{p_i \mid J_i \in \bigcup_{j=m}^{z} R_{0,j} \cup \{J_{l_0}\}\}$. This implies that for $m = z-1, \ldots, j_0+1$, $P - \sum_{J_i \in \bigcup_{j=m}^{z} R_{1,j}} p_i \leq P - \sum_{J_i \in \bigcup_{j=m}^{z} R_{0,j}} p_i \leq T_{m-1}$. Thus, $j_0 \geq j_1$ holds. The proof of the following inequalities in (i) is similar to that of the first inequality. We conclude that (i) holds.

By Algorithm WF, we have that $J_{l_0} \in J \setminus \bigcup_{j=j_0}^{z} R_{0,j}$ and $J_{l_1} \in J \setminus \bigcup_{j=j_1}^{z} R_{1,j}$, where $w_{l_0} = \min\{w_i \mid J_i \in J \setminus \bigcup_{j=j_0}^{z} R_{0,j}\}$ and $w_{l_1} = \min\{w_i \mid J_i \in J \setminus \bigcup_{j=j_1}^{z} R_{1,j}\}$. Due to (i), $j_0 \geq j_1$. By (2), $\bigcup_{j=j_0}^{z} R_{1,j} = \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\}$ if $\sum_{j=j_0}^{z} |R_{0,j}| < \sum_{j=j_0}^{z} v_j C$. In the case, since $J \setminus \bigcup_{j=j_0}^{z} R_{0,j} \supseteq J \setminus \bigcup_{j=j_1}^{z} R_{1,j}$, we have that $w_{l_0} \leq w_{l_1}$. By (3), $\bigcup_{j=j_0}^{z} R_{1,j} = \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\} \setminus \{J_l\}$ if $\sum_{j=j_0}^{z} |R_{0,j}| = \sum_{j=j_0}^{z} v_j C$, where $w_l = \max\{w_i \mid J_i \in \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\}\}$. In the case, since $J \setminus \bigcup_{j=j_0}^{z} R_{0,j} \supseteq \{J_l\} \cup J \setminus \bigcup_{j=j_1}^{z} R_{1,j}$, along with $w_l \geq w_{l_0}$, we have that $w_{l_0} \leq w_{l_1}$. Thus, the first inequality $w_{l_0} \leq w_{l_1}$ in (ii) holds. The proof of the following inequalities in (ii) is similar to that of the first inequality. We conclude that (ii) holds.

For any $k \in \{0, 1, \ldots, t-1\}$, suppose $J_{l_k} \in N_{0,k_0}$. Clearly, J_{l_k} is a late order if it is delivered at time T_j, where $k_0 < j \leq z$. To show (iii), there are two cases to consider: (a) $k_0 < j_k$ and (b) $k_0 \geq j_k$.

(a) Consider $k_0 < j_k$. Assume that $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{k+1,j}$ does not hold. It follows from the argument of (i) that $\sum_{j=j_k}^{z} |R_{k,j}| = \sum_{j=j_k}^{z} v_j C$, $\bigcup_{j=j_k}^{z} R_{k+1,j} = \bigcup_{j=j_k}^{z} R_{k,j}$, and $w_{l_k} = \max\{w_i \mid J_i \in \bigcup_{j=j_k}^{z} R_{k,j} \cup \{J_{l_k}\}\}$; that is, $p_{l_k} = \min\{p_i \mid J_i \in \bigcup_{j=j_k}^{z} R_{k,j} \cup \{J_{l_k}\}\}$. This, along with $w_{l_k} = \min\{w_i \mid J_i \in J \setminus \bigcup_{j=j_k}^{z} R_{k,j}\}$, that is, $p_{l_k} = \max\{p_i \mid J_i \in J \setminus \bigcup_{j=j_k}^{z} R_{k,j}\}$, implies that, for any $J_{i_1} \in J \setminus \bigcup_{j=j_k}^{z} R_{k,j}$ and $J_{i_2} \in \bigcup_{j=j_k}^{z} R_{k,j}$, $p_{i_1} \leq p_{i_2}$. Thus, we have that there is a SPT schedule (S_1, S_2) such that $|S_2| = \sum_{j=j_k}^{z} v_j C$ and $\sum_{J_i \in S_1} p_i > T_{j_k - 1}$, where $S_1 = J \setminus \bigcup_{j=j_k}^{z} R_{k,j}$ and $S_2 = \bigcup_{j=j_k}^{z} R_{k,j}$. This implies that there is no feasible schedule for the instance of the problem, which contradicts the assumption that there is a feasible schedule for the given instance. Thus, $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{k+1,j}$ and is a late order if the orders in $R_{k+1,j}$ are delivered at time T_j. Further, it follows from the argument of (i) that $\bigcup_{j=j_k}^{z} R_{k+2,j} = \bigcup_{j=j_k}^{z} R_{k+1,j} \cup \{J_{l_{k+1}}\}$ if $\sum_{j=j_k}^{z} |R_{k+1,j}| < \sum_{j=j_k}^{z} v_j C$ and $\bigcup_{j=j_k}^{z} R_{k+2,j} = \bigcup_{j=j_k}^{z} R_{k+1,j} \cup \{J_{l_{k+1}}\} \setminus \{J_l\}$ if $\sum_{j=j_k}^{z} |R_{k+1,j}| = \sum_{j=j_k}^{z} v_j C$, where $w_l = \max\{w_i \mid J_i \in \bigcup_{j=j_k}^{z} R_{k+1,j} \cup \{J_{l_{k+1}}\}\}$. This, along with $w_{l_k} \leq w_{l_{k+1}}$ and J_{l_k} merged into $\bigcup_{j=j_k}^{z} R_{k+1,j}$ before $J_{l_{k+1}}$, implies that $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{k+2,j}$ and is a late order if the orders in $R_{k+2,j}$ are delivered at time T_j. Similarly, we can show $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{i,j}$ and is a late order if the orders in $R_{i,j}$ are delivered at time T_j for $i = k+3, \ldots, t$.

(b) Consider $k_0 \geq j_k$. Note the fact that order J_{l_k} is pushed by the iterations from N_{0,k_0} into $J \setminus \bigcup_{j=j_k}^{z} R_{k,j}$. By Steps (2.1) and (2.2), we have that $|R_{k,j}| = v_j C$ for $j = j_k, \ldots, k_0$ and $w_{l_k} = \max\{w_i \mid J_i \in \bigcup_{j=j_k}^{k_0} R_{k,j}\}$. This implies $J_{l_k} \in \bigcup_{j=k_0+1}^{z} R_{k+1,j}$ ($\subseteq \bigcup_{j=j_k}^{z} R_{k+1,j}$). Otherwise, it follows from the argument of (i) that $\sum_{j=k_0+1}^{z} |R_{k,j}| = \sum_{j=k_0+1}^{z} v_j C$, $\bigcup_{j=k_0+1}^{z} R_{k+1,j} = \bigcup_{j=k_0+1}^{z} R_{k,j}$, and $w_{l_k} = \max\{w_i \mid J_i \in \bigcup_{j=k_0+1}^{z} R_{k,j} \cup \{J_{l_k}\}\}$, and then we have that $\sum_{j=j_k}^{z} |R_{k,j}| = \sum_{j=j_k}^{z} v_j C$, $\bigcup_{j=j_k}^{z} R_{k+1,j} = \bigcup_{j=j_k}^{z} R_{k,j}$, and $w_{l_k} = \max\{w_i \mid J_i \in \bigcup_{j=j_k}^{z} R_{k,j} \cup \{J_{l_k}\}\}$. Similar to the argument of case (a), we can derive a contradiction. Similarly, we can show $J_{l_k} \in \bigcup_{j=j_k}^{z} R_{i,j}$ and is a late order if the orders in $R_{i,j}$ are delivered at time T_j for $i = k+2, \ldots, t$. This ends the proof for (iii). □

Theorem 2. *Algorithm WF correctly finds a feasible schedule that minimizes the total weight of late orders in $O(zn^2 \log n)$ time.*

Proof. We first prove that $\bigcup_{j=j_k}^{z} R_{k,j}$ has not only exactly k late orders $\{J_{l_0}, \ldots, J_{l_{k-1}}\}$ but also the maximum total processing time and the minimum total weight $\sum_{i=0}^{k-1} w_{l_i}$ of late orders among all schedules for $k = 0, 1, \ldots, t-1$, where $\{J_{l_0}, \ldots, J_{l_{k-1}}\} = \emptyset$ and $\sum_{i=0}^{k-1} w_{l_i} = 0$ if $k = 0$. By the definition of $N_{0,z}$ and Steps (2.1) and (2.2) of Algorithm WF, we have that $R_{0,z}$ is a set of early orders delivered at time T_z with the maximum total processing time among all schedules. Similarly, we have that $\bigcup_{j=m}^{z} R_{0,j}$ is a set of early orders with the maximum total processing time among all schedules for $m = z-1, \ldots, j_0$.

By (2) and (3), we have that $\bigcup_{j=j_0}^{z} R_{1,j} = \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\}$ if $\sum_{j=j_0}^{z} |R_{0,j}| < \sum_{j=j_0}^{z} v_j C$ and $\bigcup_{j=j_0}^{z} R_{1,j} = \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\} \setminus \{J_l\}$ if $\sum_{j=j_0}^{z} |R_{0,j}| = \sum_{j=j_0}^{z} v_j C$ where $w_l = \max\{w_i \mid J_i \in \bigcup_{j=j_0}^{z} R_{0,j} \cup \{J_{l_0}\}\}$. This, along with (iii) of Lemma 1, $w_{l_0} = \min\{w_i \mid J_i \in J \setminus \bigcup_{j=j_0}^{z} R_{0,j}\}$ and $\bigcup_{j=j_0}^{z} R_{0,j}$ being a set of early orders with the maximum total processing time among all schedules, implies that $\bigcup_{j=j_0}^{z} R_{1,j}$ has not only exactly a late order J_{l_0} but also the maximum total processing time and the minimum weight w_{l_0} among all schedules. Further, by Steps (2.1) and (2.2) of Algorithm WF, we have that $\bigcup_{j=m}^{z} R_{1,j}$ has not only exactly a late order J_{l_0} but also the maximum total processing time and the minimum weight w_{l_0} among all schedules for $m = j_0 - 1, \ldots, j_1$. Similarly, we can show the result for $k = 2, \ldots, t-1$.

We below prove that $(R_{t,1}, \ldots, R_{t,z})$ is a feasible schedule minimizing the total weight of late orders. By Algorithm WF, the feasibility is obvious. For any feasible schedule $S = (R_1, \ldots, R_z)$, where the orders in R_j are delivered at time T_j for $j = 1, \ldots, z$, we have that $\bigcup_{j=j_0}^{z} R_j$ contains at least a late order. Otherwise, by $\sum_{J_i\in\bigcup_{j=j_0}^{z} R_j} p_i \leq \sum_{J_i\in\bigcup_{j=j_0}^{z} R_{0,j}} p_i$, we have that $\sum_{J_i\in S\setminus\bigcup_{j=j_0}^{z} R_j} p_i \geq P - \sum_{J_i\in\bigcup_{j=j_0}^{z} R_{0,j}} p_i > T_{j_0-1}$. This contradicts the feasibility of S. Given that $\bigcup_{j=j_0}^{z} R_j$ contains at least a late order, $\bigcup_{j=j_1}^{z} R_j$ contains at least two late orders. Otherwise, $\bigcup_{j=j_1}^{z} R_j$ contains a late order. By $\sum_{J_i\in\bigcup_{j=j_1}^{z} R_j} p_i \leq \sum_{J_i\in\bigcup_{j=j_1}^{z} R_{1,j}} p_i$, we have that $\sum_{J_i\in S\setminus\bigcup_{j=j_1}^{z} R_j} p_i \geq P - \sum_{J_i\in\bigcup_{j=j_1}^{z} R_{1,j}} p_i > T_{j_1-1}$. This contradicts the feasibility of S. Similarly, we can show that $\bigcup_{j=j_k}^{z} R_j$ contains $k+1$ late orders at least for $k = 2,\ldots,t-1$. Thus, $(R_{t,1},\ldots,R_{t,z})$ is a feasible schedule minimizing the number of late orders. This, along with $\sum_{i=0}^{t-1} w_{l_i}$ being the minimum total weight, implies that Algorithm WF correctly finds a feasible schedule that minimizes the total weight of late orders.

We now show that the algorithm can be implemented to run in $O(zn^2 \log n)$ time. The algorithm consists of $n+1$ iterations at most since there are n late orders at most. To calculate the complexity of the algorithm we first calculate in a preprocessing step the sets $(\overline{d_1},\ldots,\overline{d_n})$ and $(N_{0,1},\ldots,N_{0,z})$. The calculation of the set $(\overline{d_1},\ldots,\overline{d_n})$ takes $O(n \log z)$. The calculation of the set $(N_{0,1},\ldots,N_{0,z})$ takes $O(n \log z)$ also. Step (1) takes $O(n)$ time. Step (2) is iterated z times. Within each iteration, the most time-consuming step is Step (2.1), sorting the orders in decreasing order of their weights, which takes $O(n \log n)$ time. Hence Step (2) takes $O(zn \log n)$ time. Thus, the overall running time of the algorithm is $O(zn^2 \log n)$ time. □

4. Number of Vehicles Used

In this section we show that the problem of minimizing the number of vehicles used subject to the constraint that the total weight of late orders is minimum can be solved in polynomial time. We assume that we have found the schedule that minimizes the total weight of late orders using the algorithm given in the previous section. The schedule $(R_{t,1},\ldots,R_{t,z})$ was obtained by Algorithm WF. By the fact that if $|R_{t,j}| < v_j C$, then each order in $\bigcup_{i=1}^{j-1} R_{t,i}$ is an early order and will be a late order if we push the order to be delivered by vehicles at T_j. Thus, either there is no feasible schedule or the total weight of late orders will be increased if we push some order in $R_{t,i}$ to be delivered by vehicles at $T_{i+1},\ldots,T_z$. By the optimality of $\sum_{i=0}^{t-1} w_{l_i}$, we have that the total weight of late orders will not be decreased if we push some order in $R_{t,i}$ to be delivered by vehicles at $T_1,\ldots,T_{i-1}$. The following algorithm finds a solution with a minimum number of vehicles used under the constraint that the total weight of late orders is minimum by pushing some orders to an earlier departure time for delivery.

Algorithm MV

Input. z sets of orders $R_{t,1},\ldots,R_{t,z}$ given by Algorithm WF.

Output. A vehicle assignment: $V_1,\ldots,V_z$, where the orders in V_j will be delivered by the vehicles available at time T_j, so that the number of vehicles used is minimum.

Method

```
(1) $T_0 := 0$. $v_0 := 0$.
(2) For $j = z$ down to 1 do the following.
    (a) Let $V_j'$ be all the orders in $R_{t,j}$, arranged in nondecreasing order of their processing times, and $|V_j'| = kC+r$, where $k$ and $r$ are nonnegative integers, and $0 \leq r < C$ and $F$ is the set of the first $r$ orders from $V_j'$.
    (b) If $\sum_{J_i\in\cup_{m=1}^{j-1} R_{t,m}\cup F} p_i > T_{j-1}$, then update $A := \emptyset$ and $V_j := V_j'$, and the orders in $V_j$ will be delivered by the vehicles at time $T_j$ and proceed to the next $j$.
    (c) Let $A'$ be the first $k_0 C + r$ orders from $V_j'$, where $k_0$ is the maximal nonnegative integer such that $\sum_{J_i\in\cup_{m=1}^{j-1} R_{t,m}\cup A'} p_i \leq T_{j-1}$.
    (d) If $\lceil |R_{t,j-1} \cup A'|/C \rceil \leq v_{j-1} C$, then update $A := A'$, $R_{t,j-1} := R_{t,j-1} \cup A$, and $V_j := V_j' \setminus A$ and the orders in $V_j$ will be delivered by the vehicles available at time $T_j$ and proceed to the next $j$.
    (e) Call Subalgorithm CA to computer $A$. Then update $R_{t,j-1} := R_{t,j-1} \cup A$ and $V_j := V_j' \setminus A$, and the orders in $V_j$ will be delivered by the vehicles at time $T_j$ and proceed to the next $j$.
```

Subalgorithm CA

```
(1) $A' :=$ the first $k_0 C + r$ orders from $V_j$, arranged in nondecreasing order of their processing times.
(2) Sort the orders in $R_{t,j-1} \cup A'$ in ascending order of their processing times.
(3) If $|R_{t,j-1} \cup A'| \leq v_{j-1} C$, then stop and output $A = A'$.
(4) For $h = 1$ to $j-2$, let $R_{t,h}' := R_{t,h}$.
(5) $B :=$ the first $|R_{t,j-1} \cup A'| - v_{j-1} C$ orders from $R_{t,j-1} \cup A'$.
(6) For $h = j-2$ down to 1 do the following.
    (a) Sort the orders in $R_{t,h}' \cup B$ in ascending order of their processing times.
    (b) If $\sum_{J_i\in\cup_{m=1}^{h} R_{t,m}'\cup B} p_i > T_h$,
        (b1) if $|A'| < C$, stop and output $A = \emptyset$;
        (b2) update $A' :=$ the first $|A'| - C$ orders from $A'$ and return (2).
    (c) If $|R_{t,h}' \cup B| \leq v_h C$, then stop and output $A = A'$.
    (d) If $h = 1$, stop and output $A = \emptyset$ if $|A'| < C$ and update $A' :=$ the first $|A'| - C$ orders from $A'$ and return (2) else.
    (e) Update $B :=$ the first $|R_{t,h}' \cup B| - v_h C$ orders from $R_{t,h}' \cup B$ and proceed to the next $h$.
```

Theorem 3. *Algorithm MV finds an optimal solution with the minimum number of vehicles used under the constraint that the total weight of late orders is minimum in $O(z^2 n^2 \log n)$ time.*

Proof. We first point out that the solution by Algorithm MV does not change the optimality of the total weight of late orders, since the solution is found by pushing some orders in $R_{t,2}, \ldots, R_{t,z}$ to an earlier departure time for delivery.

Let V_z and A be the output data obtained by the first iteration of Algorithm MV, where $V_z = V_z' \setminus A$ and A is the set of the first $|A|$ orders from V_z' (the orders of $V_z' = R_{t,z}$ have been arranged in nondecreasing order of their processing times). Step (2)(b) corresponds to the case where $\sum_{J_i \in \cup_{m=1}^{z-1} R_{t,m} \cup F} p_i > T_{z-1}$, where F the set of the first $|F|$ orders from V_z' and $0 < |F| < C$. In the case, It is impossible to deliver all orders in F by the vehicles at $T_1, \ldots, T_{z-1}$. This, along with F consisting of the first $|F|$ orders from V_z', means that the number of vehicles used at T_z cannot decrease by one. Thus, $\lceil |V_z'|/C \rceil$ is the minimal number of vehicles used at T_z. However, we cannot push part of orders in F to be delivered by vehicles at $T_{z-1}, \ldots, T_1$, since, if we do that, it not only does not decrease the number of vehicles used at T_z but also increases the amount of orders delivered by vehicles at $T_{z-1}, \ldots, T_1$. In the case, the data $V_z = V_z'$ and $A = \emptyset$ are optimal. Under the case of $\sum_{J_i \in \cup_{m=1}^{z-1} R_{t,m} \cup F} p_i \leq T_{z-1}$, Step (2)(c) computes the set of orders A' which made $\lceil |V_z'|/C \rceil - \lceil |A'|/C \rceil$ the minimal number possibly of vehicles used at T_z, where A' are the first $k_0 C + r$ orders from V_z'. This is because k_0 is the maximal nonnegative integer such that $\sum_{J_i \in \cup_{m=1}^{z-1} R_{t,m} \cup A'} p_i \leq T_{z-1}$ and A' consists of some small orders in V_z'. Step (2)(d) corresponds to the case where $\lceil |R_{t,z-1} \cup A'|/C \rceil \leq v_{z-1} C$. This means that we can push all of orders in A' to be delivered by vehicles at T_{z-1}. Thus, in the case, the output data $A = A'$ and $V_z = V_z' \setminus A$ make $\lceil |V_z|/C \rceil$ the minimal number of vehicles used at T_z. Under the case of $\lceil |R_{t,z-1} \cup A'|/C \rceil > v_{z-1} C$, Step (2)(e) calls Subalgorithm CA to decide a set of orders A such that $\lceil |V_z|/C \rceil$ is the minimal number of vehicles used at T_z, where $V_z = V_z' \setminus A$. We show below that Subalgorithm CA can really do that.

Subalgorithm CA consists of $k_0 + 1$ main iterations for the first $k_0 C + r$ orders, the first $(k_0 - 1)C + r$ orders, …, and the first r orders from V_z', respectively. We now run the first main iteration for the first $k_0 C + r$ orders A' from V_z'. Due to the corresponding case by Step (2)(e), we have that $A' \neq \emptyset$ and $|R_{t,z-1} \cup A'| > v_{z-1} C$. We need to proceed through Step (4). Step (4) assigns the orders in $R_{t,h}$ to a temporary set $R_{t,h}'$ for each h from 1 to $j-2$. This is necessary since Subalgorithm CA operates on $R_{t,h}'$ without changing the content of $R_{t,h}$. Step (5) assigns the first $|R_{t,z-1} \cup A'| - v_{z-1} C$ orders from $R_{t,z-1} \cup A'$ to a set B. If we want to push the orders in A' to be delivered by vehicles at $T_{z-1}, \ldots, T_1$, all orders of B are the minimal increment undertaken by vehicles at $T_{z-2}, \ldots, T_1$ to deliver. Now we proceed through Step (6).

Step (6) consists of $z - 2$ secondary iterations. We now run the first secondary iteration. If $\sum_{J_i \in \cup_{m=1}^{z-2} R_{t,m}' \cup B} p_i > T_{z-2}$, it is impossible to deliver all orders in B by the vehicles at $T_1, \ldots, T_{z-2}$. This means that the number of vehicles used at T_z should to be at least $\lceil |V_z'|/C \rceil - \lceil |A'|/C \rceil + 1$. Step (6)(b1) corresponds to the case where $|A'| < C$ and outputs $A = \emptyset$. This, along with $A' \neq \emptyset$ and $V_z = V_z'$, implies that $\lceil |V_z|/C \rceil$ is the minimal number of vehicles used at T_z. Step (6)(b2) corresponds to the case where $|A'| \geq C$ and starts the second main iteration for the first $(k_0 - 1)C + r$ orders A' from V_z' to check whether it is possible to push the orders in A' to be delivered by vehicles at $T_{z-1}, \ldots, T_1$. On the other hand, $\sum_{J_i \in \cup_{m=1}^{z-2} R_{t,m}' \cup B} p_i \leq T_{z-2}$. Step (6)(c) corresponds to the case where $|R_{t,z-2}' \cup B| \leq v_{z-2} C$. This means that we can push all of orders in A' to be delivered by vehicles at T_{z-1} and T_{z-2}. Thus, the output data $A = A'$ by Step (6)(c) and $V_z = V_z' \setminus A$ make $\lceil |V_z|/C \rceil$ the minimal number of vehicles used at T_z. Otherwise, Step (6)(e) starts the second secondary iteration to check whether it is possible to push the orders in a set of the first $|R_{t,z-2}' \cup B| - v_{z-2} C$ orders from $R_{t,z-2}' \cup B$ to be delivered by vehicles at $T_{z-3}, \ldots, T_1$. Similarly, we can show the result for the following secondary iterations. If needed, we proceed through the last secondary iteration. Suppose that B is a set of orders output by the $(z - 3)$th secondary iteration. In the case of $\sum_{J_i \in R_{t,1}' \cup B} p_i > T_1$, Step (6)(b1) outputs $A = \emptyset$ if $|A'| < C$ and Step (6)(b2) starts the second main iteration for the first $(k_0 - 1)C + r$ orders A' from V_z' else. In the case of $\sum_{J_i \in R_{t,1}' \cup B} p_i \leq T_1$, Step (6)(c) outputs data $A = A'$ if $|R_{t,1}' \cup B| \leq v_1 C$. Otherwise, Step (6)(d) outputs $A = \emptyset$ if $|A'| < C$ and starts the second main iteration for the first $(k_0 - 1)C + r$ orders A' from V_z' else, since it is impossible to deliver all orders in the set of the first $|R_{t,1}' \cup B| - v_1 C$ orders from $R_{t,1}' \cup B$ by the vehicles at $T_0 = 0$. Similarly, we can show the result for the following main iterations in Subalgorithm CA, which can really decide a set orders A such that $\lceil |V_z|/C \rceil$ is the minimal number of vehicles used at T_z, where $V_z = V_z' \setminus A$. For the following iterations in Algorithm MV, we can show the results similarly. At last, the algorithm must be able to find an optimal solution with the minimum number of vehicles used.

We now look at the time complexity of Algorithm MV. In the algorithm, Step (1) takes constant time. Step (2) is iterated z times. Inside the iteration loop, the most time-consuming steps are (2)(a) and (2)(e). Step (2)(a) calls for sorting the orders which takes $O(n \log n)$ time. Step (2)(e) calls Subalgorithm CA which takes $O(zn^2 \log n)$ time. Thus, the overall time complexity of Algorithm MV is $O(z^2 n^2 \log n)$. □

5. Conclusion

In this paper, we have given polynomial-time algorithms for minimizing: (1) the total weight of late orders and (2) the number of vehicles used subject to the condition that the total weight of late orders is minimum. An interesting open question is whether the problem related to release dates is NP-hard or not.

Conflict of Interests

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

Acknowledgment

This work was supported in part by the Zhejiang Natural Science Foundation of China Grant Y6110054.

References

[1] C. J. Langley, E. van Dort, and S. R. Sykes, *2005 Third-Party Logistics: Results and Findings of the 10th Annual Study*, Georgia Institute of Technology, 2006.

[2] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds., pp. 85–104, Plenum Press, New York, NY, USA, 1972.

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

[4] K. E. Stecke and X. Zhao, "Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode," *Manufacturing and Service Operations Management*, vol. 9, no. 2, pp. 206–224, 2007.

[5] R. A. Melo and L. A. Wolsey, "Optimizing production and transportation in a commit-to-delivery business mode," *European Journal of Operational Research*, vol. 203, no. 3, pp. 614–618, 2010.

[6] W. Zhong, Z.-L. Chen, and M. Chen, "Integrated production and distribution scheduling with committed delivery dates," *Operations Research Letters*, vol. 38, no. 2, pp. 133–138, 2010.

[7] K. P. Li, V. K. Ganesan, and A. I. Sivakumar, "Synchronized scheduling of assembly and multi-destination air-transportation in a consumer electronics supply chain," *International Journal of Production Research*, vol. 43, no. 13, pp. 2671–2685, 2005.

[8] K. P. Li, V. K. Ganesan, and A. I. Sivakumar, "Scheduling of single stage assembly with air transportation in a consumer electronic supply chain," *Computers & Industrial Engineering*, vol. 51, no. 2, pp. 264–278, 2006.

[9] K. O. Li, A. I. Sivakumar, and V. K. Ganesan, "Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain," *European Journal of Operational Research*, vol. 187, no. 2, pp. 442–455, 2008.

[10] M. Zandieh and S. Molla-Alizadeh-Zavardehi, "Synchronizing production and air transportation scheduling using mathematical programming models," *Journal of Computational and Applied Mathematics*, vol. 230, no. 2, pp. 546–558, 2009.

[11] J. K. Lenstra, A. H. Rinnooy Kan, and P. Brucker, "Complexity of machine scheduling problems," in *Annals of Discrete Mathematics*, vol. 1, pp. 343–362, North-Holland, Amsterdam, The Netherlands, 1977.

[12] Q. Wang, R. Batta, and R. J. Szczerba, "Sequencing the processing of incoming mail to match an outbound truck delivery schedule," *Computers & Operations Research*, vol. 32, no. 7, pp. 1777–1791, 2005.

[13] B. Fu, Y. Huo, and H. Zhao, "Coordinated scheduling of production and delivery with production window and delivery capacity constraints," *Theoretical Computer Science*, vol. 422, pp. 39–51, 2012.

[14] J. Y. Leung and Z.-L. Chen, "Integrated production and distribution with fixed delivery departure dates," *Operations Research Letters*, vol. 41, no. 3, pp. 290–293, 2013.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization