Machine Scheduling with Deliveries to Multiple Customer Locations*

Chung-Lun Li†
George Vairaktarakis‡
Chung-Yee Lee§

April 2003
Revised: November 2003

Abstract

One important issue in production and logistics management is the coordination of activities between production and delivery. In this paper, we develop a single-machine scheduling model that incorporates routing decisions of a delivery vehicle which serves customers at different locations. The objective is to minimize the sum of job arrival times. The problem is NP-hard in the strong sense in general. We develop a polynomial time algorithm for the case when the number of customers is fixed. More efficient algorithms are developed for several special cases of the problem. In particular, an algorithm is developed for the single-customer case with a complexity lower than the existing ones.

Keywords: Scheduling, dynamic programming, computational complexity

---

* This research was supported in part by the Research Grants Council of Hong Kong under grant no. PolyU6041/01E

†Department of Logistics, Faculty of Business, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. *Email: lgtclli@polyu.edu.hk*

‡Weatherhead School of Management, Department of Operations, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106-7235, U.S.A. *Email: George.Vairaktarakis@case.edu*

§Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. *Email: cylee@ust.hk*

1 Introduction

The coordination of activities along different stages of a supply chain has received a lot of attention in production and operations management research. One particularly important issue in this area of study is the coordination of production and delivery schedules. In many production systems, finished products are delivered from the factory to multiple customer locations, warehouses, or distribution centers by delivery vehicles.

Traditional scheduling models focus on the determination of schedules for production such that some performance measures are optimized without considering the coordination between the machine schedule and the delivery plan. These models implicitly assume that there are infinitely many vehicles available for delivering finished jobs to their destinations so that the finished product can be transported to customers without delay. However, in reality, the number of vehicles is limited and the vehicles may need to deliver to more than one customer location to increase their utilization.

In this paper, we incorporate the delivery plan of a vehicle into a single manufacturing facility model. We consider the case when a single vehicle with infinite or bounded capacity is available to serve different customer locations. The objective is to determine the job processing sequence in the plant together with the delivery schedule so as to minimize the sum of job arrival times.

A few machine scheduling models with transportation components have been studied in the literature. The earliest work is by Maggu and Das [9]. They consider a two-machine flow shop problem with a sufficiently large number of transporters to transfer jobs to the second machine immediately after the jobs are completed on the first machine in which the transportation time is job-dependent. Maggu, Das and Kumar [10] study the same problem with the additional constraint of some jobs being scheduled consecutively. Kise, Shioyama and Ibaraki [6] consider a variant of the problem with only one transporter in which the transporter can carry one job at a time. Other flow shop scheduling problems with transportation of jobs between machines have been studied by Langston [7], Panwalkar [11], Sahni and Vairaktarakis [14], Stern and Vitner [15], Stevens and Germill [16], and Vairaktarakis [17]. All these scheduling models focus on the transportation of jobs between machines in a flow shop environment rather than the shipping of finished goods to customers.

Another line of scheduling research with transportation considerations focuses on the transportation of finished jobs to customers. Hall and Shmoys [5] and Potts [12] consider scheduling models with delivery times in which they have implicitly assumed that there is always a vehicle available for the delivery. Recently, Lee and Chen [8] have incorporated transportation time and vehicle capacity requirements into machine scheduling models. They analyze 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 direction of the delivery. Chang and Lee [2] have extended Lee and Chen's work to the situation when each job occupies a different amount of space in the vehicle. Our work differs from [2] and [8] in that we consider delivery to multiple customers at different locations.

In another related paper, Chen and Vairaktarakis [3] consider the problem of minimizing a convex combination of the total job arrival times and the total transportation cost. In their model, production takes place on parallel identical machines, the number of trucks is unlimited, each truck has limited capacity, and the number of customers is fixed. A similar problem setting is considered by Geismar, Lei and Sriskandarajah [4], 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.

We now describe our problem formally. We consider a given set of n jobs, $\{J_1, J_2, \ldots, J_n\}$, to be processed by a single manufacturing facility (i.e., a plant). Associated with each job J_i is a processing time and its associated customer. There are m customers in total ($m \leq n$) and are located at different locations. There is a delivery vehicle available with capacity K ($K \leq n$) such that at most K jobs can be delivered in each trip. If the vehicle is uncapacitated, then we assume that $K = n$. After the processing of J_i, the job needs to be delivered to its customer by the vehicle. All jobs are available for processing at time 0, and the vehicle is available at the plant at time 0. We would like to determine the sequence in which the jobs should be processed in the plant, the departure times of the vehicle from the plant, and the routing of the vehicle, so as to minimize the sum of job arrival times at the customers. For simplicity, we assume that $J_1, J_2, \ldots, J_n$ are arranged in nondecreasing processing times, i.e., in the shortest processing time first (SPT) order.

The following notation will be used throughout the paper:

n = number of jobs;

$(J_1, J_2, \ldots, J_n)$ = sequence of jobs arranged in SPT order;

p_j = processing time of J_j at the plant;

$P_j = \sum_{i=1}^{j} p_i$ = sum of processing times of jobs $J_1, J_2, \ldots, J_j$;

m = number of customers;

n_k = number of jobs that need to be delivered to customer k, where $\sum_{k=1}^{m} n_k = n$;

$(J_1^{(k)}, J_2^{(k)}, \ldots, J_{n_k}^{(k)})$ = sequence of jobs (arranged in SPT order) that need to be delivered to customer k;

$p_j^{(k)}$ = processing time of $J_j^{(k)}$;

$P_j^{(k)} = \sum_{i=1}^{j} p_i^{(k)}$ = sum of processing times of jobs $J_1^{(k)}, J_2^{(k)}, \ldots, J_j^{(k)}$;

K = capacity of the delivery vehicle ($K \leq n$);

t_{0k} = one-way travel time between the plant and customer k;

$t_{k\ell}$ = travel time between customers k and ℓ;

C_j = completion time of processing of J_j at the plant;

$\hat{C}_j$ = arrival time of job J_j at its customer.

Using the above notation, the objective of our problem is the minimization of $\sum_{i=1}^{n} \hat{C}_i$. We assume that travel times are symmetric and satisfy the triangle inequality (i.e., $t_{jk} = t_{kj}$ and $t_{jk} + t_{k\ell} \geq t_{j\ell}$ for all $j, k, \ell = 0, 1, \ldots, m$).

To facilitate the understanding of our model, consider an example with five jobs, two customers, and an uncapacitated vehicle, where J_1, J_2, J_4 will be delivered to customer 1 and J_3, J_5 will be delivered to customer 2. Thus, $n_1 = 3$ and $n_2 = 2$. The processing times of the jobs are given in Figure 1(a), and the one-way travel times are given in Figure 1(b). Consider the schedule shown in Figure 1(c), where the vehicle will depart from the plant at time 5 to visit customer 1, followed by customer 2, and return to the plant afterward. The vehicle will then depart from the plant again at time 13 to visit customer 1 and return to the plant afterward. Finally, it will depart from the plant at time 19 to visit customer 2. Hence, in this feasible solution, the total job arrival time is $8+8+9+16+22=63$.

Insert Figure 1 about here

In the following sections, we first show that our model is NP-hard in the strong sense. This implies that the existence of a pseudo-polynomial time algorithm for determining the optimal solution of the general problem is highly unlikely. Then, we develop polynomial time algorithms for a number of special cases of the model.

2 Properties of the Model

The following theorem states the computational complexity of our model.

Theorem 1 *The problem is NP-hard in the strong sense.*

Proof: We transform the Traveling Repairman Problem (TRP) into our problem. In the TRP, we are given a set of locations and travel times between any two of them, and we wish to find the route through them that minimizes the sum of delays for reaching each location (see Afrati *et al.* [1]). An instance of the decision problem of TRP can be described mathematically as follows: Given a distance matrix $\{t'_{ij}\}$ between n' locations and a distinguished starting location 0 (where t'_{ij} is a positive integer) and given a threshold L', find a permutation $(\pi(0), \pi(1), \ldots, \pi(n'))$ with $\pi(0) = 0$, such that

$$\sum_{i=1}^{n'-1} \sum_{j=1}^{i} t'_{\pi(j-1),\pi(j)} \leq L'. \tag{1}$$

Sahni and Gonzalez [13] have shown that TRP is NP-hard. In fact, from the construction of Sahni and Gonzalez' proof, it is easy to see that TRP is NP-hard in the strong sense.

Given such an instance of TRP, we construct the following instance of our problem:

$$m = n = K = n';$$

$$p_i = 1/n', \quad (i = 1, 2, \ldots, n);$$

$$t_{ij} = t'_{ij}, \quad (i, j = 0, 1, \ldots, m);$$

$$\text{threshold, } L = L' + n';$$

where J_i belongs to customer i ($i = 1, 2, \ldots, n$).

Clearly, this construction can be achieved in polynomial time. It is easy to show that in this constructed instance, it is optimal to deliver all n jobs in a single shipment and have the vehicle leave the plant at time 1. Indeed, compare this single-shipment schedule with a two-shipment schedule having k jobs ($1 \leq k < n$) delivered in the first shipment. The first k jobs in the two-shipment schedule can get delivered earlier by a total of no more than $k(1 - \frac{k}{n})$ units of time, but the remaining $n - k$ jobs will get delayed by a total of at least $(n-k)(1+\frac{k}{n})$ because all t_{ij}'s are positive integers and hence every round trip takes at least 2 time units. Note that $k(1 - \frac{k}{n}) < (n - k)(1 + \frac{k}{n})$. Hence, in order to attain the minimal total job arrival time, all n jobs must be delivered together in a single trip, with the vehicle leaving the plant at time 1 and following a minimum distance "traveling repairman tour" to deliver the jobs. Therefore, it is easy to see that the TRP has a solution satisfying (1) if and only if the above instance of our problem has a solution with a total job arrival time no greater than L, which includes the total time that the n' jobs spent at the plant (i.e., 1 unit of time for each of the n' jobs) and the total time that the jobs spent on traveling (i.e., L'). This completes the proof of the theorem. $\square$

Note that from the proof of Theorem 1, we know that the problem remains strongly NP-hard if the delivery vehicle is uncapacitated. Next, we present some properties of the optimal solution to the problem.

Theorem 2 *There exists an optimal schedule that satisfies the following properties:*

(i) Jobs are processed in the plant without idle time.

(ii) If J_i is processed earlier than J_j in the plant, then J_i leaves the plant no later than J_j.

(iii) For $k = 1, 2, \ldots, m$, jobs $J_1^{(k)}, J_2^{(k)}, \ldots, J_{n_k}^{(k)}$ are processed in the plant in this order, i.e., in SPT order.

(iv) When the vehicle finishes a delivery and returns to the plant, if there are still jobs that need to be transported, then the vehicle either (a) transports the next batch of jobs immediately or (b) waits and starts the next delivery at a completion time of a job, say J_j, at the plant. In case (b), the vehicle departs at time C_j.

Proof: (i) If there exists idle time in the plant, we can always move the subsequent jobs earlier without increasing the objective value. (ii) Suppose that in an optimal schedule, J_i is processed earlier than J_j in the plant but leaves the plant later than J_j. Then interchanging the positions of these two jobs at the plant while retaining the same delivery schedule will provide us with a new solution with the same total job arrival time. Thus, by repeatedly applying this argument, an optimal schedule that satisfies property (ii) can be obtained. (iii) Consider an optimal schedule where $J_{i+1}^{(k)}$ is processed earlier than $J_i^{(k)}$ in the plant. Interchanging the positions of these two jobs at the plant and their positions in the delivery schedule will provide us with a new solution with the same total job arrival time. An optimal schedule in which jobs $J_1^{(k)}, J_2^{(k)}, \ldots, J_{n_k}^{(k)}$ are processed in SPT order can be obtained by repeatedly applying this argument. (iv) If the vehicle does not immediately transport the next batch of jobs and if it does not start the next delivery at a job completion time, then we can move the delivery time of the next batch of jobs earlier and the objective value will be lower. If the vehicle waits and starts the next delivery at C_j, then clearly, the schedule is not optimal unless the vehicle's waiting is for the purpose of delivering J_j. $\square$

It suffices to restrict our search of solutions that satisfy the properties in the above theorem. In the following sections, we will focus on the development of solution procedures for various cases of the problem. We will consider only schedules that satisfy properties (i)–(iv) of Theorem 2.

3 The Single-Customer Case

In this section, we consider the special case of a single customer. Lee and Chen [8] developed an algorithm for solving this special case, and their algorithm has a running time complexity of $O(n^3)$. We now present a more efficient dynamic programming algorithm for solving this special case. For computational purposes that become evident below, we include a dummy job J_{n+1} with $p_{n+1} = \max\{p_n, n{\cdot}2t_{01}\}$. By Theorem 2(i)&(iii), the jobs are processed consecutively at the plant in SPT order, i.e., $J_1, J_2, \ldots, J_{n+1}$. Define

$f(j) =$ minimum total job arrival time for jobs $J_1, J_2, \ldots, J_j$ when the last shipment departs from the plant at time P_j and delivers J_j (maybe along with some other jobs), for $1 \leq j \leq n+1$.

Consider the delivery schedule of the job subset $\{J_1, J_2, \ldots, J_j\}$ and suppose that there is a vehicle departure at time P_j. There are two possible cases.

Case 1: $P_j < P_1 + 2t_{01}$. In this case, no job departs from the plant before time P_j. Thus, by Theorem 2(ii), jobs $J_1, J_2, \ldots, J_j$ are batched together to depart from the plant at time P_j. This batch of jobs arrives at the customer at time $P_j + t_{01}$. Hence, the contribution of this delivery to the objective value is

$$F_j \equiv j(P_j + t_{01}).$$

Case 2: $P_j \geq P_1 + 2t_{01}$. In any optimal schedule of this case, at least one job must depart from the plant before time P_j. Let

$$i = \max\{i' \mid i' < j \text{ and the vehicle departs from the plant at time } P_{i'} \text{ and delivers } J_{i'}\}.$$

By Theorem 2(iv), the vehicle departs from the plant at time P_i and is followed by a number of consecutive nonstop shipments as depicted in Figure 2. In the last one of these nonstop shipments, the vehicle returns to the plant at time τ, where $\tau \leq P_j$. By Theorem 2(ii), jobs $J_{i+1}, J_{i+2}, \ldots, J_j$ are delivered either by these nonstop shipments or by the shipment that departs from the plant at time P_j. Let S_{ij} be the number of nonstop shipments between time P_i and P_j (see Figure 2), and let $\eta_i(k)$ be the number of jobs delivered in the k-th nonstop shipment after time P_i. Clearly, $S_{ij} \le \lfloor \frac{P_j - P_i}{2t_{01}} \rfloor$. Also, $S_{ij} \le n$, since a shipment must carry at least one job. Then, the contribution of jobs $J_{i+1}, J_{i+2}, \ldots, J_j$ to the objective value is

$$F'_{ij} \equiv \sum_{k=2}^{S_{ij}} \eta_i(k) \cdot \Big[P_i + (2k-1)t_{01}\Big] + \eta_{ij} \cdot (P_j + t_{01}),$$

where η_{ij} is the number of jobs completed by time P_j that are not delivered by these nonstop shipments ($\eta_{ij} = +\infty$ if the number of those jobs is greater than the vehicle capacity K). The contribution of the η_{ij} jobs that are not delivered by the nonstop shipments is captured by the second term in the above expression. The contribution of the nonstop shipments is captured by the first term in which "$P_i + (2k-1)t_{01}$" is the arrival time of the k-th shipment. The contribution of the first nonstop shipment is not included here, since it is already included in $f(i)$.

Insert Figure 2 about here

The above observations establish the recurrence relation of

$$f(j) = \min_{\substack{i>0 \text{ s.t.} \\ P_i \le P_j - 2t_{01}}} \Big\{ f(i) + F'_{ij} \Big\},$$

for $j = 1, 2, \ldots, n{+}1$ such that $P_j \ge P_1 + 2t_{01}$, and the boundary condition of

$$f(j) = F_j,$$

for $j = 1, 2, \ldots, n{+}1$ such that $P_j < P_1 + 2t_{01}$. They provide the optimal schedule for the single-customer case with objective function value

$$f^* = f(n{+}1) - (P_{n+1} + t_{01}).$$

In the calculation of f^*, we exclude the contribution of the dummy job J_{n+1}. This is because its processing time is no less than $n \cdot 2t_{01}$, and hence, there exists an optimal schedule in which it is shipped by itself. The technical reason for introducing J_{n+1} is precisely to make the presentation of f^* easier. Otherwise, one would have to know the time that the last shipment departs from the plant. Assuming that all the values of F'_{ij} have been determined, the computational time required by the above recurrence relation is $O(n^2)$.

To demonstrate how this dynamic program works, consider an example with $m = 1$, $n = 4$, $K = 2$, and $t_{01} = 3$. The processing times of the jobs (including the dummy job J_5) are given in Figure 3(a). The boundary conditions are:

$$f(1) = F_1 = (1)(P_1 + t_{01}) = (1)(1 + 3) = 4;$$
$$f(2) = F_2 = (2)(P_2 + t_{01}) = (2)(3 + 3) = 12;$$
$$f(3) = F_3 = (3)(P_3 + t_{01}) = (3)(5 + 3) = 24.$$

For $j = 4, 5$, the values of S_{ij}, $\eta_i(k)$ $(k = 2, 3, \ldots, S_{ij})$, η_{ij}, and F'_{ij} are shown in Figure 3(b). Thus,

$$\begin{aligned} f(4) &= \min\{f(1){+}F'_{14},\ f(2){+}F'_{24}\} \\ &= \min\{4{+}\infty,\ 12{+}26\} = 38; \\ f(5) &= \min\{f(1){+}F'_{15},\ f(2){+}F'_{25},\ f(3){+}F'_{35},\ f(4){+}F'_{45}\} \\ &= \min\{4{+}73,\ 12{+}67,\ 24{+}51,\ 38{+}37\} = 75; \\ f^* &= f(5) - (P_5 + t_{01}) = 75 - (34 + 3) = 38. \end{aligned}$$

Since $f(5)$ attains its minimum when $i = 3$, an optimal solution is to have the vehicle leave the plant at $P_3 = 5$, followed by some nonstop shipment(s). This optimal schedule is depicted in Figure 3(c). Note that $f(5)$ also attains a minimum when $i = 4$ and $f(4)$ attains its minimum when $i = 2$. Hence, an alternative optimal solution is to have the vehicle leave the plant at $P_2 = 3$, followed by some nonstop shipment(s), and then have the vehicle leave the plant again at $P_4 = 10$. This alternative optimal schedule is shown in Figure 3(d).

Insert Figure 3 about here

We now describe how the values of F'_{ij} are computed efficiently.

Lemma 1 *For each $i = 1, 2, \ldots, n$, the values of S_{ij} $(j = i{+}1, i{+}2, \ldots, n{+}1)$, the values of $\eta_i(k)$ $(k = 2, 3, \ldots, S_{i,n+1})$, and the values of η_{ij} $(j = i{+}1, i{+}2, \ldots, n{+}1)$ can be determined in $O(n)$ time.*

Proof: Consider the consecutive nonstop shipments after the vehicle departs from the plant at time P_i. Define $\Pi(y) = \sum_{j=i+1}^{i+y} p_j$, where $\Pi(0) = 0$. Clearly, $\Pi(1), \Pi(2), \ldots, \Pi(n{-}i{+}1)$ can be predetermined in $O(n)$ time. Let y_ℓ denote the total number of jobs delivered in nonstop shipments $2, 3, \ldots, \ell$. Then, the values of $S_{i,i+1}, S_{i,i+2}, \ldots, S_{i,n+1}$ and $\eta_i(2), \eta_i(3), \ldots, \eta_i(S_{i,n+1})$ can be determined by the following procedure:

Step 1: Set $y_1 \leftarrow 0$ and $\ell \leftarrow 2$.

Step 2: Search for the largest integer y_ℓ ($y_{\ell-1} \le y_\ell \le y_{\ell-1} + K$) such that $\Pi(y_\ell) \le (\ell{-}1)\cdot 2t_{01}$.

Step 3: If $y_\ell > y_{\ell-1}$, then

set $\eta_i(\ell) \leftarrow y_\ell - y_{\ell-1}$;

set $\ell \leftarrow \ell + 1$ and go to Step 2.

If $y_\ell = y_{\ell-1}$, then

set $S_{ij} \leftarrow \min\left\{\ell{-}1, \lfloor \frac{P_j - P_i}{2t_{01}} \rfloor\right\}$ for $j = i{+}1, i{+}2, \ldots, n{+}1$.

Note that in Step 3, if $y_\ell > y_{\ell-1}$, then the ℓ-th nonstop shipment will deliver $(y_\ell - y_{\ell-1})$ jobs. Otherwise, there is not enough time between the completion of shipment $l - 1$ and time P_j for an additional shipment. Therefore, $S_{ij} \le \ell - 1$ for $j = i{+}1, i{+}2, \ldots, n{+}1$. It is easy to see that the running time of this procedure is $O(n)$.

For $j > i$, the number of jobs completed during the time interval $(P_i, P_j]$ that are not delivered by the nonstop shipments is equal to

$$\bar{\eta}_{ij} = (j - i) - \sum_{k=2}^{S_{ij}} \eta_i(k).$$

Note that

$$\eta_{ij} = \begin{cases} \bar{\eta}_{ij}, & \text{if } \bar{\eta}_{ij} \le K; \\ +\infty, & \text{if } \bar{\eta}_{ij} > K. \end{cases}$$

Therefore, $\eta_{i,i+1}, \eta_{i,i+2}, \ldots, \eta_{i,n+1}$ can be obtained recursively in $O(n)$ time as well. $\Box$

After the quantities in Lemma 1 are predetermined, $F'_{i,i+1}, F'_{i,i+2}, \ldots, F'_{i,n+1}$ can be obtained recursively in $O(n)$ time for each $i = 1, 2, \ldots, n{+}1$. This implies that the overall running time required for the dynamic program presented here is bounded by $O(n^2)$.

4 The Multiple-Customer Case

In this section, we consider the general problem with m customers. In subsection 4.1, we will develop an algorithm for solving this general case. This algorithm has a polynomial time complexity when m is a fixed number. Note that the general problem allows the vehicle to visit multiple customers in each delivery trip (i.e., milk runs are allowed). In subsection 4.2, we will discuss the special case when no milk runs are allowed, that is, the vehicle has to return to the plant immediately after a delivery is made.

4.1 The general case

We first present a dynamic programming algorithm for solving the problem with two customers. By Theorem 2(iv) and assuming that the vehicle will stay idle before the completion time of the first job, there can only be a finite number of time points for the start time of a trip of the vehicle. The vehicle will transport a batch of jobs either at the completion of the last job in this batch or immediately after it returns to the plant. Thus, the possible departure times of the vehicle from the plant are $C_\ell + q_1\cdot 2t_{01} + q_2\cdot 2t_{02} + q_3\cdot(t_{01}+t_{02}+t_{12})$ ($\ell = 1, 2, \ldots, n$; $q_1 = 0, 1, \ldots, n_1$; $q_2 = 0, 1, \ldots, n_2$; $q_3 = 0, 1, \ldots, \min\{n_1, n_2\}$). Let T be the set of these departure times. By Theorem 2(i)&(iii), $C_\ell = P_i^{(1)} + P_j^{(2)}$ for some $i = 0, 1, \ldots, n_1$ and $j = 0, 1, \ldots, n_2$. Thus, $|T| = O(n^5)$.

Define $g(i, j, t)$ as the minimum possible total arrival time for jobs $J_i^{(1)}, J_{i+1}^{(1)}, \ldots, J_{n_1}^{(1)}$ and $J_j^{(2)}, J_{j+1}^{(2)}, \ldots, J_{n_2}^{(2)}$ given that the vehicle is available for these jobs at time t, where $t \in T$, $i = 1, 2, \ldots, n_1$, and $j = 1, 2, \ldots, n_2$. Define

$$\delta_x = \begin{cases} 1, & \text{if } x > 0; \\ 0, & \text{if } x = 0. \end{cases}$$

We have the following recurrence relation:

$$\begin{aligned} g(i,j,t) = \min_{\substack{0\le u\le n_1-i+1\\ 0\le v\le n_2-j+1\\ 0<u+v\le K}} \Big\{ &(1-\delta_v)\cdot u(\hat{t}+t_{01}) + (1-\delta_u)\cdot v(\hat{t}+t_{02}) \\ &+ \delta_u\delta_v\cdot \min\{u(\hat{t}+t_{01}) + v(\hat{t}+t_{01}+t_{12}),\ v(\hat{t}+t_{02}) + u(\hat{t}+t_{02}+t_{12})\} \\ &+ g(i+u,\ j+v,\ \hat{t}+(1-\delta_v)\cdot 2t_{01} + (1-\delta_u)\cdot 2t_{02} + \delta_u\delta_v\cdot(t_{01}+t_{02}+t_{12}))\Big\}, \end{aligned}$$

for $t \in T$, $i = 1, 2, \ldots, n_1+1$, and $j = 1, 2, \ldots, n_2+1$, where $\hat{t} = \max\{t,\ P^{(1)}_{i+u-1} + P^{(2)}_{j+v-1}\}$. The boundary condition is

$$g(n_1+1, n_2+1, t) = 0,$$

for all $t \in T$, and the objective is $g(1, 1, 0)$.

In the above recurrence relation, we consider the delivery of jobs $J^{(1)}_i, J^{(1)}_{i+1}, \ldots, J^{(1)}_{n_1}$ and $J^{(2)}_j, J^{(2)}_{j+1}, \ldots, J^{(2)}_{n_2}$. Variables u and v are the number of jobs to be delivered to customers 1 and 2, respectively, during the first trip. By Theorem 2(ii)–(iii), these $u + v$ jobs are $J^{(1)}_i, J^{(1)}_{i+1}, \ldots, J^{(1)}_{i+u-1}$ and $J^{(2)}_j, J^{(2)}_{j+1}, \ldots, J^{(2)}_{j+v-1}$. If the vehicle is available for delivering these jobs at time t, then it departs at time $\hat{t}$. If $v = 0$ (i.e., $\delta_v = 0$) then the batch will arrive at customer 1 at time $\hat{t} + t_{01}$. If $u = 0$ (i.e., $\delta_u = 0$), then the batch will arrive at customer 2 at time $\hat{t} + t_{02}$. If $u, v > 0$ (i.e., $\delta_u \delta_v = 1$), then the batch will either arrive at customers 1 and 2 at times $\hat{t} + t_{01}$ and $\hat{t} + t_{01} + t_{12}$, respectively, or arrive at customers 2 and 1 at times $\hat{t} + t_{02}$ and $\hat{t} + t_{02} + t_{12}$, respectively. The condition "$u + v \le K$" ensures that the vehicle capacity is not violated. Note that the total number of possible combinations of i, j, and t is $O(n^4 \cdot |T|)$. Therefore, this dynamic program solves the two-customer problem in $O(n^9)$ time.

Next, we consider the case with m customers, where m is fixed. The above dynamic programming method can easily be extended to this case by extending the optimal value function to $g(i_1, i_2, \ldots, i_m, t)$. When there are m customers located at different locations, the number of possible departure times, $|T|$, is $O(n^{m(m+3)/2})$. This is because there are $(n+1)^m$ possible values for the job completion time $P^{(1)}_{i_1} + \cdots + P^{(m)}_{i_m}$ and $(n+1)^{m(m+1)/2}$ possible values for the travel time of a set of nonstop shipments (since there are $m(m+1)/2$ links in the network of customers and each of them is traversed at most n times). Thus, the number of possible states in the dynamic program is $O(n^{m(m+5)/2})$, because there are $(n+1)^m$ possible values for the m-tuple $(i_1, i_2, \ldots, i_m)$. Evaluating each $g(i_1, i_2, \ldots, i_m, t)$ takes $O(n^m)$ time. (Note that the number of terms in the right-hand side of the recurrence relation is dependent on m. However, it becomes a constant when m is fixed.) Hence, the overall complexity of the dynamic program is $O(n^{m(m+7)/2})$. This implies that the m-customer problem is polynomial time solvable when m is fixed.

4.2 The case with no milk run allowed

We now discuss a special situation in which the vehicle is only allowed to deliver to customers directly and no milk runs are allowed. Note that this can be viewed as a special case of our model with $t_{ij} = t_{0i} + t_{0j}$ for any pair of customers i and j. This is because if $t_{ij} = t_{0i} + t_{0j}$ for all i and j, then each milk run can be converted into a set of direct shipments without affecting the job arrival times.

We first consider the problem with only two customers. In this case, the possible departure times of the vehicle at the plant are $C_\ell + q_1 \cdot 2t_{01} + q_2 \cdot 2t_{02}$ ($\ell = 1, 2, \ldots, n$; $q_1 = 0, 1, \ldots, n_1$; $q_2 = 0, 1, \ldots, n_2$). Thus, the number of possible departure times, $|T|$, is $O(n^4)$. Moreover, the recurrence relation presented in subsection 4.1 can be simplified to

$$\begin{aligned} g(i, j, t) \;=\; \min\Big\{ & \min_{0<u\leq \min\{n_1-i+1, K\}} \big\{ u(\hat{t} + t_{01}) + g(i{+}u,\, j,\, \hat{t} + 2t_{01}) \big\}, \\ & \min_{0<v\leq \min\{n_2-j+1, K\}} \big\{ v(\hat{t} + t_{02}) + g(i,\, j{+}v,\, \hat{t} + 2t_{02}) \big\} \Big\}, \end{aligned}$$

for $t \in T$, $i = 1, 2, \ldots, n_1{+}1$, and $j = 1, 2, \ldots, n_2{+}1$. Thus, the number of possible states is $O(n^6)$, and each iteration of the dynamic program takes only $O(n)$ time. Hence, when milk runs are not allowed, the complexity of the dynamic program for the two-customer problem is reduced to $O(n^7)$.

Now, consider the case with m customers, where m is fixed. The above dynamic programming method can easily be extended to this case. When there are m customers located at different locations and milk runs are not allowed, the number of possible departure times, $|T|$, is $O(n^{2m})$. Thus, the number of possible states in the dynamic program is $O(n^{3m})$. When m is fixed, evaluating each $g(i_1, i_2, \ldots, i_m, t)$ takes $O(n)$ time. Hence, the overall complexity of the dynamic program is $O(n^{3m+1})$. Note that this computational complexity is significantly lower than that of the general case presented in subsection 4.1. However, it remains an interesting open question whether or not this special case is polynomial time solvable when m is not fixed.

Next, we consider the special case of the no-milk-run problem when the vehicle is uncapacitated and present a dynamic program that can solve this special case with a lower computational complexity. We first consider the case with only two customers. Define $h(i, j, \tau, k)$ as the minimum possible total arrival time for jobs $J_1^{(1)}, J_2^{(1)}, \ldots, J_i^{(1)}, J_1^{(2)}, J_2^{(2)}, \ldots, J_j^{(2)}$ if the vehicle's last departure from the plant with any of these jobs takes place at time τ, and that last shipment is delivered to customer k, where $i \in \{0, 1, \ldots, n_1\}$, $j \in \{0, 1, \ldots, n_2\}$, $\tau \in \{\tau' \in T \mid \tau' \geq P_i^{(1)} + P_j^{(2)}\}$, and $k \in \{1, 2\}$. We have the following recurrence relations:

$$h(i,j,P_i^{(1)}+P_j^{(2)},1) = \min\Big\{h(i-1,j,P_i^{(1)}+P_j^{(2)},1), \min_{\substack{k'=1,2;\ \tau'\in T;\\ \tau'\leq P_i^{(1)}+P_j^{(2)}-2t_{0k'}}} \Big\{h(i-1,j,\tau',k')\Big\}\Big\} + P_i^{(1)} + P_j^{(2)} + t_{01}; \quad (2)$$

$$h(i,j,P_i^{(1)}+P_j^{(2)},2) = \min\Big\{h(i,j-1,P_i^{(1)}+P_j^{(2)},1), \min_{\substack{k'=1,2;\ \tau'\in T;\\ \tau'\leq P_i^{(1)}+P_j^{(2)}-2t_{0k'}}} \Big\{h(i,j-1,\tau',k')\Big\}\Big\} + P_i^{(1)} + P_j^{(2)} + t_{02}; \quad (3)$$

and for $\tau > P_i^{(1)}+P_j^{(2)}$,

$$h(i,j,\tau,1) = \min\Big\{h(i-1,j,\tau,1),\ h(i-1,j,\tau-2t_{01},1),\ h(i-1,j,\tau-2t_{02},2)\Big\} + \tau + t_{01}; \quad (4)$$

$$h(i,j,\tau,2) = \min\Big\{h(i,j-1,\tau,2),\ h(i,j-1,\tau-2t_{01},1),\ h(i,j-1,\tau-2t_{02},2)\Big\} + \tau + t_{02}. \quad (5)$$

The boundary conditions are $h(0,0,\tau,k) = 0$ for all $\tau \in T$, $k \in \{1,2\}$ and $h(i,j,\tau,k) = +\infty$ if $\tau < P_i^{(1)}+P_j^{(2)}$. The objective is $\min_{\tau\in T; k=1,2}\{h(n_1,n_2,\tau,k)\}$.

Equation (2) is for the case when the vehicle departs from the plant immediately at the completion of $J_i^{(1)}$ and delivers this job to customer 1 (see Figure 4(a)). There are two possibilities. If the vehicle also carries $J_{i-1}^{(1)}$ on this trip, then the minimum total job arrival time for jobs $J_1^{(1)}, \ldots, J_{i-1}^{(1)}, J_1^{(2)}, \ldots, J_j^{(2)}$ is $h(i-1,j,P_i^{(1)}+P_j^{(2)},1)$. Otherwise, let k' denote the customer that the vehicle visits in the preceeding trip and let τ' denote the departure time of that trip. Then τ' must be no greater than $P_i^{(1)} + P_j^{(2)} - 2t_{0k'}$ and the minimum total job arrival time for jobs $J_1^{(1)}, \ldots, J_{i-1}^{(1)}, J_1^{(2)}, \ldots, J_j^{(2)}$ is $h(i-1,j,\tau',k')$. This explains the construction of Equation (2). The construction of Equation (3) follows the same argument. The computational effort required to compute each $h(i,j,P_i^{(1)}+P_j^{(2)},k)$ is $O(|T|)$. Thus, the running time required to compute all $h(i,j,P_i^{(1)}+P_j^{(2)},k)$ is $O(n^2 \cdot |T|)$.

Insert Figure 4 about here

Equation (4) is for the case when the vehicle departs from the plant at time $\tau > P_i^{(1)} + P_j^{(2)}$ when carrying $J_i^{(1)}$ to its customer. By Theorem 2(iv), there must be another trip immediately before this one, that is, the departure time of the preceeding trip must be $\tau - 2t_{0k'}$ if it is a delivery to customer k' (see Figure 4(b)). Thus, there are three possible scenarios. Either $J_{i-1}^{(1)}$ also departs from the plant at time τ, or it departs from the plant at time $\tau - 2t_{01}$, or $J_j^{(2)}$ departs from the plant at time $\tau - 2t_{02}$. This explains the construction of Equation (4). The construction of Equation (5) follows the same argument. When $\tau > P_i^{(1)} + P_j^{(2)}$, the computational effort required to compute each $h(i, j, \tau, k)$ is $O(1)$. The number of combinations of i, j, τ, and k is $O(n^2 \cdot |T|)$. Therefore, the overall complexity of this dynamic program is $O(n^2 \cdot |T|) = O(n^6)$.

Now, consider the case with m customers when m is fixed. The above dynamic programming method can be extended to this case. Recall that the number of possible departure times, $|T|$, is $O(n^{2m})$. It is easy to check that, when m is fixed, the complexity of the extended dynamic program is $n^m \cdot |T| = O(n^{3m})$.

5 Conclusions

We have presented an analysis of the single-machine scheduling problem with deliveries to multiple customers. While the problem is NP-hard in general, the single-customer case can be solved efficiently in $O(n^2)$ time. We have developed a dynamic program for solving the general case. The computational complexity of the dynamic program is quite high when m is greater than 1, although it is a polynomial function of n when m is fixed. We have shown that the complexity can be lowered when the deliveries are restricted to direct shipments. The complexity is further lowered when the vehicle is uncapacitated.

In future research, it will be worth considering the development of efficient and effective heuristics for the general problem with an arbitrary number of customers. It is also worth considering the development of optimal algorithms with lower computational complexity for the case when the number of customers is fixed. Furthermore, it is interesting to extend the existing model to and develop solution methods for problems with multiple delivery vehicles as well as problems with other objective functions.

References

[1] Afrati, F., S. Cosmadakis, C.H. Papadimitriou and G. Papageorgiou, The complexity of the travelling repairman problem, *Theoretical Informatics and Applications* **20,** 79–87, 1986.

[2] Chang, Y.-C. and C.-Y. Lee, Machine scheduling with job delivery coordination, *European Journal of Operational Research,* forthcoming.

[3] Chen, Z.-L. and G.L. Vairaktarakis, Integrated scheduling of production and distribution operations, *Management Science,* forthcoming.

[4] Geismar, H.N., L. Lei and C. Sriskandarajah, The integrated production and transportation scheduling problem for a product with a short life span and non-instantaneous transportation time. Submitted for publication.

[5] Hall, L.A. and D.B. Shmoys, Jackson's rule of single-machine scheduling: Making a good heuristic better, *Mathematics of Operations Research* **17,** 22–35, 1992.

[6] Kise, H., T. Shioyama and T. Ibaraki, Automated two-machine flowshop scheduling: A solvable case, *IIE Transactions* **23,** 10–16, 1991.

[7] Langston, M.A., Interstage transportation planning in the deterministic flow-shop environment, *Operations Research* **35,** 556–564, 1987.

[8] Lee, C.-Y. and Z.-L. Chen, Machine scheduling with transportation considerations, *Journal of Scheduling* **4,** 3–24, 2001.

[9] Maggu, P.L. and G. Das, On $2 \times n$ sequencing problem with transportation times of jobs, *Pure and Applied Mathematika Sciences* **12,** 1–6, 1980.

[10] Maggu, P.L., G. Das and R. Kumar, On equivalent-job for job-block in $2 \times n$ sequencing problem with transportation-times, *Journal of the Operations Research Society of Japan* **24,** 136–146, 1981.

[11] Panwalkar, S.S., Scheduling of a two-machine flowshop with travel time between machines, *Journal of the Operational Research Society* **42,** 609–613, 1991.

[12] Potts, C.N., Analysis of a heuristic for one machine sequencing with release dates and delivery times, *Operations Research* **28,** 1436–1441, 1980.

[13] Sahni, S. and T. Gonzalez, P-complete approximation problems, *Journal of the Association for Computing Machinery* **23,** 555–565, 1976.

[14] Sahni, S. and G. Vairaktarakis, The master–slave paradigm in parallel computer and industrial settings, *Journal of Global Optimization* **9,** 357–377, 1996.

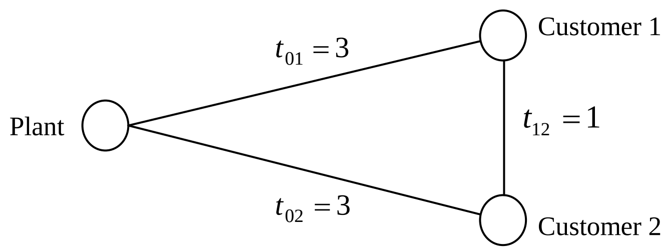
[15] Stern, H.I. and G. Vitner, Scheduling parts in a combined production-transportation work cell, *Journal of the Operational Research Society* **41,** 625–632, 1990.

[16] Stevens, J.W. and D.D. Germill, Scheduling a two-machine flowshop with travel times to minimize maximum lateness, *International Journal of Production Research* **35,** 1–15, 1997.

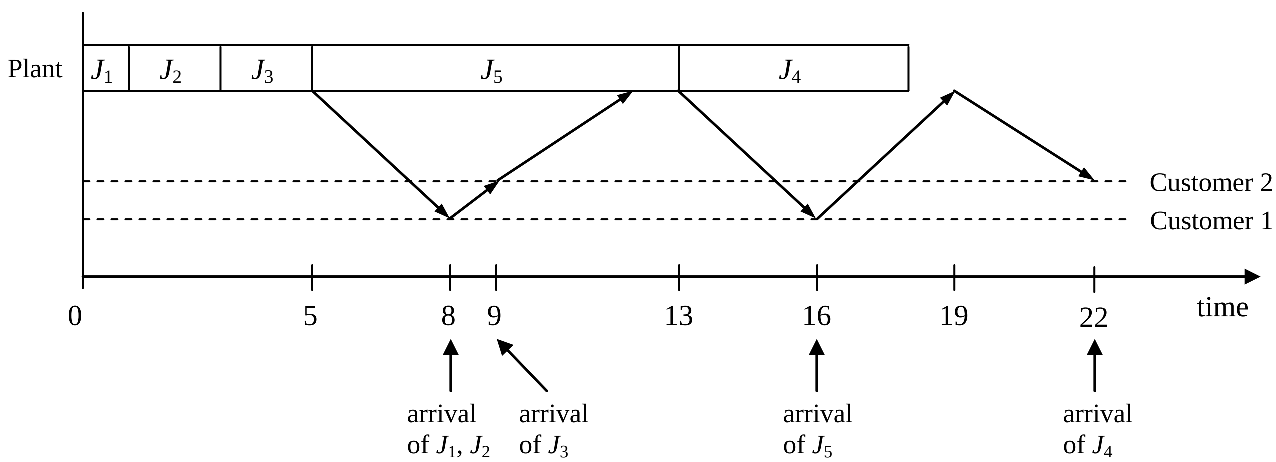
[17] Vairaktarakis G., Analysis of algorithms for master–slave systems, *IIE Transactions* **29,** 939–949, 1997.

j	1	2	3	4	5
p_j	1	2	2	5	8
customer of J_j	1	1	2	2	1

(a) problem data



(b) travel times between the plant and the customers



(c) a feasible schedule

Figure 1. A two-customer example

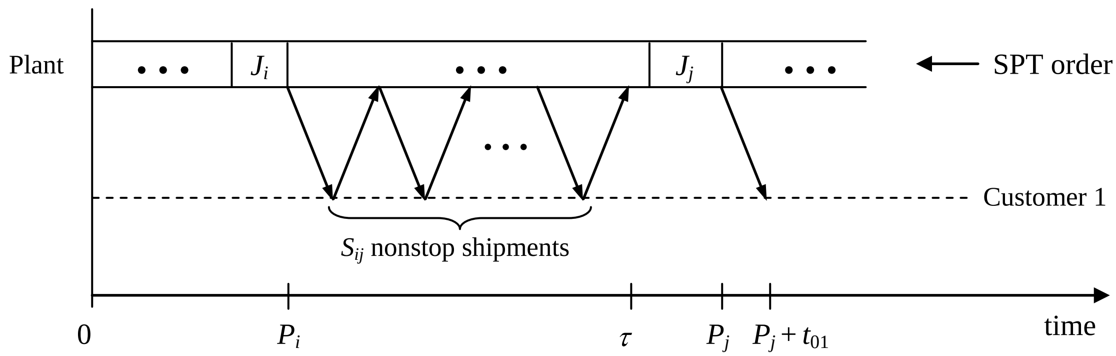


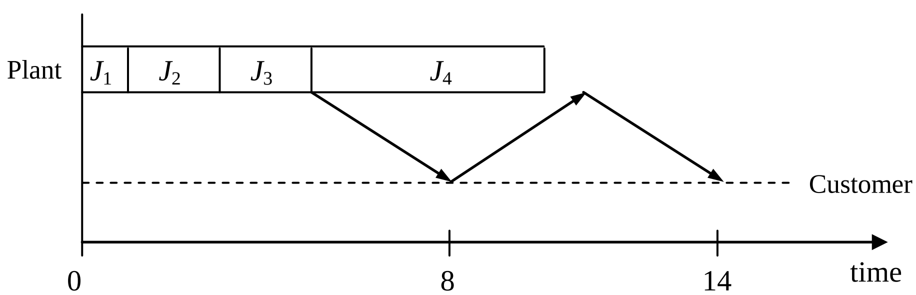
Figure 2. Nonstop shipments take place after the delivery of J_i in the single-customer case

j	1	2	3	4	5
p_j	1	2	2	5	24
P_j	1	3	5	10	34

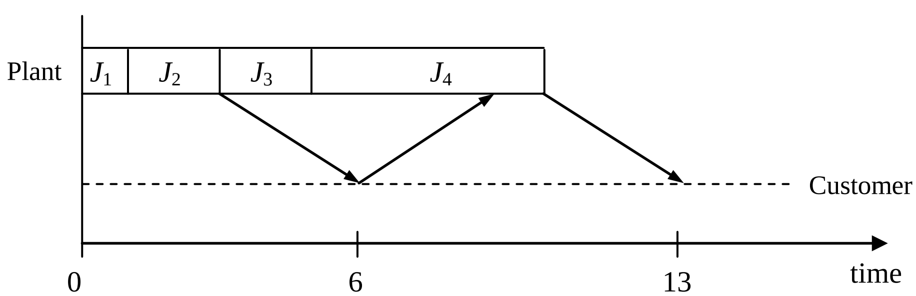
(a) job processing times and cumulative processing times

$S_{14} = 1$	–	$\eta_{14} = +\infty$	$F'_{14} = +\infty$
$S_{24} = 1$	–	$\eta_{24} = 2$	$F'_{24} = 26$
$S_{15} = 3$	$\eta_1(2) = 2,\ \eta_1(3) = 1$	$\eta_{15} = 1$	$F'_{15} = 73$
$S_{25} = 3$	$\eta_2(2) = 1,\ \eta_2(3) = 1$	$\eta_{25} = 1$	$F'_{25} = 67$
$S_{35} = 2$	$\eta_3(2) = 1$	$\eta_{35} = 1$	$F'_{35} = 51$
$S_{45} = 1$	–	$\eta_{45} = 1$	$F'_{45} = 37$

(b) the values of S_{ij}, $\eta_i(k)$, η_{ij}, and F'_{ij}



(c) an optimal schedule ($\Sigma\hat{C}_j = 8+8+8+14 = 38$)



(d) an alternative optimal schedule ($\Sigma\hat{C}_j = 6+6+13+13 = 38$)

Figure 3. An example for the single-customer algorithm

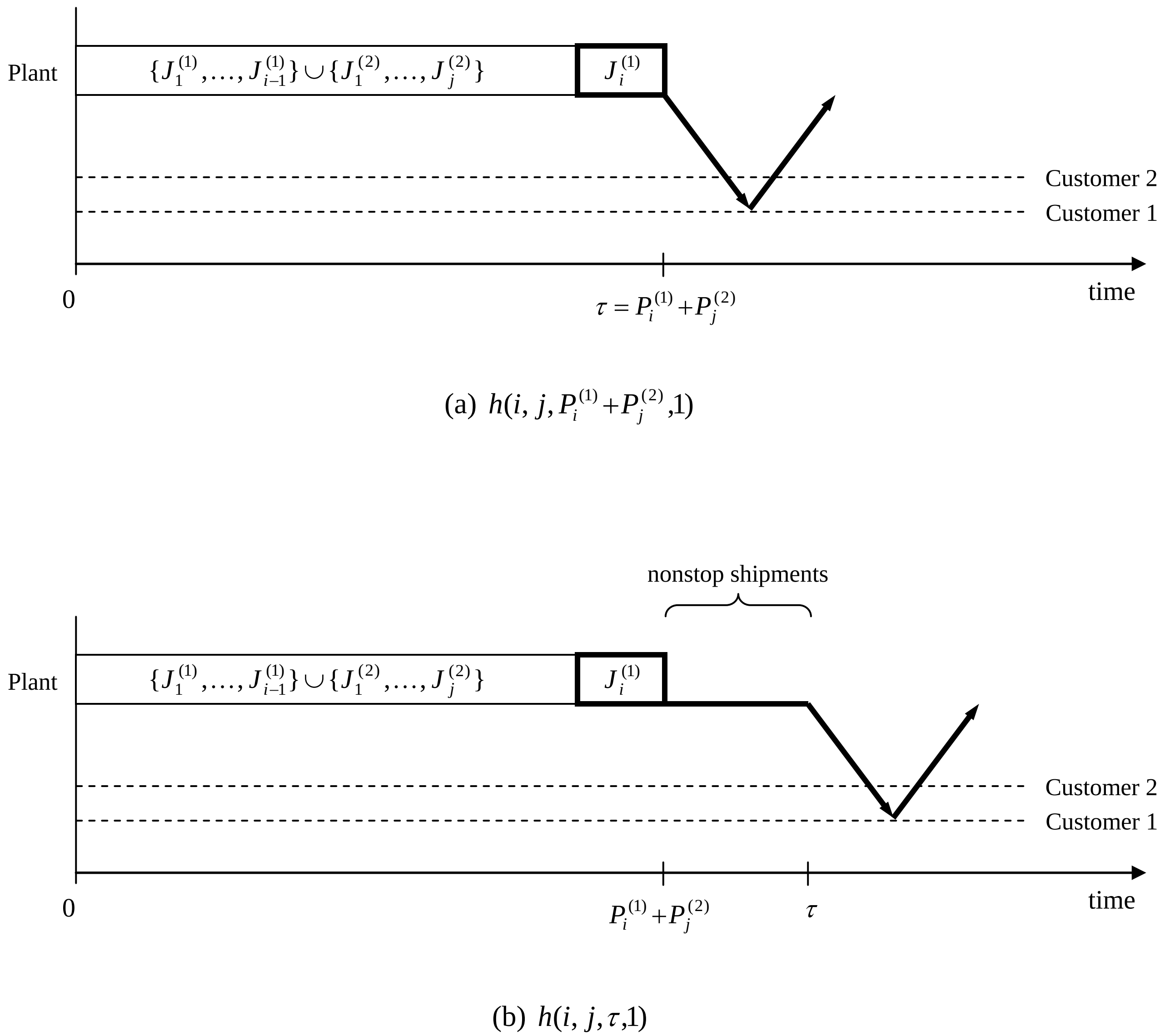


Figure 4. Recurrence relations of the two-customer, no-milk-run, uncapacitated problem