Coordinating Production and Distribution of Jobs with Bundling Operations

Chung-Lun Li

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

George Vairaktarakis

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

August 2005

Revised January 2006

Abstract

We consider an integrated scheduling and distribution model in which jobs completed by two different machines must be bundled together for delivery. The objective is to minimize the sum of delivery cost and customers' waiting cost. Such a model not only attempts to coordinate the job schedules on both machines, but it also aims to coordinate the machine schedules with the delivery plan. Polynomial-time heuristics and approximation schemes are developed for the model with only direct shipments as well as the general model with milk-run deliveries.

Key words: Scheduling; distribution; bundling operations; coordination; polynomial-time approximation schemes

1 Introduction

Production planning and distribution management are both critical to the success of a manufacturer. Traditional management approaches consider these two functions separately and sequentially. For example, when the production department of a manufacturing firm plans its machine schedules, it normally ignores the delivery arrangements of the finished jobs. Recent advances in supply chain management research have drawn a lot of attention to the integration and coordination among different functions across a supply chain. In particular, the integration of production and delivery schedules has high potential to achieve significant savings in a manufacturer's operating cost.

In many production and transportation planning environments, the delivery of finished goods is constrained by the processing of different component parts of the product. It is also quite common that different components of a product have to be processed by their own dedicated machines or work centers. For example, in the printing of newspapers and magazines, a printing machine can be used specifically for printing the main pages, while a different printer is dedicated to the printing of supplements and inserts. In such a case, a job is not ready for delivery unless the processing of both printing tasks is complete. In other words, the printed main pages and the printed supplements/inserts must be bundled together for delivery. Take another example in furniture production, where wooden parts and glass parts are typically produced by different work centers. Both the glass and wooden parts of the finished product must be bundled together before it can be delivered to the customers.

In the above examples, if we want to obtain a "systemwide" optimal production and delivery plan, it is important that the sequencing and scheduling of the tasks at the machines (or work centers) and the delivery arrangements of the finished jobs are considered at the same time. When such an integrated plan is developed, the scheduler faces a tradeoff between providing quick deliveries and minimizing shipping costs. Quick deliveries minimize customers' waiting time, while low shipping costs directly benefit the company's bottom line.

In this paper, we develop and analyze a mathematical model in which there are two machines located at the same production facility. Each job is composed of two tasks; each of them must be processed by a dedicated machine. The job is ready for delivery once the processing of both of its tasks has been finished. We will call such a machine processing environment "bundling operations." The distribution of finished jobs is carried out by third-party carriers, and therefore, there is an unlimited number of vehicles available. The objective is to minimize the sum of transportation cost, which depends mainly on the number of vehicles dispatched and their routing distances, and customers' waiting cost, which depends on the arrival time of the finished jobs at customers' locations.

A number of researchers have considered machine scheduling problems with bundling operations but without delivery considerations. Such machine scheduling problems can also be viewed as open-shop scheduling problems with jobs overlap. They are similar to the classical open-shop model, but they allow the operations of any given job to overlap in time. Wagneur and Sriskandarajah (1993), Ahmadi *et al.* (2005), Leung *et al.* (2005c), and Yang (2005) have studied the NP-hardness of the machine scheduling problem with bundling operations. In particular, Ahmadi *et al.* (2005) and Yang (2005) have proven independently that the two-machine bundling operations problem with an objective of minimizing total job completion time is NP-hard in the strong sense. Sung and Yoon (1998) and Ahmadi *et al.* (2005) have considered the same problem with a generalized objective of minimizing the total weighted job completion time and have developed efficient heuristics for it. Leung *et al.* (2005a, 2005b, 2006b) have studied the same problem with multiple machines, job release dates, due dates, and various objective functions. Cai and Zhou (2004) have studied a stochastic scheduling model with bundling operations.

The machine scheduling problem with bundling operations is similar to the "customer order scheduling problem on parallel machines." The customer order problem differs from the bundling operations problem in that tasks from a job can be processed by any machine. Different variants of the customer order problem have been studied by Julien and Magazine (1990), Blocher and Chhajed (1996), Blocher *et al.* (1998), Leung *et al.* (2006a, 2005d), Yang (2003, 2005), and Yang and Posner

(2005), among others. None of the abovementioned research work has considered the delivery of the completed jobs of the bundling machines.

Another scheduling model related to ours is the "three-machine assembly-type flowshop" model developed by Lee *et al.* (1993). In their model, two machines are responsible for producing component parts, while the finished components are fed into an assembly machine for final assembly. The difference between our model and the assembly-type flowshop model is that instead of having an assembly machine for final assembly, we consider the delivery arrangements of the finished components. The delivery arrangements are much more complicated than the operations of a single assembly machine, since they involve delivery cost, delivery time, vehicle capacity, and routing decisions.

The issues of integrating production and distribution have been addressed by a number of researchers. Some of the work focuses on the strategic or tactic level of decision; see, for example, Fumero and Vercellis (1999) and Sarmiento and Nagi (1999). The integration of production and distribution at the operations scheduling level has received increasing attention recently; see, for example, Hall and Potts (2003) and Chen (2004). In particular, some of the recent work considers the integration of machine scheduling and finished job delivery. Lee and Chen (2001) have developed and analyzed scheduling models in which all of the jobs belong to one customer and the number of delivery vehicles is limited, and therefore, the capacitated vehicles need to travel back and forth between the machine and the customer. Li and Ou (2005) have considered a similar scheduling model with pickup and delivery considerations. Chang and Lee (2004) have extended one of Lee and Chen's models to the situation in which every job occupies a different amount of vehicle space. Li *et al.* (2005) have studied scheduling problems with multiple customer locations and routing decisions of a single delivery vehicle. Chen and Vairaktarakis (2005) have studied the problem of minimizing a convex combination of distribution cost and a customer service measure, where multiple capacitated delivery vehicles are used for delivering the finished jobs to multiple customer locations. Geismar *et al.* (2005) have considered a similar problem setting, but their model involves a product with a short shelf life and allows no inventory in the process. Pundoor and Chen (2005) have studied a scheduling

model with direct deliveries of jobs to customers and order due dates. Chen and Pundoor (2006) have analyzed an integrated order assignment, machine scheduling, and transportation planning model with multiple machine locations and a single destination. However, all of these papers focus on the integration of a single/parallel machine schedule and the transportation decisions. In this paper, we consider the integration of bundling operations and the delivery arrangements of finished jobs to multiple customer locations, where we take into consideration the transportation cost, customers' waiting time, and vehicle capacity. We also consider two types of delivery, namely direct delivery and delivery with milk runs.

Our model is mathematically defined as follows. There are two machines M_1, M_2 located at a production plant and a given set of n jobs $\mathbf{J} = \{J_1, J_2, \ldots, J_n\}$, where each job J_j is made up of a pair of tasks, namely an a-task and a b-task. The a-task of J_j must be processed by M_1 and requires an uninterrupted processing time of $a_j \geq 0$, while the b-task of J_j must be processed by M_2 and requires an uninterrupted processing time of $b_j \geq 0$ ($j = 1, 2, \ldots, n$). A job is delivered to its customer after its a-task and b-task have both been completed. Customers are located in various geographical locations. There are $h > 0$ customer locations in total, where h is a fixed number. Thus, $\mathbf{J}$ is partitioned into $\{\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_h\}$, where $\mathbf{J}_i$ is the subset of jobs whose customers are located at location i. The travel time from the plant to customer location i is $t_{0i} \geq 0$, and the travel time from customer location i to customer location j is $t_{ij} \geq 0$ (note: $t_{ii} = 0$). At most $\Gamma > 0$ completed jobs may be delivered together in the same batch, and the delivery cost of a batch of jobs to customer locations $i_1, i_2, \ldots, i_k$ following this sequence is $f + g_{0i_1} + g_{i_1 i_2} + \cdots + g_{i_{k-1} i_k} + g_{i_k 0}$. Here, Γ represents the capacity of the delivery vehicle, $f \geq 0$ is the fixed cost for dispatching the vehicle, and $g_{ij} \geq 0$ is the travel cost between two locations. In addition to this delivery cost, there is a waiting cost of μE_j if job J_j arrives at its customer at time E_j, where $\mu \geq 0$ represents a customer's cost of waiting per time unit. Suppose the customer of J_j is at location i_k and the delivery vehicle that carries job J_j departs from the plant location at time D_j and travels through customer locations $i_1, i_2, \ldots, i_k$ in this sequence. Then $E_j = D_j + t_{0i_1} + t_{i_1 i_2} + \cdots + t_{i_{k-1} i_k}$. All jobs are available for processing

at time 0. We assume that there are enough vehicles available so that a batch of jobs can leave the machine location as soon as all of them have finished processing. For simplicity of analysis, we assume that for each $j = 1, 2, \ldots, n$, at least one of a_j, b_j is nonzero. The objective is to schedule the a-tasks and b-tasks of the jobs on the machines, assign jobs to batches, and sequence the deliveries of the batches so as to minimize the total cost, which includes the delivery cost of each batch of jobs and the cost of waiting of each job. We denote this "general problem" as $\mathbf{P}^{gen}$.

For simplicity, we also assume that $\Gamma \leq n$. Let $n_i = |\mathbf{J}_i|$ (note: $\sum_{i=1}^{h} n_i = n$). Let C'_j and C''_j denote the completion time of the a-task and b-task, respectively, of job J_j at the machines. The completion time of processing of job J_j is given as $C_j = \max\{C'_j, C''_j\}$. Thus, $D_j \geq C_j$ for $j = 1, 2, \ldots, n$.

For example, a feasible solution to a problem instance with $n = 5$, $h = 4$, and $\Gamma = 3$ is depicted in Figure 1. In this problem instance, jobs J_1, J_2, and J_5 belong to customers at different locations, while the customers of jobs J_3 and J_4 are at the same location. In the solution depicted in the figure, the first vehicle departs from the machine after jobs J_1, J_2, and J_5 are completed, and it first delivers J_1, followed by J_5, and then followed by J_2. The second vehicle departs from the machine after jobs J_3 and J_4 are completed, and it delivers these two jobs to their customers simultaneously. The total cost of this solution is $\mu\big[3(C_1+t_{01})+2t_{14}+t_{42}\big]+\mu\big[2(C_2+t_{03})\big]+\big[f+g_{01}+g_{14}+g_{42}+g_{20}\big]+\big[f+g_{03}+g_{30}\big]$, where $C_1 = b_2 + b_5 + b_1$ and $C_2 = a_2 + a_1 + a_5 + a_3 + a_4$.

In this paper, we consider problem $\mathbf{P}^{gen}$ as well as two of its variants. The first variant, which we will call the "production scheduling subproblem," does not consider the batching and delivery of jobs, and the objective becomes the minimization of the sum of job completion times, $\sum_{j=1}^{n} C_j$. This is the case when the delivery time and cost (i.e., t_{0i}, t_{ij}, g_{0i}, g_{ij}, and f) are negligible. We denote this production scheduling subproblem as $\mathbf{P}^{pdt}$. The second variant, which we will call the "direct delivery problem," allows only the delivery of jobs to one customer location at a time (i.e., it disallows "milk-run deliveries"). This is the case when $g_{ij} = +\infty$ for all $i, j \neq 0$. In this case, the delivery cost of a batch becomes $f + g_{0i} + g_{i0}$ if the batch is delivered to the customers at location

i. We denote this direct delivery problem as $\mathbf{P}^{dir}$.

As mentioned earlier, Ahmadi *et al.* (2005) and Yang (2005) have shown that problem $\mathbf{P}^{pdt}$ is strongly NP-hard. Since $\mathbf{P}^{pdt}$ is a special case of both $\mathbf{P}^{dir}$ and $\mathbf{P}^{gen}$, it is unlikely that polynomial-time algorithms exist for problems $\mathbf{P}^{pdt}$, $\mathbf{P}^{dir}$, and $\mathbf{P}^{gen}$. Hence, we target at the development of polynomial-time heuristics and approximation schemes for these problems. Throughout the paper, we denote an optimal schedule of any problem $\mathbf{P}$ as $\sigma^*(\mathbf{P})$, and we denote the total cost of any schedule σ as $Z(\sigma)$.

The rest of the paper is organized as follows. In Section **??**, we develop polynomial-time approximation schemes for the production scheduling subproblem. In Section **??**, we extend the polynomial-time approximation schemes to solve the direct delivery problem. Similarly, in Section **??**, we extend our approximation schemes to solve the general problem. We conduct computational experiments and report their results in Section **??**. We conclude our paper in Section **??**. Proofs of all theorems and lemmas are provided in the Appendix.

2 The Production Scheduling Subproblem

In this section, we consider problem $\mathbf{P}^{pdt}$. This special case contains only the two-machine scheduling decision with bundling operations. For simplicity, in this case we assume $\mu = 1$. Thus, the objective is to minimize the sum of job completion times. The following lemma provides some important properties of problem $\mathbf{P}^{pdt}$.

Lemma 1 *There exists an optimal solution to problem* $\mathbf{P}^{pdt}$ *in which (i) there is no idle time on either machine, (ii) the job sequences on both machines are identical, and (iii) if* ($a_j < a_k$ *and* $b_j \leq b_k$) *or* ($a_j = a_k$ *and* $b_j < b_k$) *then* J_j *precedes* J_k.

It suffices to restrict our search to solutions that satisfy the properties stated in Lemma **??**. In the remainder of this section, we will only consider schedules that satisfy these properties.

2.1 Approximation Schemes

In this subsection, we develop approximation schemes for problem $\mathbf{P}^{pdt}$. First, suppose the job set $\mathbf{J}$ is partitioned into subsets $Q^{(1)}, Q^{(2)}, \ldots, Q^{(\ell)}$ that satisfy the following condition:

$(C1)$ If $J_j, J_k \in Q^{(r)}$ and $a_j < a_k$, then $b_j \leq b_k \quad (r = 1, 2, \ldots, \ell)$.

We arrange the jobs in each $Q^{(r)}$ in nondecreasing order of a_j and b_j, and let $J_j^{(r)}$ denote the jth job in the sorted list. Let $a_j^{(r)}$ and $b_j^{(r)}$ be the processing time of the a-task and b-task, respectively, of $J_j^{(r)}$. Denote

$$\mathbf{J}(q_1, \ldots, q_\ell) = \bigcup_{r=1}^{\ell} \{J_1^{(r)}, J_2^{(r)}, \ldots, J_{q_r}^{(r)}\},$$

$$A(q_1, \ldots, q_\ell) = \sum_{r=1}^{\ell} \sum_{j=1}^{q_r} a_j^{(r)},$$

$$B(q_1, \ldots, q_\ell) = \sum_{r=1}^{\ell} \sum_{j=1}^{q_r} b_j^{(r)},$$

and

$$C(q_1, \ldots, q_\ell) = \max\{A(q_1, \ldots, q_\ell), B(q_1, \ldots, q_\ell)\}.$$

Here, $A(q_1, \ldots, q_\ell)$ and $B(q_1, \ldots, q_\ell)$ denote the total processing time of the a-tasks and b-tasks, respectively, of the jobs in $\mathbf{J}(q_1, \ldots, q_\ell)$, while $C(q_1, \ldots, q_\ell)$ denotes the completion time of processing of the last job in a schedule that consists of the jobs in $\mathbf{J}(q_1, \ldots, q_\ell)$. The following dynamic program generates an optimal schedule.

Define $F_1(q_1, q_2, \ldots, q_\ell)$ as the minimum total flow time of the partial schedule that consists of jobs in $\mathbf{J}(q_1, \ldots, q_\ell)$, where $q_r = 0, 1, \ldots, |Q^{(r)}|$ $(r = 1, 2, \ldots, \ell)$. We have the following recurrence relation:

$$F_1(q_1, q_2, \ldots, q_\ell) = \min_{\substack{r=1,2,\ldots,\ell \\ \text{s.t. } q_r > 0}} \Big\{F_1(q_1, \ldots, q_{r-1}, q_r - 1, q_{r+1}, \ldots, q_\ell) + C(q_1, \ldots, q_\ell)\Big\}.$$

The boundary condition is $F_1(0,0,\ldots,0)=0$, and the objective is $F_1(|Q^{(1)}|,|Q^{(2)}|,\ldots,|Q^{(\ell)}|)$.

Sorting the jobs in each subset $Q^{(r)}$ requires $O(n\log n)$ time. Suppose ℓ is fixed. Then predetermining the values of $A(q_1,\ldots,q_\ell)$, $B(q_1,\ldots,q_\ell)$, and $C(q_1,\ldots,q_\ell)$ requires $O(n^\ell)$ time. Evaluating the values of all $F_1(q_1,q_2,\ldots,q_\ell)$ using the recurrence relation also takes $O(n^\ell)$ time. Hence, the overall computational complexity of this dynamic program is $O(n^\ell)$ for any fixed $\ell\geq 2$.

Next, we present a heuristic for problem $\mathbf{P}^{pdt}$. The heuristic uses a positive integer parameter β to control the efficiency and effectiveness of the heuristic. Let $\lambda=\frac{1}{2}\sqrt{(\beta+4)/\beta}-\frac{1}{2}>0$. Define:

$$S_r^a=\big\{J_j \;\big|\; (r-1)\lambda a_j\leq b_j<r\lambda a_j\big\} \qquad (r=1,2,\ldots,\beta);$$
$$S_r^b=\big\{J_j \;\big|\; (r-1)\lambda b_j\leq a_j<r\lambda b_j\big\} \qquad (r=1,2,\ldots,\beta);$$
$$S^c=\big\{J_j \;\big|\; a_j\geq\beta\lambda b_j \text{ and } b_j\geq\beta\lambda a_j\big\},$$

(see Figure 2). Let $\Pi_\beta=\{S_1^a,S_2^a,\ldots,S_\beta^a,S_1^b,S_2^b,\ldots,S_\beta^b,S^c\}$. Observe that $\lambda<1/\beta$, and hence Π_β is a well-defined partition of $\mathbf{J}$. Using this job partition, we construct an auxiliary problem $\mathbf{P}_\beta^{pdt}$ with task processing times $(\hat{a}_j,\hat{b}_j)$, $j=1,2,\ldots,n$, defined as follows:

$$\begin{cases}\hat{a}_j=a_j,\\ \hat{b}_j=(r-1)\lambda a_j,\end{cases} \qquad \text{if } J_j\in S_r^a \qquad (r=1,2,\ldots,\beta);$$
$$\begin{cases}\hat{a}_j=(r-1)\lambda b_j,\\ \hat{b}_j=b_j,\end{cases} \qquad \text{if } J_j\in S_r^b \qquad (r=1,2,\ldots,\beta);$$
$$\hat{a}_j=\hat{b}_j=\min\{a_j,b_j\}, \qquad \text{if } J_j\in S^c.$$

Note that if $J_j\in S_r^b$ then $\hat{a}_j$ differs from a_j by at most λb_j, and if $J_j\in S^c$ then $\hat{a}_j$ differs from a_j by at most $(\frac{1}{\beta\lambda}-1)b_j$. Also, if $J_j\in S_r^a$ then $\hat{b}_j$ differs from b_j by at most λa_j, and if $J_j\in S^c$ then $\hat{b}_j$ differs from b_j by at most $(\frac{1}{\beta\lambda}-1)a_j$. For any given value of β, the value of λ has been selected in such a way that $\max\{\lambda,\frac{1}{\beta\lambda}-1\}$ is minimized so that the error of the heuristic solution is as small as possible.

Clearly, problem $\mathbf{P}^{pdt}$ satisfies condition $(C1)$ with $\ell=2\beta+1$. Hence, we can obtain an optimal schedule for $\mathbf{P}_\beta^{pdt}$ by the above dynamic program. We take the job sequence of this schedule and use

it as a heuristic solution to the original problem $\mathbf{P}^{pdt}$. We denote this heuristic as $H1$ and let its solution be σ_β^{H1}. The computational complexity of $H1$ is $O(n^{2\beta+1})$. The following theorem provides a worst-case error bound for this heuristic.

Theorem 1 $[Z(\sigma_\beta^{H1}) - Z(\sigma^*(\mathbf{P}^{pdt}))]/Z(\sigma^*(\mathbf{P}^{pdt})) \leq \frac{1}{2}\sqrt{\frac{\beta+4}{\beta}} - \frac{1}{2}$ *for* $\beta = 1, 2, \ldots$.

We now present an alternative heuristic for problem $\mathbf{P}^{pdt}$. Let $\bar{\lambda} = 1/\beta$. Define:

$$\bar{S}_r^a = \left\{ J_j \;\middle|\; (r-1)\bar{\lambda}a_j \leq b_j < r\bar{\lambda}a_j \right\} \qquad (r = 1, 2, \ldots, \beta);$$

$$\bar{S}_r^b = \left\{ J_j \;\middle|\; (r-1)\bar{\lambda}b_j \leq a_j < r\bar{\lambda}b_j \right\} \qquad (r = 1, 2, \ldots, \beta-1);$$

$$\bar{S}_\beta^b = \left\{ J_j \;\middle|\; (\beta-1)\bar{\lambda}b_j \leq a_j \leq b_j \right\}.$$

Let $\bar{\Pi}_\beta = \{\bar{S}_1^a, \bar{S}_2^a, \ldots, \bar{S}_\beta^a, \bar{S}_1^b, \bar{S}_2^b, \ldots, \bar{S}_\beta^b\}$, which is a partition of $\mathbf{J}$. Using this job partition, we construct an auxiliary problem $\bar{\mathbf{P}}_\beta^{pdt}$ with task processing times $(\hat{a}_j, \hat{b}_j)$, $j = 1, 2, \ldots, n$, defined as follows:

$$\begin{cases} \hat{a}_j = a_j, \\ \hat{b}_j = (r-1)\bar{\lambda}a_j, \end{cases} \qquad \text{if } J_j \in \bar{S}_r^a \qquad (r = 1, 2, \ldots, \beta);$$

$$\begin{cases} \hat{a}_j = (r-1)\bar{\lambda}b_j, \\ \hat{b}_j = b_j, \end{cases} \qquad \text{if } J_j \in \bar{S}_r^b \qquad (r = 1, 2, \ldots, \beta).$$

Clearly, this auxiliary problem satisfies condition $(C1)$ with $\ell = 2\beta$. Hence, we can obtain an optimal schedule for $\bar{\mathbf{P}}_\beta^{pdt}$ by the above dynamic program. We take the job sequence of this schedule and use it as a heuristic solution to the original problem $\mathbf{P}^{pdt}$. We denote this heuristic as $H2$ and let its solution be σ_β^{H2}. The computational complexity of $H2$ is $O(n^{2\beta})$. The following theorem provides a worst-case error bound for heuristic $H2$.

Theorem 2 $[Z(\sigma_\beta^{H2}) - Z(\sigma^*(\mathbf{P}^{pdt}))]/Z(\sigma^*(\mathbf{P}^{pdt})) \leq \frac{1}{\beta}$ *for* $\beta = 1, 2, \ldots$.

From Theorems **??** and **??**, it is easy to see that $H1$ and $H2$ are both polynomial-time approximation schemes for $\mathbf{P}^{pdt}$. Table 1 shows the results of these two theorems for different values of β.

Table 1. Approximation schemes for problem $\mathbf{P}^{pdt}$.

β	Heuristic $H1$		Heuristic $H2$	
	Error Bound	Running Time	Error Bound	Running Time
1	61.8%	$O(n^3)^{\dagger}$	100.0%	$O(n^2)^{\ddagger}$
2	36.6%	$O(n^5)$	50.0%	$O(n^4)$
3	26.4%	$O(n^7)$	33.3%	$O(n^6)$
4	20.7%	$O(n^9)$	25.0%	$O(n^8)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

†This running time is improved to $O(n^2)$ in Section **??**.
‡This running time is improved to $O(n \log n)$ in Section **??**.

2.2 Efficient Heuristics

When $\beta = 1$, heuristic $H1$ has a running time of $O(n^3)$ and an error bound of 61.8%, while heuristic $H2$ has a running time of $O(n^2)$ and an error bound of 100%. In this subsection, we show that the auxiliary problems $\mathbf{P}_1^{pdt}$ and $\bar{\mathbf{P}}_1^{pdt}$ can be solved optimally in $O(n^2)$ and $O(n \log n)$ time, respectively. This provides us with more efficient heuristics, namely heuristics $H3$ and $H4$, for the production scheduling subproblem with error bounds of 61.8% and 100%, respectively.

We first consider $\mathbf{P}_1^{pdt}$, which has the following task processing times:

$$(\hat{a}_j, \hat{b}_j) = \begin{cases} (a_j, 0), & \text{if } J_j \in S_1^a; \\ (0, b_j), & \text{if } J_j \in S_1^b; \\ (c_j, c_j), & \text{if } J_j \in S^c; \end{cases}$$

where $c_j = \min\{a_j, b_j\}$. Let $N_a = |S_1^a|$, $N_b = |S_1^b|$, and $N_c = |S^c|$. Denote $S_1^a = \{J_1^a, J_2^a, \ldots, J_{N_a}^a\}$, $S_1^b = \{J_1^b, J_2^b, \ldots, J_{N_b}^b\}$, and $S^c = \{J_1^c, J_2^c, \ldots, J_{N_c}^c\}$. Let the processing times of J_j^a, J_j^b, and J_j^c be $(\tilde{a}_j, 0)$, $(0, \tilde{b}_j)$, and $(\tilde{c}_j, \tilde{c}_j)$, respectively. We assume that the jobs in S_1^a, S_1^b, and S^c are indexed in such a way that $\tilde{a}_1 \leq \tilde{a}_2 \leq \cdots \leq \tilde{a}_{N_a}$, $\tilde{b}_1 \leq \tilde{b}_2 \leq \cdots \leq \tilde{b}_{N_b}$, and $\tilde{c}_1 \leq \tilde{c}_2 \leq \cdots \leq \tilde{c}_{N_c}$. Clearly, in the optimal solution of $\mathbf{P}_1^{pdt}$, the jobs in each of the subsets S_1^a, S_1^b, and S^c are processed in the Shortest Processing Time first (SPT) order. Hence, we first arrange the jobs in each of S_1^a, S_1^b, and S^c in SPT order. This requires a computational time of $O(n \log n)$. We have the following lemma.

Lemma 2 *There exists an optimal solution to* $\mathbf{P}_1^{pdt}$ *in which* J_j^a *precedes* J_k^b *if and only if* $\sum_{r=1}^{j} \tilde{a}_r \leq \sum_{r=1}^{k} \tilde{b}_r$, *for* $j = 1, 2, \ldots, N_a$ *and* $k = 1, 2, \ldots, N_b$.

Using Lemma **??**, we can identify the optimal processing sequence for the jobs in $S_1^a \cup S_1^b$ (i.e., first identify the last job, then the second last job, and so on), and this optimal sequence is independent of the positions of those jobs in S_c. This requires $O(n)$ time. Finally, we merge this job sequence with the sequence of jobs in S_c. This is done by a dynamic program similar to the one presented in subsection **??**, which requires a running time of $O(n^2)$. Therefore, the overall complexity of this heuristic (i.e., heuristic $H3$) is $O(n^2)$.

Next, we consider $\bar{\mathbf{P}}_1^{pdt}$, which has the following task processing times:

$$(\hat{a}_j, \hat{b}_j) = \begin{cases} (a_j, 0), & \text{if } J_j \in \bar{S}_1^a; \\ (0, b_j), & \text{if } J_j \in \bar{S}_1^b. \end{cases}$$

We first arrange the jobs in each of $\bar{S}_1^a$ and $\bar{S}_1^b$ in SPT order. Note that $\bar{\mathbf{P}}_1^{pdt}$ is a special case of $\mathbf{P}_1^{pdt}$ with $S_c = \emptyset$. Thus, we can apply Lemma **??** to identify the optimal processing sequence for the jobs in $\mathbf{J}$. The running time of this heuristic (i.e., heuristic $H4$) is $O(n \log n)$.

3 The Direct Delivery Model

In this section, we consider problem $\mathbf{P}^{dir}$, that is, the case with direct deliveries. The following lemma provides some important properties of this problem. It is a straightforward extension of Lemma **??**, and therefore, its proof is omitted.

Lemma 3 *There exists an optimal solution to problem* $\mathbf{P}^{dir}$ *in which (i) there is no idle time on either machine, (ii) the job sequences on both machines are identical, (iii) if* J_j *and* J_k *are to be delivered to the same customer location, and if* ($a_j < a_k$ *and* $b_j \leq b_k$) *or* ($a_j = a_k$ *and* $b_j < b_k$), *then* J_j *precedes* J_k *in the processing sequence, and (iv) if* J_j *precedes* J_k *in the processing sequence, then* $D_j \leq D_k$, *where* D_j *is the time of departure of* J_j *from the plant location.*

It suffices to restrict our search to solutions that satisfy the properties stated in Lemma **??**. In the remainder of this section, we only consider schedules that satisfy these properties. The approximation schemes presented in Section **??** are extended below to problem $\mathbf{P}^{dir}$.

3.1 Approximation Schemes

Recall that $\mathbf{J}_i$ is the subset of jobs to be delivered to customer location i ($i = 1, 2, \ldots, h$). We now extend the notation defined in Section **??**. For $i = 1, 2, \ldots, h$, suppose the job subset $\mathbf{J}_i$ is partitioned into subsets $Q_i^{(1)}, Q_i^{(2)}, \ldots, Q_i^{(\ell_i)}$ that satisfy condition (C1). We arrange the jobs in each $Q_i^{(r)}$ in nondecreasing order of a_j and b_j, and let $J_{ij}^{(r)}$ denote the jth job in the sorted list. Let $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ denote the processing time of the a-task and b-task, respectively, of $J_{ij}^{(r)}$. Denote

$$\mathbf{J}(q_{11}, \ldots, q_{h\ell_h}) = \bigcup_{i=1}^{h} \bigcup_{r=1}^{\ell_i} \{J_{i1}^{(r)}, J_{i2}^{(r)}, \ldots, J_{iq_{ir}}^{(r)}\},$$

$$A(q_{11}, \ldots, q_{h\ell_h}) = \sum_{i=1}^{h} \sum_{r=1}^{\ell_i} \sum_{j=1}^{q_{ir}} a_{ij}^{(r)},$$

$$B(q_{11}, \ldots, q_{h\ell_h}) = \sum_{i=1}^{h} \sum_{r=1}^{\ell_i} \sum_{j=1}^{q_{ir}} b_{ij}^{(r)},$$

and

$$C(q_{11}, \ldots, q_{h\ell_h}) = \max\{A(q_{11}, \ldots, q_{h\ell_h}), B(q_{11}, \ldots, q_{h\ell_h})\}.$$

Then the following dynamic program generates an optimal schedule.

Define $F_2(q_{11}, \ldots, q_{1\ell_1}; q_{21}, \ldots, q_{2\ell_2}; \ldots; q_{h1}, \ldots, q_{h\ell_h}; i, \gamma)$ as the minimum total cost of the partial schedule that consists of jobs in $\mathbf{J}(q_{11}, \ldots, q_{h\ell_h})$ given that the last batch, which carries γ of these jobs to customer location i, is free of charge (i.e., incurs zero travel cost and zero customer waiting cost), where $q_{kr} = 0, 1, \ldots, |Q_k^{(r)}|$ ($k = 1, 2, \ldots, h$; $r = 1, 2, \ldots, \ell_k$), $i = 1, 2, \ldots, h$, and $\gamma = 0, 1, \ldots, \Gamma$. For notational convenience, we use $F_2(q_{11}, \ldots, q_{h\ell_h}; i, \gamma)$ to represent $F_2(q_{11}, \ldots, q_{1\ell_1}; q_{21}, \ldots, q_{2\ell_2}; \ldots; q_{h1}, \ldots, q_{h\ell_h}; i, \gamma)$, and we use $F_2(q_{11}, \ldots, q_{ir} - u_{ir}, \ldots, q_{h\ell_h}; i, \gamma)$ to represent the function with the same parameters, except that there is one parameter, namely q_{ir}, being replaced by $q_{ir} - u_{ir}$.

We have the following recurrence relation:

$$F_2(q_{11},\ldots,q_{h\ell_h};i,\gamma)$$
$$=\begin{cases}\min\limits_{\substack{k=1,2,\ldots,h\\ \phi=1,2,\ldots,\Gamma}}\Big\{F_2(q_{11},\ldots,q_{h\ell_h};k,\phi)+(f+g_{0k}+g_{k0})+\mu\phi\big[C(q_{11},\ldots,q_{h\ell_h})+t_{0k}\big]\Big\}, & \text{if } \gamma=0;\\ \min\limits_{\substack{r=1,2,\ldots,\ell_i \text{ s.t. } q_{ir}>0\\ u_{ir}=1,2,\ldots,\min\{q_{ir},\gamma\}}}\Big\{F_2(q_{11},\ldots,q_{ir}-u_{ir},\ldots,q_{h\ell_h};i,\gamma-u_{ir})\Big\}, & \text{if } \gamma\geq 1.\end{cases}$$

This recurrence relation is interpreted as follows: When $\gamma=0$, a customer location k and a batch size ϕ are selected for the last delivery batch of the partial schedule, where a travel cost of $f+g_{0k}+g_{k0}$ and a total customer waiting cost of $\mu\phi\big[C(q_{11},\ldots,q_{h\ell_h})+t_{0k}\big]$ for the entire batch are incurred. When $\gamma\geq 1$, u_{ir} jobs are selected from $Q_i^{(r)}$ $(r=1,2,\ldots,\ell_i)$ and assigned to the last batch in the partial schedule.

The boundary condition of this dynamic program is

$$F_2(0,\ldots,0;i,\gamma)=\begin{cases}0, & \text{if } \gamma=0;\\ +\infty, & \text{if } \gamma\geq 1;\end{cases}$$

for $i=1,2,\ldots,h$. The objective is $F_2(|Q_1^{(1)}|,\ldots,|Q_1^{(\ell_1)}|;|Q_2^{(1)}|,\ldots,|Q_2^{(\ell_2)}|;\ldots;|Q_h^{(1)}|,\ldots,|Q_h^{(\ell_h)}|;i,0)$, where i can be any customer location. The values of $C(q_{11},\ldots,q_{h\ell_h})$ can be predetermined efficiently before solving the dynamic program. Thus, the running time of the dynamic program is $O(\bar{\ell}n^{\ell_1+\cdots+\ell_h}\Gamma^2)\leq O(\bar{\ell}n^{h\bar{\ell}+2})$, where $\bar{\ell}=\max\{\ell_1,\ell_2,\ldots,\ell_h\}$.

The above dynamic program solves the direct delivery problem optimally if each $\mathbf{J}_i$ is partitioned into subsets that satisfy condition (C1). We now present a heuristic for problem $\mathbf{P}^{dir}$. For each job subset $\mathbf{J}_i$ $(i=1,2,\ldots,h)$, we apply partition Π_β and adjust the task processing times as in the auxiliary problem $\mathbf{P}_\beta^{pdt}$. Let this modified problem be $\mathbf{P}_\beta^{dir}$. We solve $\mathbf{P}_\beta^{dir}$ by the above dynamic program. We take the job sequence and the batches induced by this schedule, and use them as a heuristic solution to problem $\mathbf{P}^{dir}$. Denote this heuristic as $H5$ and its solution as σ_β^{H5}. The computational complexity of $H5$ is $O(\beta n^{h(2\beta+1)+2})$, because in this case $\ell=2\beta+1$.

An alternative heuristic can be developed similarly. For each job subset $\mathbf{J}_i$ $(i=1,2,\ldots,h)$, we apply partition $\bar{\Pi}_\beta$ and adjust the task processing times as in the auxiliary problem $\bar{\mathbf{P}}_\beta^{pdt}$. Let

this modified problem be $\bar{\mathbf{P}}_\beta^{dir}$. We solve $\bar{\mathbf{P}}_\beta^{dir}$ by the above dynamic program. We take the job sequence and the batches induced by this schedule, and use them as a heuristic solution to problem $\mathbf{P}^{dir}$. Denote this heuristic as $H6$ and its solution as σ_β^{H6}. The computational complexity of $H6$ is $O(\beta n^{h(2\beta)+2})$, because in this case $\ell = 2\beta$.

The following theorems provide worst-case error bounds for heuristics $H5$ and $H6$.

Theorem 3 $[Z(\sigma_\beta^{H5}) - Z(\sigma^*(\mathbf{P}^{dir}))]/Z(\sigma^*(\mathbf{P}^{dir})) \leq \frac{1}{2}\sqrt{\frac{\beta+4}{\beta}} - \frac{1}{2}$ *for* $\beta = 1, 2, \ldots$.

Theorem 4 $[Z(\sigma_\beta^{H6}) - Z(\sigma^*(\mathbf{P}^{dir}))]/Z(\sigma^*(\mathbf{P}^{dir})) \leq \frac{1}{\beta}$ *for* $\beta = 1, 2, \ldots$.

From Theorems **??** and **??**, it is easy to see that $H5$ and $H6$ are both polynomial-time approximation schemes for $\mathbf{P}^{dir}$. Table 2 shows the results of these two theorems for different values of β.

Table 2. Approximation schemes for problem $\mathbf{P}^{dir}$.

β	Heuristic $H5$		Heuristic $H6$	
	Error Bound	Running Time	Error Bound	Running Time
1	61.8%	$O(n^{3h+2})$	100.0%	$O(n^{2h+2})^\dagger$
2	36.6%	$O(n^{5h+2})$	50.0%	$O(n^{4h+2})$
3	26.4%	$O(n^{7h+2})$	33.3%	$O(n^{6h+2})$
4	20.7%	$O(n^{9h+2})$	25.0%	$O(n^{8h+2})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

†This running time is improved to $O(n^{h+1})$ in Section **??**.

3.2 A More Efficient Heuristic

We now develop a more efficient heuristic for problem $\mathbf{P}^{dir}$. This new heuristic has the same error bound as heuristic $H6$ when $\beta = 1$, but it has a lower computational complexity.

First, suppose we have decided the processing sequence of the jobs in set $\mathbf{J}_i$, for $i = 1, 2, \ldots, h$. Then the following dynamic program will determine the optimal processing sequence of jobs

$J_1, J_2, \ldots, J_n$. (In other words, this dynamic program will merge different job sequences optimally.) Let a'_{ij} and b'_{ij} denote the processing time of the a-task and b-task, respectively, of the jth job in set $\mathbf{J}_i$. Denote $A'(q_1, \ldots, q_h) = \sum_{i=1}^{h} \sum_{j=1}^{q_r} a'_{ij}$, $B'(q_1, \ldots, q_h) = \sum_{i=1}^{h} \sum_{j=1}^{q_r} b'_{ij}$, and $C'(q_1, \ldots, q_h) = \max\{A'(q_1, \ldots, q_h), B'(q_1, \ldots, q_h)\}$. Define $F_3(q_1, q_2, \ldots, q_h)$ as the minimum total cost of the partial schedule that consists of the first q_i jobs from the processing sequence of $\mathbf{J}_i$ $(i = 1, 2, \ldots, h)$. Then

$$F_3(q_1, q_2, \ldots, q_h) = \min_{\substack{i=1,2,\ldots,h \text{ s.t. } q_i>0 \\ u_i=1,2,\ldots,\max\{q_i,\Gamma\}}} \Big\{ F_3(q_1, \ldots, q_{i-1}, q_i - u_i, q_{i+1}, \ldots, q_h) + (f + g_{0i} + g_{i0}) + \mu u_i \big[C'(q_1, \ldots, q_h) + t_{0i} \big] \Big\}.$$

In this recurrence relation, u_i denotes the number of jobs selected for the last delivery batch in the partial schedule, and this batch is delivered to customer location i. The boundary condition is $F_3(0, 0, \ldots, 0) = 0$, and the objective is $F_3(n_1, n_2, \ldots, n_h)$. Next, we present our new heuristic.

Heuristic $H7$:

Step 1. Let $p_j = \frac{1}{2}(a_j + b_j)$ $(j = 1, 2, \ldots, n)$. For $i = 1, 2, \ldots, h$, arrange the elements of $\mathbf{J}_i$ in nondecreasing order of p_j.

Step 2. Apply the above dynamic program to determine the overall job processing sequence and to assign the jobs to batches.

The running time of heuristic $H7$ is $O(n^h \Gamma) \leq O(n^{h+1})$. Let σ^{H7} denote the schedule generated by heuristic $H7$. We now analyze its effectiveness. We construct an auxiliary problem $\mathbf{P'}$ with task processing times (p_j, p_j), $j = 1, 2, \ldots, n$. By property (iii) of Lemma **??**, in an optimal schedule of problem $\mathbf{P'}$, the elements of each $\mathbf{J}_i$ must be arranged in nondecreasing order of p_j. Hence, an optimal schedule of $\mathbf{P'}$ can be obtained by heuristic $H7$. Therefore, $\sigma^*(\mathbf{P'})$ and σ^{H7} have the same job sequence, the same assignment of jobs to batches, and the same delivery cost.

Lemma 4 $Z(\sigma^*(\mathbf{P'})) \leq Z(\sigma^*(\mathbf{P}^{dir}))$.

Lemma **??** provides us with a lower bound on the optimal solution value of problem $\mathbf{P}^{dir}$. The next theorem provides a worst-case error bound for heuristic $H7$.

Theorem 5 $[Z(\sigma^{H7}) - Z(\sigma^*(\mathbf{P}^{dir}))]/Z(\sigma^*(\mathbf{P}^{dir})) \leq 1$.

4 The General Problem

In this section, we consider problem $\mathbf{P}^{gen}$ where milk-run deliveries are allowed. The following lemma provides some important properties of this problem. It is a straightforward extension of Lemmas **??** and **??**, and therefore, its proof is omitted.

Lemma 5 *There exists an optimal solution to problem $\mathbf{P}^{gen}$ in which (i) there is no idle time on either machine, (ii) the job sequences on both machines are identical, (iii) if J_j and J_k are to be delivered to the same customer location, and if ($a_j < a_k$ and $b_j \leq b_k$) or ($a_j = a_k$ and $b_j < b_k$), then J_j precedes J_k in the processing sequence, (iv) if J_j precedes J_k in the processing sequence, then $D_j \leq D_k$, and (v) if J_j precedes J_k in the processing sequence, and if J_j and J_k belong to the same delivery batch, then J_j arrives at its customer location no later than J_k.*

In the remainder of this section, we only consider schedules that satisfy properties (i)–(v) in Lemma **??**. Heuristics $H5$, $H6$, and $H7$ can all be modified to solve $\mathbf{P}^{gen}$. The dynamic program in Section **??** can be modified as follows. Define $F_4(q_{11}, \ldots, q_{1\ell_1}; q_{21}, \ldots, q_{2\ell_2}; \ldots; q_{h1}, \ldots, q_{h\ell_h}; i, \gamma, \psi)$ as the minimum total cost of the partial schedule that consists of jobs in $\mathbf{J}(q_{11}, \ldots, q_{h\ell_h})$ given that: (i) The last delivery batch must include exactly γ jobs from $\mathbf{J}(q_{11}, \ldots, q_{h\ell_h})$. (ii) The last batch is free of charge (i.e., incurs zero travel cost and zero customer waiting cost) if it is delivered to customer location i. (iii) Another $\psi - \gamma$ jobs from $\mathbf{J} \setminus \mathbf{J}(q_{11}, \ldots, q_{h\ell_h})$ are already in that batch; these $\psi - \gamma$ jobs can be delivered to their customer locations at no cost (i.e., no travel cost and no customer waiting cost) provided that their first stop of delivery is location i.

Function F_4 is defined for $q_{kr} = 0, 1, \ldots, |Q_k^{(r)}|$ $(k = 1, 2, \ldots, h;\ r = 1, 2, \ldots, \ell_k)$; $i = 1, 2, \ldots, h$; $\gamma = 0, 1, \ldots, \psi$; and $\psi = 1, 2, \ldots, \Gamma$. Note that in the definition of F_4, we are allowed to select γ jobs that belong to different customer locations and assign them to the last delivery batch. However, if some of these γ jobs belong to customer locations other than i, then a cost adjustment is required. For example, if all of these γ jobs belong to a customer at location k $(k \neq i)$, then an additional travel cost of $g_{0k}+g_{ki}-g_{0i}$ and an additional customer waiting cost of $\mu\big[\gamma(t_{0k}-t_{0i})+(\psi-\gamma)(t_{0k}+t_{ki}-t_{0i})\big]$ are incurred. The increase in travel cost of $g_{0k}+g_{ki}-g_{0i}$ is due to the rerouting of the vehicle from the plant location through location k to location i (instead of traveling directly from the plant location to location i). The increase in customer waiting cost is due to the following: (i) The γ jobs now have a delivery time of t_{0k} instead of t_{0i}. (ii) It now takes $t_{0k} + t_{ki}$ units of time (instead of t_{0i} units of time) for the $\psi - \gamma$ jobs to arrive at customer location i.

We have the following recurrence relation:

$$
\begin{aligned}
&F_4(q_{11}, \ldots, q_{h\ell_h}; i, \gamma, \psi) \\
&= \begin{cases}
\min\limits_{\substack{k=1,2,\ldots,h \\ \phi=1,2,\ldots,\Gamma}} \Big\{ F_4(q_{11}, \ldots, q_{h\ell_h}; k, \phi, \phi) \\
\qquad\qquad + (f + g_{0k} + g_{k0}) + \mu\phi\big[C(q_{11}, \ldots, q_{h\ell_h}) + t_{0k}\big] \Big\}, & \text{if } \gamma = 0; \\
\min\Bigg\{ \min\limits_{\substack{r=1,2,\ldots,\ell_i \text{ s.t. } q_{ir}>0 \\ u_{ir}=1,2,\ldots,\min\{q_{ir},\gamma\}}} \Big\{ F_4(q_{11}, \ldots, q_{ir} - u_{ir}, \ldots, q_{h\ell_h}; i, \gamma - u_{ir}, \psi) \Big\}, \\
\qquad \min\limits_{\substack{k\in\{1,2,\ldots,h\}\setminus\{i\};\ r=1,2,\ldots,\ell_k \text{ s.t. } q_{kr}>0 \\ u_{kr}=1,2,\ldots,\min\{q_{kr},\gamma\}}} \Big\{ F_4(q_{11}, \ldots, q_{kr} - u_{kr}, \ldots, q_{h\ell_h}; k, \gamma - u_{kr}, \psi) \\
\qquad\qquad + (g_{0k} + g_{ki} - g_{0i}) + \mu\big[\psi(t_{0k} + t_{ki} - t_{0i}) - \gamma t_{ki}\big] \Big\} \Bigg\}, & \text{if } \gamma \geq 1.
\end{cases}
\end{aligned}
$$

This recurrence relation is interpreted as follows: When $\gamma = 0$, a customer location k and a batch size ϕ are selected for the last delivery batch of the partial schedule, where no other job has yet occupied the batch. In such a case, a new vehicle trip is created. As a result, a travel cost of $f + g_{0k} + g_{k0}$ and a total customer waiting cost of $\mu\phi\big[C(q_{11}, \ldots, q_{h\ell_h}) + t_{0k}\big]$ for the entire batch are incurred. This is the same argument as in the dynamic program for the direct delivery model. When $\gamma \geq 1$, there are two major choices. The first choice is to select $u_{ir} > 0$ jobs from $Q_i^{(r)}$ and assign them to the last

delivery batch in the partial schedule, which will be delivered to customer location i by the vehicle at no cost. The second choice is to select jobs for a different customer location k. This will increase the travel cost of the vehicle by $g_{0k} + g_{ki} - g_{0i}$, and it will increase the customer waiting cost by $\mu\big[\gamma(t_{0k} - t_{0i}) + (\psi - \gamma)(t_{0k} + t_{ki} - t_{0i})\big] = \mu\big[\psi(t_{0k} + t_{ki} - t_{0i}) - \gamma t_{ki}\big]$.

The boundary condition of this dynamic program is

$$F_4(0, \ldots, 0; i, \gamma, \psi) = \begin{cases} 0, & \text{if } \gamma = 0; \\ +\infty, & \text{if } \gamma \geq 1; \end{cases}$$

for $i = 1, 2, \ldots, h$ and $\psi = 1, 2, \ldots, \Gamma$. The objective is $F_4(|Q_1^{(1)}|, \ldots, |Q_1^{(\ell_1)}|; |Q_2^{(1)}|, \ldots, |Q_2^{(\ell_2)}|; \ldots; |Q_h^{(1)}|, \ldots, |Q_h^{(\ell_h)}|; i, 0, \Gamma)$, where i can be any customer location. The running time of the dynamic program is $O(\bar{\ell} n^{\ell_1 + \cdots + \ell_h} \Gamma^3) \leq O(\bar{\ell} n^{h\bar{\ell}+3})$, where $\bar{\ell} = \max\{\ell_1, \ell_2, \ldots, \ell_h\}$. Hence, the modified heuristic $H5$ (denoted as $H5'$) has a running time of $O(\beta n^{h(2\beta+1)+3})$, and the modified heuristic $H6$ (denoted as $H6'$) has a running time of $O(\beta n^{h(2\beta)+3})$.

Remark 1 *In the above dynamic program, function F_4 is defined carefully such that the recursive calculations can be performed efficiently. The recurrence relation in the dynamic program requires only the selection of u_{kr} jobs at a time. On the other hand, Chen and Vairaktarakis (2005) have developed several dynamic programs for various versions of their integrated production and distribution model. In each iteration of their dynamic programs, an exhaustive enumeration of all possible job combinations and delivery routes for a delivery batch is performed. This results in very high computational complexities for their dynamic programs. The technique employed in our dynamic program can be applied to Chen and Vairaktarakis's dynamic programs to avoid the enumeration of all possible routes. For example, in their algorithm DP2, the running time can be reduced from $O(c^k n^k k^{k-1})$ to $O(c^3 n^k k^2)$ if such a technique is used.*

It is easy to see that Theorems **??** and **??** remain valid for these modified heuristics. Therefore, polynomial-time approximation schemes exist for problem $\mathbf{P}^{gen}$. To modify heuristic $H7$ for solving $\mathbf{P}^{gen}$, we can use the above dynamic program to perform Step 2 of the heuristic by setting $\ell_1 = \ell_2 =$

$\cdots = \ell_h = 1$. As a result, the modified heuristic (denoted as $H7'$) has a running time of $O(n^{h+3})$, and Theorem **??** remains valid for this modified heuristic.

Table 3 summarizes the results of the analysis of problem $\mathbf{P}^{gen}$.

Table 3. Approximation schemes for problem $\mathbf{P}^{gen}$.

β	Heuristic $H5'$		Heuristic $H6'$	
	Error Bound	Running Time	Error Bound	Running Time
1	61.8%	$O(n^{3h+3})$	100.0%	$O(n^{2h+3})^{\dagger}$
2	36.6%	$O(n^{5h+3})$	50.0%	$O(n^{4h+3})$
3	26.4%	$O(n^{7h+3})$	33.3%	$O(n^{6h+3})$
4	20.7%	$O(n^{9h+3})$	25.0%	$O(n^{8h+3})$
⋮	⋮	⋮	⋮	⋮

†This running time is improved to $O(n^{h+3})$ by heuristic $H7'$.

5 Computational Experiments

In this section, we report computational experiments conducted using randomly generated problems. We focus on the testing of the performance of heuristics $H1$ and $H2$ for problem $\mathbf{P}^{pdt}$, heuristics $H5$ and $H7$ with $\beta = 1$ for problem $\mathbf{P}^{dir}$, and heuristics $H5'$ and $H7'$ with $\beta = 1$ for problem $\mathbf{P}^{gen}$. We run experiments with n = 20, 40, and 80 jobs, h = 3, 4, and 5 customer locations, and Γ = 4, 8, and 12 for the vehicle capacity. The task processing times a_j and b_j are integers uniformly generated from the range $[1, 100]$. The unit cost of waiting, μ, takes on the values 0.25, 1, and 4. The machines are located at the center of a square with width w equal to either 100 or 200 units, and the h customer locations are uniformly and independently located within the square. In all test instances, the number of jobs per customer location is balanced, that is, this number does not differ by more than 1 for the h locations. The speed of the vehicle is set equal to 1.

Our fixed and variable transportation costs are generated similar to Chen and Vairaktarakis (2005). We use a scaling parameter ρ that captures the relative importance of the customer waiting cost in comparison to the transportation cost. Parameter ρ is selected in such a way that when

$\mu = 1$ and all delivery batches are full, the total waiting cost is approximately equal to the total transportation cost . When the waiting and transportation costs are equalized in this fashion, the optimal solution is neither driven primarily by the schedule which is optimal for the production part alone, nor by the schedule which is optimal for the transportation part alone. Rather, the optimal solution is a coordination between the two and the resulting problem instances are harder. For a given value of ρ, the fixed charge f is an integer drawn from the uniform distribution $\mathcal{U}[50\rho, 250\rho]$, while travel costs g_{ij} are integers from $\mathcal{U}[0.8t_{ij}\rho, 1.2t_{ij}\rho]$, where t_{ij} is the Euclidean distance between locations i and j.

Note that the average task processing time is approximately 50, the average workload on each machine is approximately $50n$, and the average task completion time is approximately $25n$. Hence, for direct deliveries, the average customer waiting cost is approximately $25n + \frac{1}{h}\sum_{i=1}^{h} t_{0i}$ when $\mu = 1$. On the other hand, the average transportation cost associated with a batch delivered to customer i is $150\rho + t_{0i}\rho$. Thus, the average transportation cost of a delivery batch is approximately $\frac{1}{h}\sum_{i=1}^{h}(150\rho + t_{0i}\rho) = \rho(150 + \frac{1}{h}\sum_{i=1}^{h} t_{0i})$. When all delivery batches are full, the total number of deliveries is $\lceil n/\Gamma \rceil$. Therefore, the total customer waiting cost is approximately $n(25n + \frac{1}{h}\sum_{i=1}^{h} t_{0i})$, and the total transportation cost is approximately $\frac{n\rho}{\Gamma}(150 + \frac{1}{h}\sum_{i=1}^{h} t_{0i})$. We select the value of ρ such that these two quantities are equal. Thus, we use the following value of ρ for the direct delivery model:

$$\rho^{dir} = \frac{n(25n + \frac{1}{h}\sum_{i=1}^{h} t_{0i})}{\frac{n}{\Gamma}(150 + \frac{1}{h}\sum_{i=1}^{h} t_{0i})} = \frac{\Gamma(25nh + \sum_{i=1}^{h} t_{0i})}{150h + \sum_{i=1}^{h} t_{0i}}.$$

The ρ value for milk-run deliveries is motivated similarly. The only difference is in approximating the average delivery time and cost. Recall that in our experiments the plant is located at the center of a square with width w. We approximate the length of a delivery tour by the circumference of a circle located inside the square. The circumference of the circle with diameter $w/2$ centered at the plant is $w\pi/2$, which is close to $1.5w$. Hence, the total customer waiting cost is approximately $n(25n + 0.75w)$, and the total transportation cost is approximately $\frac{n\rho}{\Gamma}(150 + 1.5w)$. Therefore, we

use the following value of ρ for the general model:

$$\rho^{gen} = \frac{n(25n + 0.75w)}{\frac{n}{\Gamma}(150 + 1.5w)} = \frac{\Gamma(25n + 0.75w)}{150 + 1.5w}.$$

For each combination of parameters, we generate 10 random problem instances. For each instance, we compute Z^{H1}, Z^{H2}, Z^{H5}, Z^{H7}, $Z^{H5'}$, $Z^{H7'}$, LB^{pdt}, LB^{dir}, and LB^{gen}, where Z^H denotes the solution generated by heuristic H, and $LB^{pdt}, LB^{dir}, LB^{gen}$ denote the lower bounds on the optimal solution values of $\mathbf{P}^{pdt}, \mathbf{P}^{dir}, \mathbf{P}^{gen}$, respectively. We set $LB^{pdt} = \max\left\{Z(\sigma^*(\mathbf{P}_1^{pdt})), Z(\sigma^*(\bar{\mathbf{P}}_1^{pdt}))\right\}$, where $\mathbf{P}_1^{pdt}$ and $\bar{\mathbf{P}}_1^{pdt}$ are the auxiliary problems in heuristics $H1$ and $H2$, respectively, with $\beta = 1$. The values of LB^{dir} and LB^{gen} are obtained similarly. Denote $LB^{H1} = LB^{H2} = LB^{pdt}$; $LB^{H5} = LB^{H7} = LB^{dir}$; and $LB^{H5'} = LB^{H7'} = LB^{gen}$. For each heuristic H, we let $e = \left[(Z^H/LB^H) - 1\right] \times 100\%$, which is an estimate of the relative error of the heuristic solution. For each combination of parameters, we calculate the average value of e for each heuristic. For each of the 9 possible combinations of n and h, we compute the overall average value of e for the 180 problem instances (there are 18 combinations of Γ, μ, w and 10 random instances per combination of parameters).

We coded the algorithms in C++ and ran the experiments on a 2.8 GHz computer. We allowed a maximum of 2 minutes per problem instance, within which we were not able to solve many of the problem instances for $n = 80$ and $k = 5$. The CPU time for the remaining problems ranges from a few seconds to a little over a minute. Table 4 summarizes the results of the computational study.

From the computational results, we observe that the average relative errors (i.e., e) are small enough to render both algorithms effective. The relative errors for $H1$, $H5$, and $H5'$ are uniformly smaller than those obtained by $H2$, $H7$, and $H7'$, respectively. This validates empirically the theoretical bounds obtained for these heuristics when $\beta = 1$. We also observe that the performance of these heuristics improves as n increases, and for those problems with delivery considerations, the performance of the heuristics deteriorates as k increases. In addition, for all six heuristics tested, the relative errors increase in the order of $\mathbf{P}^{pdt}, \mathbf{P}^{dir}, \mathbf{P}^{gen}$. This is because in problem $\mathbf{P}^{pdt}$, the heuristic suboptimality is due to the production part only. For the other two problems, the heuristics intro-

duce suboptimality in both the production and the distribution parts, and hence $\mathbf{P}^{dir}$ and $\mathbf{P}^{gen}$ are expected to exhibit larger heuristic errors than $\mathbf{P}^{pdt}$. Further, the suboptimality introduced in $\mathbf{P}^{gen}$ is greater than that in $\mathbf{P}^{dir}$, because the optimization of the associated delivery tours does not exist in direct deliveries.

Table 4. Computational results (average values of e).

	$k=3$			$k=4$			$k=5$		
	$\mathbf{P}^{pdt}$	$\mathbf{P}^{dir}$	$\mathbf{P}^{gen}$	$\mathbf{P}^{pdt}$	$\mathbf{P}^{dir}$	$\mathbf{P}^{gen}$	$\mathbf{P}^{pdt}$	$\mathbf{P}^{dir}$	$\mathbf{P}^{gen}$
	$H1$	$H5$	$H5'$	$H1$	$H5$	$H5'$	$H1$	$H5$	$H5'$
$n=20$	1.3%	1.6%	2.4%	1.6%	2.1%	3.5%	1.6%	2.2%	4.1%
$n=40$	1.1%	1.3%	2.1%	1.2%	1.9%	3.1%	1.3%	2.0%	3.8%
$n=80$	1.0%	1.1%	2.0%	1.0%	1.3%	3.0%	-	-	-
	$H2$	$H7$	$H7'$	$H2$	$H7$	$H7'$	$H2$	$H7$	$H7'$
$n=20$	1.7%	2.3%	3.5%	1.8%	2.2%	4.8%	1.7%	2.4%	5.3%
$n=40$	1.3%	2.2%	3.0%	1.2%	2.0%	4.0%	1.3%	2.0%	4.7%
$n=80$	1.0%	1.4%	2.9%	1.0%	1.6%	3.6%	-	-	-

6 Conclusions

We have developed a polynomial-time approximation scheme for the two-machine scheduling problem with bundling operations to minimize the total flow-time of jobs. We have also generalized this polynomial-time approximation scheme to the model with deliveries to remote customers so as to minimize the sum of transportation cost and customers' waiting cost. Our analysis covers the case with direct shipments as well as the general case with milk-run deliveries. We have also developed more efficient heuristics with constant worst-case error bounds for these problems.

Some future research directions on this topic are of interest. First, it is interesting to extend the model and its analysis to more than two machines with bundling operations and delivery considerations. Second, this paper considers the case where the two machines are located at the same place.

If the machines are located at different locations, the delivery plan should include the possibility of having the same vehicle pick up the two finished tasks of the same job and deliver them to the customers. Such an extension is also an interesting topic for future research.

Acknowledgments

The authors would like thank the Associate Editor and two referees for their helpful comments and suggestions. This research was supported in part by the Research Grants Council of Hong Kong under grant number PolyU6132/02E. Part of George Vairaktarakis' work was performed at the Department of Logistics at The Hong Kong Polytechnic University.

References

Ahmadi, R., U. Bagchi and T.A. Roemer (2005). Coordinated scheduling of customer orders for quick response. *Naval Research Logistics,* **52,** 493–512.

Blocher, J.D. and D. Chhajed (1996). The customer order lead time problem on parallel machines. *Naval Research Logistics,* **43,** 629–654.

Blocher, J.D., D. Chhajed and M. Leung (1998). Customer order scheduling in a general job shop environment. *Decision Sciences,* **29,** 951–981.

Cai, X. and X. Zhou (2004). Deterministic and stochastic scheduling with teamwork tasks. *Naval Research Logistics,* **51,** 818–840.

Chang, Y.-C. and C.-Y. Lee (2004). Machine scheduling with job delivery coordination. *European Journal of Operational Research,* **158,** 470–487.

Chen, Z.-L. (2004). Integrated production and distribution operations: Taxonomy, models, and

review. D. Simchi-Levi, D. Wu and Z.-J. Shen (eds.), *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era.* Kluwer Academic Publishers, Boston, MA.

Chen, Z.-L. and G. Pundoor (2006). Order assignment and scheduling in a supply chain. *Operations Research,* forthcoming.

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

Fumero, F. and C. Vercellis (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation Science,* **33,** 330–340.

Geismar, H.N., G. Laporte, L. Lei and C. Sriskandarajah (2005). The integrated production and transportation scheduling problem for a product with a short life span and non-instantaneous transportation time. Working paper.

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

Julien, F.M. and M.J. Magazine (1990). Scheduling customer orders: An alternative production scheduling approach. *Journal of Manufacturing and Operations Management,* **3,** 177–199.

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

Lee, C.-Y., T.C.E. Cheng and B.M.T. Lin (1993). Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem. *Management Science,* **39,** 616–625.

Leung, J.Y.-T., H. Li and M. Pinedo (2005a). Order scheduling in an environment with dedicated resources in parallel. *Journal of Scheduling,* **8,** 355–386.

Leung, J.Y.-T., H. Li and M. Pinedo (2005b). Scheduling orders for multiple product types to minimize total weighted completion time. Working paper.

Leung, J.Y.-T., H. Li, M. Pinedo and C. Sriskandarajah (2005c). Open shops with jobs overlap—revisited. *European Journal of Operational Research,* **163,** 569–571.

Leung, J.Y.-T., H. Li, M. Pinedo and J. Zhang (2005d). Minimizing total weighted completion time when scheduling orders in a flexible environment with uniform machines. Working paper.

Leung, J.Y.-T., H. Li and M. Pinedo (2006a). Approximation algorithms for minimizing total weighted completion time of orders on identical machines in parallel. *Naval Research Logistics,* forthcoming.

Leung, J.Y.-T., H. Li and M. Pinedo (2006b). Scheduling orders for multiple product types with due date related objectives. *European Journal of Operational Research,* **168,** 370–389.

Li, C.-L. and J. Ou (2005). Machine scheduling with pickup and delivery. *Naval Research Logistics,* **52,** 617–630.

Li, C.-L., G. Vairaktarakis and C.-Y. Lee (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research,* **164,** 39–51.

Pundoor, G. and Z.-L. Chen (2005). Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost. *Naval Research Logistics,* **52,** 571–589.

Sarmiento, A.M. and R. Nagi (1999). A review of integrated analysis of production-distribution systems. *IIE Transactions,* **31,** 1061–1074.

Sung, C.S. and S.H. Yoon (1998). Minimizing total weighted completion time at a pre-assembly stage composed of two feeding machines. *International Journal of Production Economics,* **54,** 247–255.

Wagneur, E. and C. Sriskandarajah (1993). Open shops with jobs overlap. *European Journal of Operational Research,* **71,** 366–378.

Yang, J. (2003). Scheduling parallel machines for the customer order problem with fixed batch sequence. *Journal of the Korean Institute of Industrial Engineers,* **29,** 304–311.

Yang, J. (2005). The complexity of customer order scheduling problems on parallel machines. *Computers and Operations Research,* **32,** 1921–1939.

Yang, J. and M.E. Posner (2005). Scheduling parallel machines for the customer order problem. *Journal of Scheduling,* **8,** 49–74.

Appendix

Proof of Lemma ??: The proof of property (i) is straightforward and is omitted. See Sung and Yoon (1998) for proofs of properties (ii) and (iii). ∎

Proof of Theorem ??: Consider the difference between schedules $\sigma^*(\mathbf{P}_\beta^{pdt})$ and σ_β^{H1}. Let Δ' denote the difference in total flow time of the a-tasks between these two schedules. Note that (i) $\hat{a}_j = a_j$ if $J_j \in S_1^a \cup S_2^a \cup \cdots \cup S_\beta^a$, (ii) $\hat{a}_j > a_j - \lambda b_j$ if $J_j \in S_1^b \cup S_2^b \cup \cdots \cup S_\beta^b$, and (iii) $\hat{a}_j \geq a_j - (\frac{1}{\beta\lambda} - 1)b_j$ if $J_j \in S^c$. Thus,

$$\Delta' \leq \sum_{\substack{j=1,\ldots,n \text{ s.t.} \\ J_j \in S_1^b \cup \cdots \cup S_\beta^b}} (n - \xi_j + 1)\lambda b_j + \sum_{\substack{j=1,\ldots,n \\ \text{s.t. } J_j \in S^c}} (n - \xi_j + 1)\Big(\frac{1}{\beta\lambda} - 1\Big)b_j,$$

where ξ_j is the position of J_j in $\sigma^*(\mathbf{P}_\beta^{pdt})$. Similarly, let Δ'' denote the difference in total flow time of the b-tasks between the two schedules. We have

$$\Delta'' \leq \sum_{\substack{j=1,\ldots,n \text{ s.t.} \\ J_j \in S_1^a \cup \cdots \cup S_\beta^a}} (n - \xi_j + 1)\lambda a_j + \sum_{\substack{j=1,\ldots,n \\ \text{s.t. } J_j \in S^c}} (n - \xi_j + 1)\Big(\frac{1}{\beta\lambda} - 1\Big)a_j.$$

It is easy to check that $\frac{1}{\beta\lambda} - 1 = \lambda$. Hence,

$$\begin{aligned} Z(\sigma_\beta^{H1}) - Z(\sigma^*(\mathbf{P}_\beta^{pdt})) &\leq \max\{\Delta', \Delta''\} \\ &\leq \lambda \cdot \max\Bigg\{ \sum_{\substack{j=1,\ldots,n \text{ s.t.} \\ J_j \in S_1^b \cup \cdots \cup S_\beta^b}} (n - \xi_j + 1) b_j + \sum_{\substack{j=1,\ldots,n \\ \text{s.t. } J_j \in S^c}} (n - \xi_j + 1) b_j, \\ &\qquad \sum_{\substack{j=1,\ldots,n \text{ s.t.} \\ J_j \in S_1^a \cup \cdots \cup S_\beta^a}} (n - \xi_j + 1) a_j + \sum_{\substack{j=1,\ldots,n \\ \text{s.t. } J_j \in S^c}} (n - \xi_j + 1) a_j \Bigg\} \\ &\leq \lambda \cdot Z(\sigma^*(\mathbf{P}_\beta^{pdt})). \end{aligned}$$

Note that $Z(\sigma^*(\mathbf{P}_\beta^{pdt})) \leq Z(\sigma^*(\mathbf{P}^{pdt}))$. Therefore, $Z(\sigma_\beta^{H1}) - Z(\sigma^*(\mathbf{P}^{pdt})) \leq \lambda Z(\sigma^*(\mathbf{P}^{pdt})) = \left(\frac{1}{2}\sqrt{\frac{\beta+4}{\beta}} - \frac{1}{2}\right) Z(\sigma^*(\mathbf{P}^{pdt}))$. ∎

Proof of Theorem ??: Similar argument as in the proof of Theorem **??**. ∎

Proof of Theorem ??: Consider the difference between schedules $\sigma^*(\mathbf{P}_\beta^{dir})$ and σ_β^{H5}. Both schedules have the same assignment of jobs to batches, and therefore, they have the same delivery cost. Using the same arguments as in the proof of Theorem **??**, we have

$$Z(\sigma_\beta^{H5}) - Z(\sigma^*(\mathbf{P}_\beta^{dir})) \leq \lambda \cdot Z(\sigma^*(\mathbf{P}_\beta^{dir})).$$

Note that $Z(\sigma^*(\mathbf{P}_\beta^{dir})) \leq Z(\sigma^*(\mathbf{P}^{dir}))$. Therefore, $Z(\sigma_\beta^{H5}) - Z(\sigma^*(\mathbf{P}^{dir})) \leq \lambda Z(\sigma^*(\mathbf{P}^{dir})) = \left(\frac{1}{2}\sqrt{\frac{\beta+4}{\beta}} - \frac{1}{2}\right) Z(\sigma^*(\mathbf{P}^{dir}))$. ∎

Proof of Lemma ??: We consider the case in which $\sum_{r=1}^{j} \tilde{a}_r \leq \sum_{r=1}^{k} \tilde{b}_r$. Suppose that J_k^b precedes J_j^a in the optimal schedule. Then moving J_k^b immediately after J_j^a will not increase the total flow time of the schedule. The proof of the case with $\sum_{r=1}^{j} \tilde{a}_r > \sum_{r=1}^{k} \tilde{b}_r$ follows a similar argument. ∎

Proof of Theorem ??: Similar argument as in the proof of Theorem **??**. ∎

Proof of Lemma ??: Consider an optimal schedule $\sigma^*(\mathbf{P}^{dir})$ of problem $\mathbf{P}^{dir}$. We construct a

feasible schedule $\sigma'(\mathbf{P}')$ for the auxiliary problem $\mathbf{P}'$ by taking the job processing sequence from $\sigma^*(\mathbf{P}^{dir})$ and assigning the jobs to batches the same way as $\sigma^*(\mathbf{P}^{dir})$. We now compare schedules $\sigma'(\mathbf{P}')$ and $\sigma^*(\mathbf{P}^{dir})$. Let B_k denote the kth batch of jobs departing from the plant. Let D'_k and D^*_k be the time of departure of batch B_k from the plant in schedules $\sigma'(\mathbf{P}')$ and $\sigma^*(\mathbf{P}^{dir})$, respectively. We have

$$D'_k = \sum_{J_j \in B_1 \cup \cdots \cup B_k} p_j = \frac{1}{2} \sum_{J_j \in B_1 \cup \cdots \cup B_k} (a_j + b_j) \leq \max \left\{ \sum_{J_j \in B_1 \cup \cdots \cup B_k} a_j, \sum_{J_j \in B_1 \cup \cdots \cup B_k} b_j \right\} = D^*_k.$$

The travel cost and travel time in schedule $\sigma'(\mathbf{P}')$ are the same as those in schedule $\sigma^*(\mathbf{P}^{dir})$. Therefore, $Z(\sigma'(\mathbf{P}')) \leq Z(\sigma^*(\mathbf{P}^{dir}))$. This implies that $Z(\sigma^*(\mathbf{P}')) \leq Z(\sigma^*(\mathbf{P}^{dir}))$. ∎

Proof of Theorem ??: We compare schedules σ^{H7} and $\sigma^*(\mathbf{P}')$. As mentioned earlier, σ^{H7} and $\sigma^*(\mathbf{P}')$ have the same job sequence, the same assignment of jobs to batches, and the same delivery cost. Let B_k denote the kth batch of jobs departing from the plant. Let D^{H7}_k and D_k be the time of departure of batch B_k from the plant in schedules σ^{H7} and $\sigma^*(\mathbf{P}')$, respectively. We have

$$D^{H7}_k = \max \left\{ \sum_{J_j \in B_1 \cup \cdots \cup B_k} a_j, \sum_{J_j \in B_1 \cup \cdots \cup B_k} b_j \right\} \leq \sum_{J_j \in B_1 \cup \cdots \cup B_k} (a_j + b_j) = 2 \sum_{J_j \in B_1 \cup \cdots \cup B_k} p_j = 2D_k.$$

Therefore, $Z(\sigma^{H7}) \leq 2Z(\sigma^*(\mathbf{P}'))$. By Lemma **??**, $Z(\sigma^{H7}) \leq 2Z(\sigma^*(\mathbf{P}^{dir}))$. This implies that $[Z(\sigma^{H7}) - Z(\sigma^*(\mathbf{P}^{dir}))]/Z(\sigma^*(\mathbf{P}^{dir})) \leq 1$. ∎

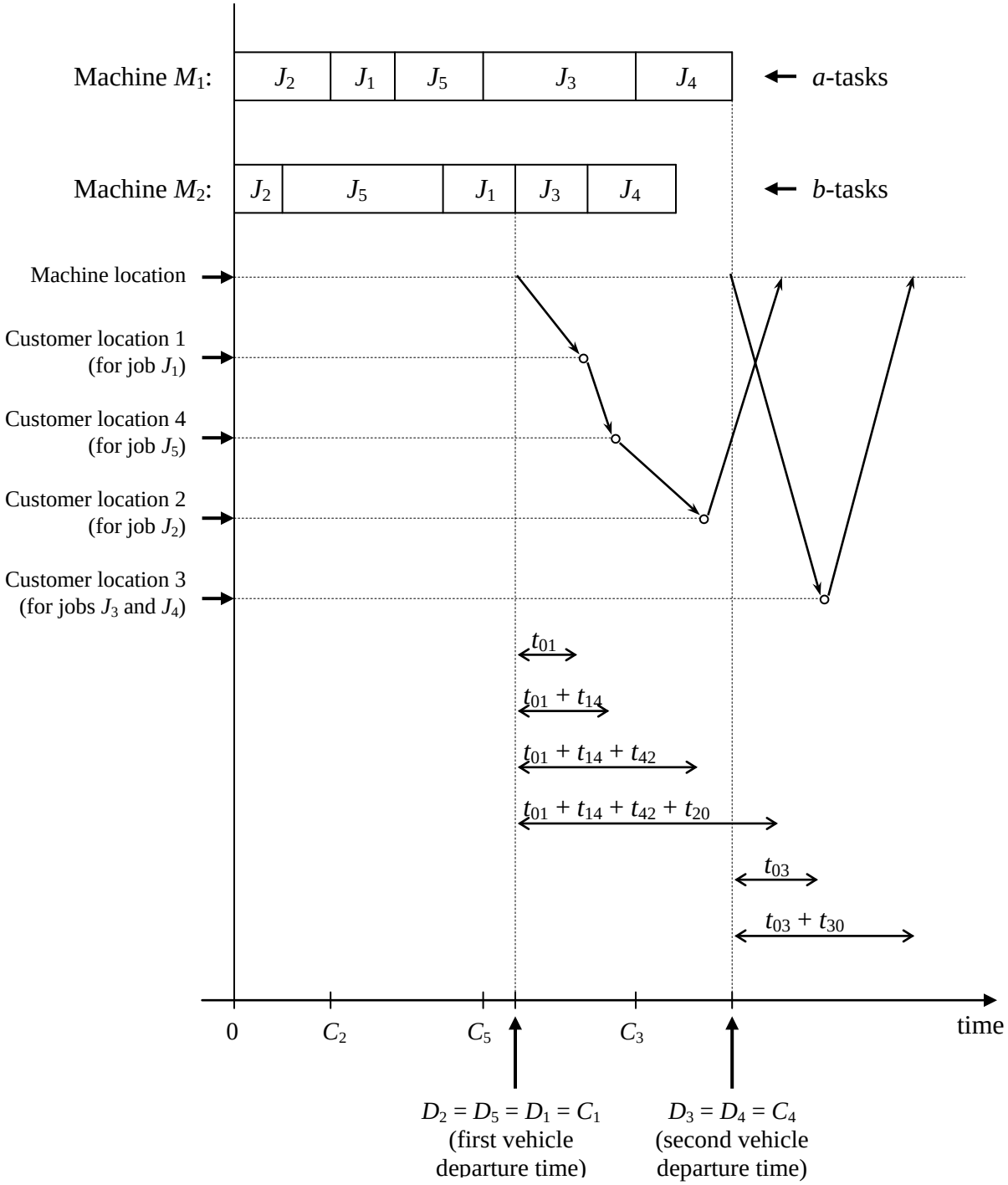

Figure 1. A numerical example

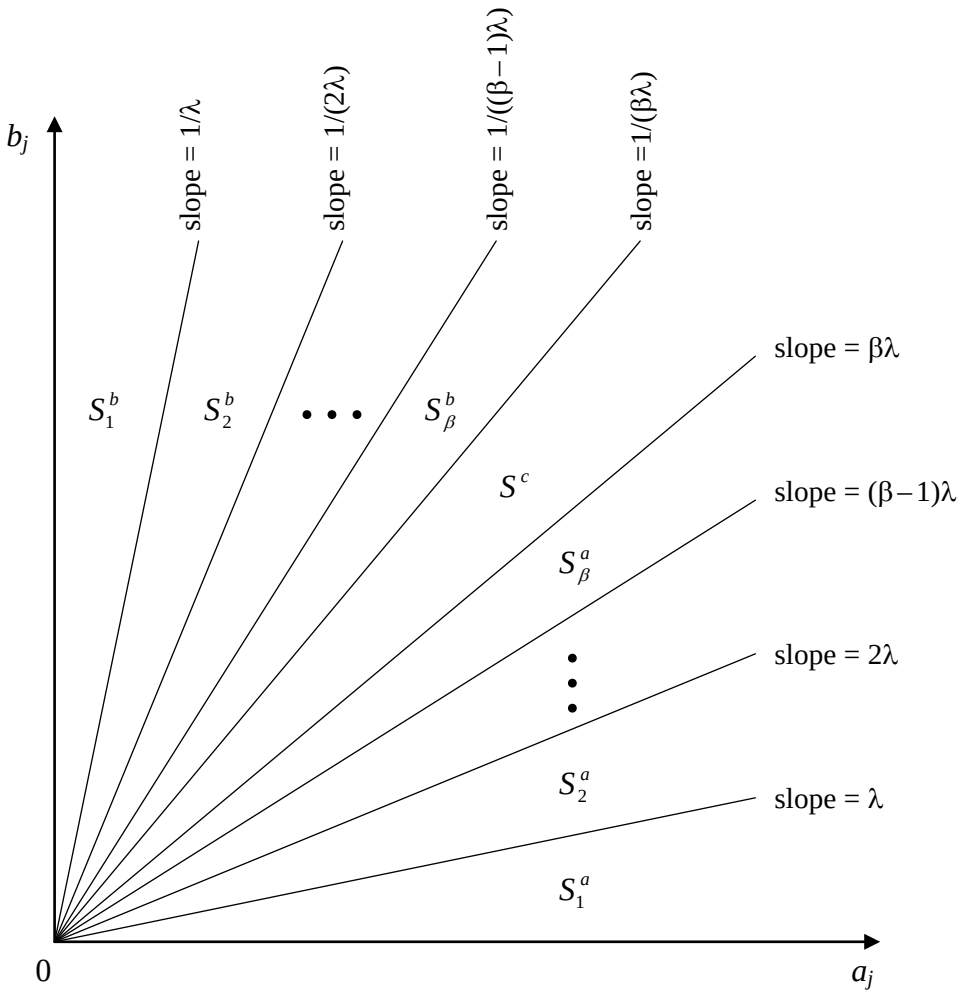

Figure 2. Partition of jobs in heuristic *H*1