Coordinated Scheduling of Customer Orders with Decentralized Machine Locations

Chung-Lun Li

Jinwen Ou

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

March 2006
Revised June 2006

Abstract

We consider a scheduling model with two machines at different locations. Each job is composed of two tasks where each task must be processed by a specific machine. The finished tasks are shipped to a distribution center in batches before they are bundled together and delivered to customers. The objective is to minimize the sum of the delivery cost and customers' waiting costs. This model attempts to coordinate the production and delivery schedules on the decentralized machines while taking into consideration the shipping cost as well as the waiting time of the customers. We develop polynomial-time heuristic algorithms for this problem and analyze their worst-case performance. Computational experiments are conducted to test the effectiveness of the heuristics and to evaluate the benefits obtained by coordinating the production and delivery of the two decentralized machines.

Key words: Scheduling; deliveries; coordination; heuristics; worst-case analysis

1 Introduction

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 production of personal computer systems, the computers and monitors are usually produced by different facilities at different locations. However, both the computers and monitors of the finished product must be bundled together before they can be delivered to customers. In order to obtain a systemwide optimal production and delivery plan, it is essential to consider the sequencing and scheduling of the tasks at each machine and the delivery arrangements of the finished tasks to their final destinations 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 consider a scheduling model which reflects the abovementioned production and delivery arrangements. In this scheduling model, each job is composed of two tasks where each task must be processed by a specific machine. The two machines are located at different locations, and different tasks of the same job can be processed by those machines simultaneously. The finished tasks are shipped to a distribution center (or consolidation center) before they are bundled together and delivered to customers. The objective is to minimize the sum of the delivery cost and customers' waiting costs. For simplicity, the delivery and waiting costs incurred after the finished jobs' arrival at the distribution center are not included in this model.

A number of researchers have considered parallel machine scheduling problems where each job order consists of products of different types and each machine is capable of producing only one specific product type. A job order is completed only after all of its tasks have finished their processing. This type of scheduling problems is usually referred to as "customer order scheduling" problems with dedicated machines. Various studies have been conducted on different variants of the problem (see Wagneur and Sriskandarajah 1993, Sung and Yoon 1998, Cai and Zhou 2004, Ahmadi *et al.* 2005, Leung *et al.* 2005a, 2005b, 2005c, 2006b, Li and Vairaktarakis 2006, and Yang 2005). However, in these "customer order scheduling" models, the machines are assumed to be located at the same location. More importantly, apart from Li and Vairaktarakis (2006), none of these works has taken the transportation of the finished tasks into account. Our model is an extension of the "customer order scheduling" framework, where machine locations, delivery batch capacities, delivery time, and delivery cost have been taken into consideration. Our work is more related to Li and Vairaktarakis (2006), since both papers consider job delivery decisions in a "customer order scheduling" setting. However, Li and Vairaktarakis have assumed that the two machines are located at the same location. They have considered the delivery of the completed orders to end customers, and have developed polynomial-time heuristics and approximation schemes for the case with only direct shipments as well as the general case with milk-run deliveries. On the other hand, we assume that the machines are located at different locations, and we consider the transportation of the finished tasks from the machines to a distribution center.

Another line of customer order scheduling research focuses on identical parallel machines (i.e., non-dedicated machines). In such models, the scheduler is allowed to assign jobs to any machine. A number of studies have examined the different variants of this problem such as the work of Blocher and Chhajed (1996), Leung *et al.* (2005d, 2006a), Yang (2003, 2005), and Yang and Posner (2005), among others. A few researchers have also developed customer order scheduling models with other machine structure. For example, Julien and Magazine (1990) have studied a customer order scheduling problem on a single machine, and Blocher *et al.* (1998) have considered a model with a job-shop setting. Unlike our model, none of these works has paid attention to decentralized machines or job delivery.

Our model is a machine-scheduling model with delivery considerations. In fact, integrated production and distribution models have received increasing attention. Recently, Chen and Pundoor

(2006) have analyzed a scheduling problem with multiple non-dedicated machines where each machine is located at a different location and has different production costs. Finished tasks are shipped to a distribution center, and each delivery shipment has a capacity limit. The decision is to assign jobs to machines, to determine the processing sequences, and to obtain a delivery schedule for the finished jobs. Thus, the setting of Chen and Pundoor's model is similar to ours. However, our model has taken jobs with multiple tasks into consideration where each task of a job must be processed by a specific machine. For a recent survey on integrated production and distribution operations, see Chen (2004).

The rest of the paper is organized as follows. In the next section, our problem is defined mathematically and several important properties of the optimal solution are developed. These properties will enable us to limit our search space for the optimal solution. In Section 3, an efficient heuristic is developed for our problem and worst-case analysis is performed. In Section 4, several variants of our model are analyzed. These variants are important stepping stones to the later development of our analysis. In Section 5, a polynomial-time heuristic with a stronger worst-case performance is presented. Computational results are reported in Section 6, followed by some concluding remarks in Section 7.

2 The Model and Its Properties

Our model is mathematically defined as follows. There are two machines M_1, M_2 and a distribution center located at different locations (see Figure 1). There is 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 T_{1j} and T_{2j}. Task T_{1j} must be processed by M_1 and requires an uninterrupted processing time of $p_{1j} \geq 0$, while task T_{2j} must be processed by M_2 and requires an uninterrupted processing time of $p_{2j} \geq 0$. Let C_{ij} denote the completion time of processing of T_{ij} on machine M_i ($i = 1, 2; j = 1, 2, \ldots, n$). A batch of tasks $\{T_{ij_1}, T_{ij_2}, \ldots, T_{ij_h}\}$ can be transported from M_i to the distribution center at a fixed delivery cost of $\lambda_i \geq 0$ after the completion of these tasks, provided that $h \leq K$, where $K \geq 1$ is the capacity of the delivery batch (i.e., K is the maximum number of tasks that a delivery vehicle can carry). Note that a variable delivery cost of $\mu_{ij} \geq 0$ for each task T_{ij} can be added to the model without affecting the analysis since the total variable delivery cost $\sum_{i=1}^{2} \sum_{j=1}^{n} \mu_{ij}$ is a constant. For simplicity, we ignore the variable delivery costs. Let $\tau_i \geq 0$ denote the travel time from M_i to the distribution center, and let D_{ij} denote the arrival time of T_{ij} at the distribution center. Thus, for the delivery batch $\{T_{ij_1}, T_{ij_2}, \ldots, T_{ij_h}\}$,

$$D_{ij_1} = D_{ij_2} = \cdots = D_{ij_h} = \max\{C_{ij_1}, C_{ij_2}, \ldots, C_{ij_h}\} + \tau_i.$$

Denote $D_j = \max\{D_{1j}, D_{2j}\}$, which is the time when both tasks of job J_j have arrived at the distribution center (i.e., the time where J_j is ready for delivery to the final customer). The customer's waiting cost of job J_j is given as γD_j, where γ is the unit cost of waiting. The objective is to schedule the tasks on each machine and to determine the delivery batches so as to minimize the sum of the total delivery cost and total customer waiting cost, i.e., $\sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} D_j$, where N_i is the number of batches of jobs transported from M_i to the distribution center. We denote this problem as **P**.

For example, a feasible solution to a problem instance of **P** with $n = 4$, $K = 3$, $\lambda_1 = \lambda_2 = 5$, $\gamma = 1$, $\tau_1 = 8$, $\tau_2 = 6$, $(p_{11}, p_{21}) = (4, 2)$, $(p_{12}, p_{22}) = (2, 5)$, $(p_{13}, p_{23}) = (10, 12)$, and $(p_{14}, p_{24}) = (10, 2)$ is depicted in Figure 2. In this solution, the first, second, and third delivery batches of M_1 contain $\{T_{12}, T_{11}\}$, $\{T_{14}\}$, and $\{T_{13}\}$, respectively. The first and second delivery batches of M_2 contain $\{T_{22}, T_{21}, T_{24}\}$ and $\{T_{23}\}$, respectively. We have $D_1 = \max\{D_{11}, D_{21}\} = \max\{14, 15\} = 15$, $D_2 = \max\{D_{12}, D_{22}\} = \max\{14, 15\} = 15$, $D_3 = \max\{D_{13}, D_{23}\} = \max\{34, 27\} = 34$, and $D_4 = \max\{D_{14}, D_{24}\} = \max\{24, 15\} = 24$. The total customer waiting cost is $\gamma \sum_{j=1}^{4} D_j = 88$, and the total delivery cost is $\lambda_1 N_1 + \lambda_2 N_2 = (5)(3) + (5)(2) = 25$. Thus, the total cost of this feasible solution is 113.

When $\lambda_1 = \lambda_2 = \tau_1 = \tau_2 = 0$, it is optimal to deliver one finished task at a time, and problem **P** reduces to the simple two-dedicated-machine order scheduling problem with an objective of minimizing the sum of job completion times, which is known to be NP-hard in the strong sense (see Ahmadi *et al.* 2005 and Yang 2005). Thus, problem **P** is strongly NP-hard as well.

The following lemma provides some important properties of the optimal solution.

Lemma 1 *There exists an optimal solution to problem* **P** *in which:*

(i) There is no idle time on either machine.

(ii) A delivery batch leaves the machine location as soon as all of its tasks have completed processing.

(iii) If two tasks of the same machine are assigned to the same delivery batch, then all the tasks processed in between these two tasks are also assigned to the same batch.

Proof: If a machine has idle time, then we can eliminate the idle time by shifting the start time of tasks to the left without increasing the waiting time and delivery costs of the jobs. This implies the validity of property (i). Property (ii) can be proven by a similar argument. Property (iii) can be proven easily by a task interchange argument. ∎

In the following sections, we will only consider schedules that satisfy properties (i)–(iii) of this lemma.

3 A Simple Heuristic Algorithm

In this section, we present a simple heuristic for problem **P**. This heuristic is efficient, and the relative error of its solution is guaranteed to be no more than 100%.

We construct a modified problem $\mathbf{P}'$ which has the same definition as problem **P**, except that the objective is to minimize $\sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} D'_j$, where $D'_j = (D_{1j} + D_{2j})/2$. Note that problem $\mathbf{P}'$ is decomposed into two independent subproblems. Subproblem i ($i = 1, 2$) is a single-machine problem with task processing times $p_{i1}, p_{i2}, \ldots, p_{in}$, delivery time τ_i, delivery cost λ_i, unit waiting cost $\gamma/2$, and batch capacity K. It is easy to see that there exists an optimal solution to subproblem

i in which the tasks are processed in nondecreasing order of task processing time. Thus, we first arrange the jobs in nondecreasing order of task processing time and reindex the tasks such that $p_{i1} \le p_{i2} \le \cdots \le p_{in}$. Then, we determine the delivery batches by the following dynamic program:

(1) Define $f_i(j)$ as the minimum total cost of the partial schedule which consists of tasks $T_{i1}, T_{i2}, \ldots, T_{ij}$ $(j = 1, 2, \ldots, n)$.

(2) Recurrence relation:

$$f_i(j) = \min_{\substack{k=0,1,\ldots,j-1 \\ \text{s.t. } k \ge j-K}} \Big\{ f_i(k) + \frac{\gamma}{2} \cdot (j-k)(P_{ij} + \tau_i) + \lambda_i \Big\},$$

where $P_{ij} = \sum_{\ell=1}^{j} p_{i\ell}$.

(3) Boundary condition: $f_i(0) = 0$.

(4) Optimal solution value: $f_i(n)$.

In the above recurrence relation, the quantity $(\gamma/2)(j-k)(P_{ij} + \tau_i) + \lambda_i$ is the total waiting and delivery cost of the jobs in the last delivery batch of the partial schedule. This delivery batch contains $j - k$ tasks.

After solving these two subproblems, an optimal schedule for problem $\mathbf{P}'$ is obtained. We use this schedule as a heuristic solution to the original problem $\mathbf{P}$ and denote this heuristic as $H1$. In the above dynamic program, the values of P_{ij} $(i = 1, 2;\ j = 1, 2, \ldots, n)$ can be predetermined in $O(n)$ time. The number of possible states is $O(n)$, and each state requires a computational time of $O(n)$. Hence, the running time of this dynamic program is $O(n^2)$. Therefore, the computational complexity of heuristic $H1$ is $O(n^2)$.

Let $Z^{H1}(\mathbf{P})$ denote the total cost of the solution generated by heuristic $H1$. Let $Z^*(\mathbf{P})$ and $Z^*(\mathbf{P}')$ denote the total costs of the optimal solutions to $\mathbf{P}$ and $\mathbf{P}'$, respectively.

Theorem 1 $\big[Z^{H1}(\mathbf{P}) - Z^*(\mathbf{P})\big] / Z^*(\mathbf{P}) \le 1.$

Proof: Because $D'_j \leq D_j$ for $j = 1, 2, \ldots, n$, the optimal solution to $\mathbf{P}'$ must have a total cost no greater than that of $\mathbf{P}$. In other words,

$$Z^*(\mathbf{P}') \leq Z^*(\mathbf{P}). \tag{1}$$

Next, consider the solution generated by heuristic $H1$. Because $D_j \leq 2D'_j$ for $j = 1, 2, \ldots, n$, we have $\sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} D_j \leq 2\big[\sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} D'_j\big]$, which implies that $Z^{H1}(\mathbf{P}) \leq 2Z^*(\mathbf{P}')$. This, together with (1), implies that $Z^{H1}(\mathbf{P}) \leq 2Z^*(\mathbf{P})$. ∎

Theorem 1 states that the relative error of the heuristic solution is guaranteed to be no more than 100%. It remains an interesting open question of whether this error bound is tight, that is, whether there exists a constant $\alpha < 1$ such that $[Z^{H1}(\mathbf{P}) - Z^*(\mathbf{P})]/Z^*(\mathbf{P}) \leq \alpha$.

4 Variants of Problem P

We now analyze three variants of problem $\mathbf{P}$. The development of effective solution methods for these variants is an important stepping stone to our later development of an improved error bound for the general problem.

4.1 When the Number of Delivery Batches Is Limited

We first consider the problem in which N_1 and N_2 are given parameters. This corresponds to the situation where the number of delivery batches from each machine location is reserved by the company in advance. We denote this problem as $\mathbf{P}(N_1, N_2)$.

To solve problem $\mathbf{P}(N_1, N_2)$, we propose the following heuristic method: Similar to heuristic $H1$, we construct a modified problem $\mathbf{P}'(N_1, N_2)$ which has the same definition as problem $\mathbf{P}(N_1, N_2)$, except that the objective is to minimize $\sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} D'_j$. Problem $\mathbf{P}'(N_1, N_2)$ is decomposed into two independent subproblems. Subproblem i ($i = 1, 2$) is a single-machine problem with task processing times $p_{i1}, p_{i2}, \ldots, p_{in}$, delivery time τ_i, delivery cost λ_i, unit waiting cost $\gamma/2$, batch capacity K, and a given number of delivery batches N_i. We arrange the tasks in nondecreasing order of task processing time, reindex the tasks such that $p_{i1} \leq p_{i2} \leq \cdots \leq p_{in}$, determine the values of $P_{ij} = \sum_{\ell=1}^{j} p_{i\ell}$ for $\ell = 1, 2, \ldots, n$, and then determine the delivery batches by the following dynamic program:

(1) Define $f_i(j, N)$ as the minimum total cost of the partial schedule which consists of tasks $T_{i1}, T_{i2}, \ldots, T_{ij}$, given that there are N deliveries available ($j = 1, 2, \ldots, n$; $N = 1, 2, \ldots, N_i$).

(2) Recurrence relation:

$$f_i(j, N) = \min_{\substack{k=0,1,\ldots,j-1 \\ \text{s.t. } k \geq j-K}} \Big\{ f_i(k, N-1) + \frac{\gamma}{2} \cdot (j-k)(P_{ij} + \tau_i) + \lambda_i \Big\}.$$

(3) Boundary conditions: $f_i(0, 0) = 0$; $f_i(j, 0) = +\infty$ for $j \geq 1$; and $f_i(0, N) = +\infty$ for $N \geq 1$.

(4) Optimal solution value: $f_i(n, N_i)$.

After solving these two subproblems, an optimal schedule for problem $\mathbf{P}'(N_1, N_2)$ is obtained. We use this schedule as a heuristic solution to problem $\mathbf{P}(N_1, N_2)$ and denote this heuristic as $H2(N_1, N_2)$. The running time of $H2(N_1, N_2)$ is $O(n^3)$. Note that the above dynamic program can be used to determine the values of all $f_i(n, N_i)$ for $i = 1, 2$ and $N_1, N_2 = \lceil n/K \rceil, \lceil n/K \rceil + 1, \ldots, n$ in $O(n^3)$ time. Hence, the heuristic solutions to $\mathbf{P}(N_1, N_2)$ for all N_1 and N_2 values can be determined in $O(n^3)$ time.

Let $Z^{H2}(\mathbf{P}(N_1, N_2))$ denote the total cost of the solution generated by heuristic $H2(N_1, N_2)$. Let $\sigma^*(\mathbf{P}(N_1, N_2))$ denote the optimal solution to problem $\mathbf{P}(N_1, N_2)$ and $Z^*(\mathbf{P}(N_1, N_2))$ be its total cost. Using the same arguments as in the proof of Theorem 1, we have

$$Z^{H2}(\mathbf{P}(N_1, N_2)) \leq \sum_{i=1}^{2} \lambda_i N_i + 2\gamma \sum_{j=1}^{n} D_j^*, \tag{2}$$

where D_j^* is the value of D_j in $\sigma^*(\mathbf{P}(N_1, N_2))$. Thus, $Z^{H2}(\mathbf{P}(N_1, N_2)) \leq 2Z^*(\mathbf{P}(N_1, N_2))$. This implies the following result, which provides a performance guarantee on heuristic $H2(N_1, N_2)$.

Theorem 2 $\big[Z^{H2}(\mathbf{P}(N_1, N_2)) - Z^*(\mathbf{P}(N_1, N_2))\big] / Z^*(\mathbf{P}(N_1, N_2)) \leq 1.$

4.2 When the Delivery Batch has Unit Capacity

Next, we consider a special case of problem $\mathbf{P}$ in which the capacity of the delivery batch is equal to one (i.e., $K = 1$). We denote this special case as $\mathbf{P}_1$. In this special case, $N_1 = N_2 = n$ in any feasible solution. Hence, throughout the analysis of this special case, we only consider solutions in which a delivery always takes place at the completion of a task. Li and Vairaktarakis (2006) have developed a polynomial-time approximation scheme (PTAS) for the problem with identical machine locations but no delivery considerations. We now extend Li and Vairaktarakis' PTAS to solve $\mathbf{P}_1$.

Lemma 2 *There exists an optimal solution to problem* $\mathbf{P}_1$ *in which:*

(i) The task processing sequences on both machines are identical.

(ii) If "$p_{1j} < p_{1k}$ *and* $p_{2j} \leq p_{2k}$*" or "*$p_{1j} \leq p_{1k}$ *and* $p_{2j} < p_{2k}$*," then* J_j *precedes* J_k *in the processing sequence.*

Proof: To prove property (i), suppose that in an optimal solution, there exists $\ell \in \{1, 2, \ldots, n\}$ such that the ℓth position of M_1 and the ℓth position of M_2 are occupied by tasks of different jobs. Then let

$$r = \max\left\{\ell \;\middle|\; \begin{array}{l}\text{the } \ell\text{th position of } M_1 \text{ and the } \ell\text{th position} \\ \text{of } M_2 \text{ are occupied by tasks of different jobs}\end{array}\right\}.$$

Let T_{1j} be the task which occupies the rth position of M_1 and T_{2k} be the task which occupies the rth position of M_2 (see Figure 3). Note that $j \neq k$. If $D_{1j} \leq D_{2k}$, then we can rearrange the processing of the tasks on M_1 by moving T_{1k} immediately behind T_{1j}, and this will not increase the arrival time of any job at the distribution center. Similarly, if $D_{1j} > D_{2k}$, then rearranging the tasks on M_2 by moving T_{2j} immediately behind T_{2k} will not increase the total cost of the schedule. Thus, by repeatedly applying this rearrangement of jobs, we can obtain an alternative optimal schedule which satisfies property (i). Property (ii) can be proven by a straightforward job interchange argument. ∎

In the rest of this subsection, we will only consider schedules that satisfy properties (i) and (ii) of this lemma. Given a positive integer β, we define the following job subsets:

$$S'_r = \left\{ J_j \in \mathbf{J} \;\middle|\; \tfrac{r-1}{\beta}\cdot p_{1j} \le p_{2j} < \tfrac{r}{\beta}\cdot p_{1j} \right\} \qquad (r = 1, 2, \ldots, \beta);$$

$$S''_r = \left\{ J_j \in \mathbf{J} \;\middle|\; \tfrac{r-1}{\beta}\cdot p_{2j} \le p_{1j} < \tfrac{r}{\beta}\cdot p_{2j} \right\} \qquad (r = 1, 2, \ldots, \beta-1);$$

$$S''_\beta = \left\{ J_j \in \mathbf{J} \;\middle|\; \tfrac{\beta-1}{\beta}\cdot p_{2j} \le p_{1j} \le p_{2j} \right\}.$$

Clearly, $\{S'_1, S'_2, \ldots, S'_\beta, S''_1, S''_2, \ldots, S''_\beta\}$ is a partition of $\mathbf{J}$. Using this job partition, we construct a modified problem $\bar{\mathbf{P}}_1$ with the following task processing times:

$$(\bar{p}_{1j}, \bar{p}_{2j}) = \begin{cases} \left(p_{1j},\, \tfrac{r-1}{\beta}\cdot p_{1j}\right), & \text{if } J_j \in S'_r \qquad (r = 1, 2, \ldots, \beta); \\ \left(\tfrac{r-1}{\beta}\cdot p_{2j},\, p_{2j}\right), & \text{if } J_j \in S''_r \qquad (r = 1, 2, \ldots, \beta); \end{cases}$$

for $j = 1, 2, \ldots, n$. The idea of this construction is to modify some of the original task processing times so that we can make use of property (ii) of Lemma 2 to obtain an optimal schedule in polynomial time. The construction is made in such a way that the changes in flow time of the tasks are under control.

By property (ii) of Lemma 2, there exists an optimal solution to $\bar{\mathbf{P}}_1$ in which the jobs in S'_r are processed in nondecreasing order of p_{1j} and the jobs in S''_r are processed in nondecreasing order of p_{2j}, for $r = 1, 2, \ldots, \beta$. Let $J_{\pi_r(1)}, J_{\pi_r(2)}, \ldots, J_{\pi_r(n_r)}$ denote the jobs in set S'_r, sorted in nondecreasing order of p_{1j} $(r = 1, 2, \ldots, \beta)$, where $n_r = |S'_r|$. Let $J_{\pi_{\beta+r}(1)}, J_{\pi_{\beta+r}(2)}, \ldots, J_{\pi_{\beta+r}(n_{\beta+r})}$ denote the jobs in set S''_r, sorted in nondecreasing order of p_{2j} $(r = 1, 2, \ldots, \beta)$, where $n_{\beta+r} = |S''_r|$. Hence, an optimal solution to $\bar{\mathbf{P}}_1$ can be obtained by optimally merging these 2β job sequences. This can be achieved by the following dynamic program.

Denote

$$\mathbf{J}(x_1, x_2, \ldots, x_{2\beta}) = \bigcup_{r=1}^{2\beta} \left\{ J_{\pi_r(1)}, J_{\pi_r(2)}, \ldots, J_{\pi_r(x_r)} \right\}.$$

Define $f(x_1, x_2, \ldots, x_{2\beta})$ as the minimum total customer waiting cost of the partial schedule which consists of the jobs in $\mathbf{J}(x_1, x_2, \ldots, x_{2\beta})$, where $x_r = 0, 1, \ldots, n_r$ for $r = 1, 2, \ldots, 2\beta$. We have the following recurrence relation:

$$f(x_1, x_2, \ldots, x_{2\beta})$$
$$= \min_{\substack{\ell=1,2,\ldots,2\beta \\ \text{s.t. } x_\ell \neq 0}} \left\{ f(x_1, \ldots, x_{\ell-1}, x_\ell - 1, x_{\ell+1}, \ldots, x_{2\beta}) + \gamma \max \left\{ \sum_{r=1}^{2\beta} \sum_{j=1}^{x_r} \bar{p}_{1j} + \tau_1, \sum_{r=1}^{2\beta} \sum_{j=1}^{x_r} \bar{p}_{2j} + \tau_2 \right\} \right\}.$$

The boundary condition is $f(0, 0, \ldots, 0) = 0$, and the optimal solution value of problem $\bar{\mathbf{P}}_1$ is $f(n_1, n_2, \ldots, n_{2\beta}) + (\lambda_1 + \lambda_2)n$. Let $\sigma^*(\bar{\mathbf{P}}_1)$ denote the optimal schedule to problem $\bar{\mathbf{P}}_1$ obtained by this dynamic program. We take the job sequence of this schedule and use it as a heuristic solution to problem $\mathbf{P}_1$.

The values of $\sum_{r=1}^{2\beta} \sum_{j=1}^{x_r} \bar{p}_{1j}$ and $\sum_{r=1}^{2\beta} \sum_{j=1}^{x_r} \bar{p}_{2j}$ $(x_r = 0, 1, \ldots, n_r;\ r = 1, 2, \ldots, 2\beta)$ can be predetermined in $O(n^{2\beta})$ time. Thus, the above dynamic program solves the problem in $O(\beta n^{2\beta})$ time. If β is a constant, then the running time of this heuristic is $O(n^{2\beta})$. We denote this heuristic as $H3(\beta)$. Let $\sigma^{H3(\beta)}(\mathbf{P}_1)$ denote the schedule generated by $H3(\beta)$, and let $\Gamma^{H3(\beta)}(\mathbf{P}_1)$ denote the total customer waiting cost of this solution. Let $\Gamma^*(\bar{\mathbf{P}}_1)$ denote the total customer waiting cost of $\sigma^*(\bar{\mathbf{P}}_1)$, and $\Gamma^*(\mathbf{P}_1)$ denote the optimal total customer waiting cost of problem $\mathbf{P}_1$.

Lemma 3 $\left[\Gamma^{H3(\beta)}(\mathbf{P}_1) - \Gamma^*(\mathbf{P}_1)\right] / \Gamma^*(\mathbf{P}_1) \leq 1/\beta$.

Proof: Let $J_{\pi(j)}$ denote the jth job in schedule $\sigma^*(\bar{\mathbf{P}}_1)$ and Δ_j denote the difference in arrival time of $J_{\pi(j)}$ at the distribution center between schedules $\sigma^{H3(\beta)}(\mathbf{P}_1)$ and $\sigma^*(\bar{\mathbf{P}}_1)$. Let Δ'_j and Δ''_j denote the difference in completion time of processing of $T_{1,\pi(j)}$ and $T_{2,\pi(j)}$, respectively, between these two schedules. We have

$$\Delta'_j = \sum_{k=1}^{j} (p_{1,\pi(k)} - \bar{p}_{1,\pi(k)}) \leq \sum_{\substack{k=1,2,\ldots,j \text{ s.t.} \\ J_{\pi(k)} \in S''_1 \cup \cdots \cup S''_\beta}} \frac{1}{\beta} \cdot p_{2,\pi(k)} = \frac{1}{\beta} \sum_{\substack{k=1,2,\ldots,j \text{ s.t.} \\ J_{\pi(k)} \in S''_1 \cup \cdots \cup S''_\beta}} \bar{p}_{2,\pi(k)} \leq \frac{1}{\beta} \sum_{k=1}^{j} \bar{p}_{2,\pi(k)}$$

and

$$\Delta''_j = \sum_{k=1}^{j} (p_{2,\pi(k)} - \bar{p}_{2,\pi(k)}) \leq \sum_{\substack{k=1,2,\ldots,j \text{ s.t.} \\ J_{\pi(k)} \in S'_1 \cup \cdots \cup S'_\beta}} \frac{1}{\beta} \cdot p_{1,\pi(k)} = \frac{1}{\beta} \sum_{\substack{k=1,2,\ldots,j \text{ s.t.} \\ J_{\pi(k)} \in S'_1 \cup \cdots \cup S'_\beta}} \bar{p}_{1,\pi(k)} \leq \frac{1}{\beta} \sum_{k=1}^{j} \bar{p}_{1,\pi(k)}.$$

Thus, for $j = 1, 2, \ldots, n$,

$$\Delta_j \leq \max\{\Delta'_j, \Delta''_j\} \leq \frac{1}{\beta}\cdot\max\left\{\sum_{k=1}^{j}\bar{p}_{1,\pi(k)},\ \sum_{k=1}^{j}\bar{p}_{2,\pi(k)}\right\} \leq \frac{1}{\beta}\cdot\max\left\{\sum_{k=1}^{j}\bar{p}_{1,\pi(k)}+\tau_1,\ \sum_{k=1}^{j}\bar{p}_{2,\pi(k)}+\tau_2\right\}.$$

Hence,

$$\Gamma^{H3(\beta)}(\mathbf{P}_1) - \Gamma^*(\bar{\mathbf{P}}_1) = \gamma\sum_{j=1}^{n}\Delta_j \leq \frac{\gamma}{\beta}\sum_{j=1}^{n}\max\left\{\sum_{k=1}^{j}\bar{p}_{1,\pi(k)}+\tau_1,\ \sum_{k=1}^{j}\bar{p}_{2,\pi(k)}+\tau_2\right\} = \frac{1}{\beta}\cdot\Gamma^*(\bar{\mathbf{P}}_1).$$

Note that $\Gamma^*(\bar{\mathbf{P}}_1) \leq \Gamma^*(\mathbf{P}_1)$. Therefore, $\Gamma^{H3(\beta)}(\mathbf{P}_1) - \Gamma^*(\mathbf{P}_1) \leq (1/\beta)\cdot\Gamma^*(\mathbf{P}_1)$. ∎

Let $Z^{H3(\beta)}(\mathbf{P}_1)$ and $Z^*(\mathbf{P}_1)$ denote the total cost of schedules $\sigma^{H3(\beta)}(\mathbf{P}_1)$ and $\sigma^*(\mathbf{P}_1)$, respectively. Note that $Z^{H3(\beta)}(\mathbf{P}_1) = n(\lambda_1+\lambda_2)+\Gamma^{H3(\beta)}(\mathbf{P}_1)$ and $Z^*(\mathbf{P}_1) = n(\lambda_1+\lambda_2)+\Gamma^*(\mathbf{P}_1)$. Hence, Lemma 3 implies the following result.

Theorem 3 $\left[Z^{H3(\beta)}(\mathbf{P}_1) - Z^*(\mathbf{P}_1)\right]/Z^*(\mathbf{P}_1) \leq 1/\beta$.

Because the running time of $H3(\beta)$ is $O(n^{2\beta})$, Theorem 3 implies that $H3(\beta)$, $\beta = 1, 2, \ldots$, is a PTAS for problem $\mathbf{P}_1$.

4.3 When the Job Processing Sequence Is Predetermined

Next, we consider the case in which the task processing sequences on both machines are given and identical. In this case, our focus is on determining the delivery schedule of the finished tasks. We will present an efficient algorithm for obtaining the optimal schedule. For the convenience of presentation, we reindex the jobs in such a way that the job processing sequence is $J_1, J_2, \ldots, J_n$. Thus, the task processing sequence on M_i is $T_{i1}, T_{i2}, \ldots, T_{in}$ $(i = 1, 2)$, and the completion time of task T_{ij} is $P_{ij} = \sum_{\ell=1}^{j} p_{i\ell}$ $(i = 1, 2;\ j = 1, 2, \ldots, n)$.

Define $f(j; k_1, k_2)$ as the minimum total cost of the partial schedule which consists of jobs $J_1, J_2, \ldots, J_j$, given that tasks $T_{i,j+1}, T_{i,j+2}, \ldots, T_{ik_i}$ have been scheduled to depart from M_i in one batch at time P_{ik_i} $(i = 1, 2)$, where $k_1, k_2 = j+1, j+2, \ldots, n$ and $j = 0, 1, \ldots, n-1$. Note that in the definition of $f(j;k_1,k_2)$, tasks $T_{1,j+1}, T_{1,j+2}, \ldots, T_{1k_1}$ form a delivery batch. This batch has unused capacity if $k_1 < j+K$. In such a case, we may choose to include task T_{1j} in this batch without incurring an additional delivery cost. Similarly, tasks $T_{2,j+1}, T_{2,j+2}, \ldots, T_{2k_2}$ form a delivery batch. If this batch has unused capacity, we may choose to include task T_{2j} in this batch at no additional delivery cost. Hence, we have the following recurrence relation:

$$\begin{aligned} f(j;k_1,k_2) = \min\Big\{ & f(j-1;k_1,k_2) + \gamma \max\{P_{1k_1}+\tau_1, P_{2k_2}+\tau_2\}, \\ & f(j-1;j,k_2) + \gamma \max\{P_{1j}+\tau_1, P_{2k_2}+\tau_2\} + \lambda_1, \\ & f(j-1;k_1,j) + \gamma \max\{P_{1k_1}+\tau_1, P_{2j}+\tau_2\} + \lambda_2, \\ & f(j-1;j,j) + \gamma \max\{P_{1j}+\tau_1, P_{2j}+\tau_2\} + \lambda_1 + \lambda_2 \Big\} \end{aligned}$$

if $k_1 \le j+K$ and $k_2 \le j+K$. In the right hand side of this equation, there are four choices. The first choice is to let T_{1j} depart from M_1 (together with $T_{1,j+1}, T_{1,j+2}, \ldots, T_{1k_1}$) at time P_{1k_1} and let T_{2j} depart from M_2 (together with $T_{2,j+1}, T_{2,j+2}, \ldots, T_{2k_2}$) at time P_{2k_2}. This does not incur any additional delivery cost. The second choice differs from the first choice in that T_{1j} is assigned to a different delivery batch which departs M_1 at time P_{1j} (i.e., immediately after the processing of T_{1j}). If this choice is made, a delivery cost of λ_1 is incurred. The third choice differs from the first choice in that T_{2j} is assigned to a different delivery batch which departs M_2 at time P_{2j} (i.e., immediately after the processing of T_{2j}). The fourth choice is to assign both T_{1j} and T_{2j} to new delivery batches.

The boundary conditions are

$$f(j;k_1,k_2) = +\infty \text{ if } k_1 > j+K \text{ or } k_2 > j+K;$$

$$f(0;k_1,k_2) = 0 \text{ if } k_1 \le K \text{ and } k_2 \le K.$$

The optimal solution value is $f(n-1;n,n) + \gamma \max\{P_{1n}+\tau_1, P_{2n}+\tau_2\} + \lambda_1 + \lambda_2$, where $f(n-1;n,n)$ is the total cost of $J_1, J_2, \ldots, J_{n-1}$, while $\gamma \max\{P_{1n}+\tau_1, P_{2n}+\tau_2\}$ and $\lambda_1+\lambda_2$ are the customer waiting cost and delivery cost, respectively, of J_n. We denote this dynamic programming algorithm as $A1$. The running time of algorithm $A1$ is $O(n^3)$.

5 An Improved Heuristic for Problem P

We now present a more effective heuristic for the general problem $\mathbf{P}$. Denote $N_{\min} = \lceil n/K \rceil$, and let β be a given positive integer parameter. The idea is to try both heuristics $H2(N_1, N_2)$ and $H3(\beta)$ on the given problem instance and select the better of the two results. Because heuristic $H3(\beta)$ is designed for the case with $K = 1$, we expect that it is only effective when the value of K is small. Therefore, we apply algorithm $A1$ (see subsection 4.3) to improve the result generated by $H3(\beta)$.

Heuristic $H4(\beta)$:

Step 1: For $N_1, N_2 = N_{\min}, N_{\min}+1, \ldots, n$, apply heuristic $H2(N_1, N_2)$ to obtain a solution to problem $\mathbf{P}(N_1, N_2)$ and denote the solution as $\sigma^{H2}(\mathbf{P}(N_1, N_2))$.

Step 2: Apply heuristic $H3(\beta)$ to obtain a solution to problem $\mathbf{P}_1$, and denote the solution as $\sigma^{H3(\beta)}(\mathbf{P}_1)$.

Step 3: Take the job processing sequence of $\sigma^{H3(\beta)}(\mathbf{P}_1)$ and apply algorithm $A1$ to obtain an optimal delivery schedule. Denote this solution as σ^{A1}.

Step 4: Select the best one among $\{\sigma^{H2}(\mathbf{P}(N_1, N_2)) \mid N_1, N_2 = N_{\min}, N_{\min}+1, \ldots, n\} \cup \{\sigma^{A1}\}$ as the solution to problem $\mathbf{P}$.

As explained in subsection 4.1, Step 1 of heuristic $H4(\beta)$ takes $O(n^3)$ time. Step 2 takes $O(n^{2\beta})$ time if β is a constant, and Step 3 takes $O(n^3)$ time. Hence, the overall running time of this heuristic is $O(n^{2\beta})$ when $\beta \geq 2$. If $K = 1$, then by Theorem 3, the relative error of the solution generated by this heuristic is guaranteed to be no more than $\frac{1}{\beta} \times 100\%$. Let $Z^{H4(\beta)}(\mathbf{P})$ denote the total cost of the solution generated by $H4(\beta)$, and $Z^*(\mathbf{P})$ denote the total cost of the optimal solution. The following theorem provides a performance guarantee on this heuristic when $K \geq 2$.

Theorem 4 *If $K \geq 2$, then $\left[Z^{H4(\beta)}(\mathbf{P}) - Z^*(\mathbf{P})\right]/Z^*(\mathbf{P}) \leq (K-1)/(K-\frac{1}{\beta})$.*

Proof: Consider an optimal solution $\sigma^*(\mathbf{P})$ to problem $\mathbf{P}$. Let $\Lambda^* = \lambda_1 N_1^* + \lambda_2 N_2^*$ denote the total delivery cost of $\sigma^*(\mathbf{P})$, where N_1^* and N_2^* are the values of N_1 and N_2, respectively, in this optimal solution. Let $\Gamma^* = \gamma \sum_{j=1}^{n} D_j^*$ denote the total customer waiting cost of $\sigma^*(\mathbf{P})$, where D_j^* is the value of D_j in this optimal solution. Recall that $Z^{H2}(\mathbf{P}(N_1, N_2))$ is the total cost of solution $\sigma^{H2}(\mathbf{P}(N_1, N_2))$ and $Z^{H3(\beta)}(\mathbf{P}_1)$ is the total cost of solution $\sigma^{H3(\beta)}(\mathbf{P}_1)$. We divide the analysis into two cases.

Case 1: $(K-1)\Lambda^* \geq (1 - \frac{1}{\beta})\Gamma^*$. In this case,

$$\frac{\Lambda^* + \Gamma^*}{\Gamma^*} \geq \frac{K - (1/\beta)}{K-1}. \tag{3}$$

Because one of the candidate solutions obtained in Step 1 of $H4(\beta)$ is $\sigma^{H2}(\mathbf{P}(N_1^*, N_2^*))$, we have $Z^{H4(\beta)}(\mathbf{P}) \leq Z^{H2}(\mathbf{P}(N_1^*, N_2^*)) \leq \Lambda^* + 2\Gamma^*$, where the second inequality follows from (2). This implies that

$$\begin{aligned} \frac{Z^{H4(\beta)}(\mathbf{P}) - Z^*(\mathbf{P})}{Z^*(\mathbf{P})} &\leq \frac{\Gamma^*}{\Lambda^* + \Gamma^*} \\ &\leq \frac{K-1}{K-(1/\beta)} \qquad \text{(by (3))}. \end{aligned}$$

Case 2: $(K-1)\Lambda^* < (1 - \frac{1}{\beta})\Gamma^*$. In this case,

$$\frac{\Lambda^* + \Gamma^*}{\Lambda^*} > \frac{K - (1/\beta)}{1 - (1/\beta)}. \tag{4}$$

Note that $N_1^* \geq n/K$ and $N_2^* \geq n/K$, which implies that $(\lambda_1 + \lambda_2)n \leq K\Lambda^*$. Consider the solution obtained in Step 2 of $H4(\beta)$, we have

$$\begin{aligned} Z^{H4(\beta)}(\mathbf{P}) &\leq Z^{H3(\beta)}(\mathbf{P}_1) \\ &\leq (\lambda_1 + \lambda_2)n + \left(1 + \frac{1}{\beta}\right)\Gamma^*(\mathbf{P}_1) \qquad \text{(by Lemma 3)} \\ &\leq K\Lambda^* + \left(1 + \frac{1}{\beta}\right)\Gamma^*(\mathbf{P}_1) \\ &\leq K\Lambda^* + \left(1 + \frac{1}{\beta}\right)\Gamma^*. \end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{Z^{H4(\beta)}(\mathbf{P}) - Z^*(\mathbf{P})}{Z^*(\mathbf{P})} &\le \frac{(K-1)\Lambda^* + (1/\beta)\Gamma^*}{\Lambda^* + \Gamma^*} \\
&= \frac{1}{\beta} + \Big(K - 1 - \frac{1}{\beta}\Big) \cdot \frac{\Lambda^*}{\Lambda^* + \Gamma^*} \\
&< \frac{1}{\beta} + \frac{(K - 1 - \frac{1}{\beta})(1 - \frac{1}{\beta})}{K - \frac{1}{\beta}} \qquad \text{(by (4))} \\
&= \frac{K-1}{K-(1/\beta)}.
\end{aligned}$$

Combining Cases 1 and 2 yields the desired result. ∎

Theorems 3 and 4 imply that there exists a polynomial-time heuristic for problem $\mathbf{P}$ with a worst-case error bound arbitrarily close to $(K-1)/K$ for any fixed integer $K \ge 1$. This error bound is larger as K gets larger, and it approaches 1 as K approaches infinity. This implies that the performance of heuristic $H4(\beta)$ has a better guarantee when the batch capacity is small.

6 Computational Experiments

To test the performance of our heuristics, a set of computational experiments has been conducted. In these experiments, we use randomly generated problems and then compare their heuristic solution values with the lower bounds of the optimal solution values. We test heuristic $H1$, as well as heuristic $H4(\beta)$ with $\beta = 2$ and 3.

Let Σ denote the set of all feasible solutions of problem $\mathbf{P}$. Define

$$LB_1(\alpha) = \min_{\sigma \in \Sigma} \left\{ \sum_{i=1}^{2} \lambda_i N_i + \gamma \sum_{j=1}^{n} \big[\alpha D_{1j} + (1-\alpha) D_{2j}\big] \right\},$$

where $0 \le \alpha \le 1$. For any given value of α, the value of $LB_1(\alpha)$ can be obtained via a dynamic program similar to that presented in Section 3. Because $\alpha D_{1j} + (1-\alpha)D_{2j} \le \max\{D_{1j}, D_{2j}\}$, we have $LB_1(\alpha) \le Z^*(\mathbf{P})$ for all $\alpha \in [0,1]$. Thus, a lower bound on $Z^*(\mathbf{P})$ is given as

$$LB_1 = \max_{\alpha \in I} \big\{ LB_1(\alpha) \big\},$$

where I is any finite subset of $[0,1]$. In our computational experiments, we have selected $I = \{0.00, 0.01, 0.02, \ldots, 1.00\}$.

Note that the total delivery cost of a given problem is at least $(\lambda_1 + \lambda_2) \cdot \lceil n/K \rceil$ and the total waiting cost of a given problem is at least $\Gamma^*(\bar{\mathbf{P}}_1)$. Thus, another lower bound on $Z^*(\mathbf{P})$ is given as

$$LB_2 = (\lambda_1 + \lambda_2) \cdot \left\lceil \frac{n}{K} \right\rceil + \Gamma^*(\bar{\mathbf{P}}_1).$$

We now develop some alternative lower bounds as follows. We reindex the jobs such that $p_{11} \leq p_{12} \leq \cdots \leq p_{1n}$. Define

$$p'_{1j} = p_{1j}$$

and

$$p'_{2j} = \min\{p_{2j}, p_{2,j+1}, \ldots, p_{2n}\}.$$

Let $\tilde{\mathbf{P}}$ denote the problem after replacing all p_{ij} with p'_{ij}. Note that $p'_{21} \leq p'_{22} \leq \cdots \leq p'_{2n}$. Thus, there exists an optimal solution to $\tilde{\mathbf{P}}$ in which the processing sequence on machine M_i is $T_{i1}, T_{i2}, \ldots, T_{in}$ for $i = 1, 2$. Hence, problem $\tilde{\mathbf{P}}$ can be solved efficiently by using the method developed in subsection 4.3. Let LB_3 denote the optimal solution value of $\tilde{\mathbf{P}}$. Clearly, LB_3 is a lower bound on $Z^*(\mathbf{P})$.

Similarly, we can reindex the jobs such that $p_{21} \leq p_{22} \leq \cdots \leq p_{2n}$ and define

$$p''_{1j} = \min\{p_{1j}, p_{1,j+1}, \ldots, p_{1n}\}$$

and

$$p''_{2j} = p_{2j}.$$

Let LB_4 denote the optimal solution value of the problem after replacing all p_{ij} by p''_{ij}. Then LB_4 is also a lower bound on $Z^*(\mathbf{P})$. We let

$$LB = \max\{LB_1, LB_2, LB_3, LB_4\},$$

which is the lower bound that we use in our computational study.

To obtain a random problem instance, we generate the task processing times p_{1j} and p_{2j} ($j = 1, 2, \ldots, n$) that are independent and uniformly distributed in the interval $(0, 1]$. We generate the delivery times τ_1 and τ_2 that are independent and uniformly distributed in the interval $(0, \tau_{\max}]$, where $\tau_{\max}$ is a given parameter. We assume that the unit cost of waiting, γ, is equal to 1 (in practice, if γ is not equal to 1 then we may rescale the monetary unit so that $\gamma = 1$). We generate the delivery costs λ_1 and λ_2 that are independent and uniformly distributed in the interval $(0, \lambda_{\max}]$, where $\lambda_{\max}$ is a given parameter.

In the computational study, the following parameters are used: n is set to 10, 20, 40, and 80; K is set to 1, 2, 4, and 8; $\tau_{\max}$ is set to 1 and 4; and $\lambda_{\max}$ is set to 1, 2, 4, and 8. Hence, there are 128 combinations of values of n, K, $\tau_{\max}$, and $\lambda_{\max}$. For each of these combinations, we generate 10 random problem instances. For each instance, we compute the heuristic solution values and the value of LB.

Denote

$$e^{H1} = \frac{Z^{H1}(\mathbf{P}) - LB}{LB} \times 100\%$$

and

$$e^{H4(\beta)} = \frac{Z^{H4(\beta)}(\mathbf{P}) - LB}{LB} \times 100\% \quad (\beta = 2, 3).$$

For each combination of n, K, $\tau_{\max}$, and $\lambda_{\max}$, we calculate the average values of e^{H1}, $e^{H4(2)}$, and $e^{H4(3)}$ (denoted as $\bar{e}^{H1}$, $\bar{e}^{H4(2)}$, and $\bar{e}^{H4(3)}$, respectively) from the 10 test instances. The quantities $\bar{e}^{H1}$, $\bar{e}^{H4(2)}$, and $\bar{e}^{H4(3)}$ are used as estimates of the relative errors of heuristics $H1$, $H4(2)$, and $H4(3)$, respectively.

Tables 1–4 summarize the computational results. From these results, we observe that heuristics $H4(2)$ and $H4(3)$ outperform heuristic $H1$ substantially while in most cases the performance of $H4(3)$ is slightly better than that of $H4(2)$. The performance of heuristics $H4(2)$ and $H4(3)$ tends to drop as K increases. This is consistent with the worst-case analysis result presented in Theorem 4. We also observe that the values of $\bar{e}^{H1}$, $\bar{e}^{H4(2)}$, and $\bar{e}^{H4(3)}$ increase as n increases. However, as stated in Theorems 1 and 4, there exist upper limits on these relative errors. The performance of these heuristics is better when $\tau_{\max} = 4$ as compared to $\tau_{\max} = 1$. This is because both the heuristic and optimal schedules of a given problem instance will remain unchanged if τ_1 and τ_2 are increased by the same amount Δ. The only difference is that the total cost of both the heuristic and optimal solutions will increase by $\gamma n\Delta$. Hence, an increase in τ_1 and τ_2 simultaneously will result in a smaller relative error of the heuristic solution. Therefore, the relative errors of the heuristics tend to decrease as $\tau_{\max}$ increases. In our experiments, when $\lambda_{\max} = 8$, over 80% of the delivery batches in the heuristic solutions are full. On the other hand, when $\lambda_{\max} = 1$, except for the case where $K = 1$, most delivery batches in the heuristic solutions are not full. In most combinations of n, K, and $\tau_{\max}$, the values of $\bar{e}^{H1}$, $\bar{e}^{H4(2)}$, and $\bar{e}^{H4(3)}$ reach a maximum at $\lambda_{\max} = 1$ or $\lambda_{\max} = 2$. The average value of $\bar{e}^{H4(3)}$ among all 1280 test instances is 6.4%, indicating that the overall effectiveness of heuristic $H4(3)$ is quite high.

Another set of computational experiments are then conducted to test the benefits of coordinating the schedules of the two decentralized machines through the use of our model. To achieve that, we compare the solutions obtained by heuristic $H4(3)$ with the solutions obtained by scheduling the production and delivery of each machine independently. We use the above randomly generated problem instances. For each problem instance, we determine

$$r = \frac{Z^{ind}(\mathbf{P}) - Z^{H4(3)}(\mathbf{P})}{Z^{ind}(\mathbf{P})} \times 100\%,$$

where $Z^{ind}(\mathbf{P})$ is the total cost of the solution to problem $\mathbf{P}$ obtained by solving two independent single-machine production and delivery problems $\mathbf{P}_1^{ind}$ and $\mathbf{P}_2^{ind}$. The objective of problem $\mathbf{P}_1^{ind}$ is to minimize $\lambda_1 N_1 + \gamma \sum_{j=1}^{n} D_{1j}$, while the objective to problem $\mathbf{P}_2^{ind}$ is to minimize $\lambda_2 N_2 + \gamma \sum_{j=1}^{n} D_{2j}$. Problems $\mathbf{P}_1^{ind}$ and $\mathbf{P}_2^{ind}$ can be solved optimally using the dynamic program presented in Section 3, except that the unit waiting cost is now γ instead of $\gamma/2$. For each combination of values of n, K, $\tau_{\max}$, and $\lambda_{\max}$, we calculate the average values of r (denoted as $\bar{r}$) from the 10 test instances. The quantity $\bar{r}$ is the percentage reduction in total cost if the coordinated schedule is used compared to the use of an uncoordinated schedule.

Table 5 summarizes the computational results. From these results, we observe that the saving obtained from coordinating the machine schedules increases as n increases. This implies that as the problem size increases, there are more saving opportunities available through coordinating the operations of the two decentralized machines. We also observe that such saving tends to increase as K increases. When K is large, it provides more flexibility to better coordinate the two machine schedules, and therefore, the benefit of coordination is more significant. The percentage saving obtained from coordination is smaller when $\tau_{\max} = 4$ as compared to $\tau_{\max} = 1$. Again, this is because an increase in τ_1 and τ_2 simultaneously will lead to an increase in both $Z^{H4(3)}(\mathbf{P})$ and $Z^{ind}(\mathbf{P})$ by the same amount. This results in a drop in r. Therefore, the percentage savings obtained from coordination tend to decrease as $\tau_{\max}$ increases.

7 Conclusions

In this paper, we studied a machine-scheduling model with two machines processing tasks at different locations where the completed tasks are delivered to a distribution center in batches. The problem is NP-hard in the strong sense. We first developed a simple heuristic and showed that the relative error of the heuristic solution must not exceed 100%. We further developed a more sophisticated polynomial-time heuristic with a better worst-case error bound which depends on the capacity of the delivery batches. Our computational study not only shows that the improved heuristic is effective in practice but also shows that the coordination of the production and delivery schedules of the two decentralized machines can provide a substantial saving in delivery and customer waiting costs.

There are several possible extensions to this research. One extension is to generalize our model and analysis to include more than two decentralized machines, tasks that occupy different amount of space in a delivery batch, and jobs with different waiting cost per time unit. Another extension is to consider the integration of production schedules of decentralized machines, deliveries from the decentralized machines to the distribution center, and the deliveries from the distribution center to end customers.

Acknowledgment

This research was supported in part by the Research Grants Council of Hong Kong under Grant No. PolyU5222/06E.

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.

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,* **3,** 555–572.

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

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.

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,* **53,** 243–260.

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 G. Vairaktarakis (2006). Coordinating production and distribution of jobs with bundling operations. *IIE Transactions,* forthcoming.

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.

Table 1. Computational results for K = 1

			n = 10	n = 20	n = 40	n = 80
K = 1	τ_{max} = 1	λ_{max} = 1	$\bar{e}^{H1}$ = 19.6% $\bar{e}^{H4(2)}$ = 5.4% $\bar{e}^{H4(3)}$ = 4.6%	$\bar{e}^{H1}$ = 25.7% $\bar{e}^{H4(2)}$ = 7.1% $\bar{e}^{H4(3)}$ = 6.0%	$\bar{e}^{H1}$ = 33.6% $\bar{e}^{H4(2)}$ = 10.8% $\bar{e}^{H4(3)}$ = 9.5%	$\bar{e}^{H1}$ = 39.6% $\bar{e}^{H4(2)}$ = 12.2% $\bar{e}^{H4(3)}$ = 10.2%
		λ_{max} = 2	$\bar{e}^{H1}$ = 16.0% $\bar{e}^{H4(2)}$ = 4.6% $\bar{e}^{H4(3)}$ = 3.6%	$\bar{e}^{H1}$ = 19.1% $\bar{e}^{H4(2)}$ = 5.6% $\bar{e}^{H4(3)}$ = 4.5%	$\bar{e}^{H1}$ = 29.9% $\bar{e}^{H4(2)}$ = 8.8% $\bar{e}^{H4(3)}$ = 7.7%	$\bar{e}^{H1}$ = 34.5% $\bar{e}^{H4(2)}$ = 10.0% $\bar{e}^{H4(3)}$ = 8.9%
		λ_{max} = 4	$\bar{e}^{H1}$ = 13.0% $\bar{e}^{H4(2)}$ = 3.6% $\bar{e}^{H4(3)}$ = 3.3%	$\bar{e}^{H1}$ = 17.2% $\bar{e}^{H4(2)}$ = 5.5% $\bar{e}^{H4(3)}$ = 4.9%	$\bar{e}^{H1}$ = 22.8% $\bar{e}^{H4(2)}$ = 5.9% $\bar{e}^{H4(3)}$ = 5.4%	$\bar{e}^{H1}$ = 33.0% $\bar{e}^{H4(2)}$ = 10.0% $\bar{e}^{H4(3)}$ = 8.7%
		λ_{max} = 8	$\bar{e}^{H1}$ = 6.0% $\bar{e}^{H4(2)}$ = 1.6% $\bar{e}^{H4(3)}$ = 1.2%	$\bar{e}^{H1}$ = 17.3% $\bar{e}^{H4(2)}$ = 4.5% $\bar{e}^{H4(3)}$ = 4.0%	$\bar{e}^{H1}$ = 20.7% $\bar{e}^{H4(2)}$ = 6.2% $\bar{e}^{H4(3)}$ = 5.1%	$\bar{e}^{H1}$ = 28.1% $\bar{e}^{H4(2)}$ = 7.4% $\bar{e}^{H4(3)}$ = 7.0%
	τ_{max} = 4	λ_{max} = 1	$\bar{e}^{H1}$ = 7.9% $\bar{e}^{H4(2)}$ = 2.5% $\bar{e}^{H4(3)}$ = 2.0%	$\bar{e}^{H1}$ = 14.3% $\bar{e}^{H4(2)}$ = 3.0% $\bar{e}^{H4(3)}$ = 2.8%	$\bar{e}^{H1}$ = 25.4% $\bar{e}^{H4(2)}$ = 7.0% $\bar{e}^{H4(3)}$ = 6.3%	$\bar{e}^{H1}$ = 32.3% $\bar{e}^{H4(2)}$ = 9.2% $\bar{e}^{H4(3)}$ = 8.3%
		λ_{max} = 2	$\bar{e}^{H1}$ = 6.4% $\bar{e}^{H4(2)}$ = 1.9% $\bar{e}^{H4(3)}$ = 1.3%	$\bar{e}^{H1}$ = 14.0% $\bar{e}^{H4(2)}$ = 3.6% $\bar{e}^{H4(3)}$ = 2.8%	$\bar{e}^{H1}$ = 21.8% $\bar{e}^{H4(2)}$ = 5.8% $\bar{e}^{H4(3)}$ = 4.9%	$\bar{e}^{H1}$ = 29.6% $\bar{e}^{H4(2)}$ = 8.1% $\bar{e}^{H4(3)}$ = 6.8%
		λ_{max} = 4	$\bar{e}^{H1}$ = 3.8% $\bar{e}^{H4(2)}$ = 1.3% $\bar{e}^{H4(3)}$ = 0.8%	$\bar{e}^{H1}$ = 14.7% $\bar{e}^{H4(2)}$ = 4.3% $\bar{e}^{H4(3)}$ = 3.4%	$\bar{e}^{H1}$ = 17.1% $\bar{e}^{H4(2)}$ = 4.0% $\bar{e}^{H4(3)}$ = 3.4%	$\bar{e}^{H1}$ = 30.1% $\bar{e}^{H4(2)}$ = 9.5% $\bar{e}^{H4(3)}$ = 8.4%
		λ_{max} = 8	$\bar{e}^{H1}$ = 3.1% $\bar{e}^{H4(2)}$ = 0.7% $\bar{e}^{H4(3)}$ = 0.6%	$\bar{e}^{H1}$ = 8.8% $\bar{e}^{H4(2)}$ = 2.4% $\bar{e}^{H4(3)}$ = 1.9%	$\bar{e}^{H1}$ = 15.1% $\bar{e}^{H4(2)}$ = 3.9% $\bar{e}^{H4(3)}$ = 3.3%	$\bar{e}^{H1}$ = 24.8% $\bar{e}^{H4(2)}$ = 6.8% $\bar{e}^{H4(3)}$ = 6.2%

Table 2. Computational results for K = 2

			n = 10	n = 20	n = 40	n = 80
K = 2	τ_{max} = 1	λ_{max} = 1	$\bar{e}^{H1}$ = 20.7% $\bar{e}^{H4(2)}$ = 6.7% $\bar{e}^{H4(3)}$ = 6.2%	$\bar{e}^{H1}$ = 27.8% $\bar{e}^{H4(2)}$ = 7.8% $\bar{e}^{H4(3)}$ = 7.1%	$\bar{e}^{H1}$ = 37.0% $\bar{e}^{H4(2)}$ = 11.7% $\bar{e}^{H4(3)}$ = 11.0%	$\bar{e}^{H1}$ = 42.4% $\bar{e}^{H4(2)}$ = 14.5% $\bar{e}^{H4(3)}$ = 12.5%
		λ_{max} = 2	$\bar{e}^{H1}$ = 14.3% $\bar{e}^{H4(2)}$ = 4.2% $\bar{e}^{H4(3)}$ = 3.6%	$\bar{e}^{H1}$ = 22.8% $\bar{e}^{H4(2)}$ = 6.6% $\bar{e}^{H4(3)}$ = 6.3%	$\bar{e}^{H1}$ = 33.2% $\bar{e}^{H4(2)}$ = 9.6% $\bar{e}^{H4(3)}$ = 8.8%	$\bar{e}^{H1}$ = 38.7% $\bar{e}^{H4(2)}$ = 11.9% $\bar{e}^{H4(3)}$ = 10.7%
		λ_{max} = 4	$\bar{e}^{H1}$ = 12.1% $\bar{e}^{H4(2)}$ = 2.6% $\bar{e}^{H4(3)}$ = 2.3%	$\bar{e}^{H1}$ = 20.7% $\bar{e}^{H4(2)}$ = 5.0% $\bar{e}^{H4(3)}$ = 5.1%	$\bar{e}^{H1}$ = 30.4% $\bar{e}^{H4(2)}$ = 9.0% $\bar{e}^{H4(3)}$ = 8.1%	$\bar{e}^{H1}$ = 37.0% $\bar{e}^{H4(2)}$ = 11.5% $\bar{e}^{H4(3)}$ = 10.0%
		λ_{max} = 8	$\bar{e}^{H1}$ = 7.4% $\bar{e}^{H4(2)}$ = 1.4% $\bar{e}^{H4(3)}$ = 1.3%	$\bar{e}^{H1}$ = 17.7% $\bar{e}^{H4(2)}$ = 5.8% $\bar{e}^{H4(3)}$ = 5.3%	$\bar{e}^{H1}$ = 23.0% $\bar{e}^{H4(2)}$ = 6.2% $\bar{e}^{H4(3)}$ = 5.7%	$\bar{e}^{H1}$ = 31.7% $\bar{e}^{H4(2)}$ = 9.2% $\bar{e}^{H4(3)}$ = 8.1%
	τ_{max} = 4	λ_{max} = 1	$\bar{e}^{H1}$ = 7.8% $\bar{e}^{H4(2)}$ = 2.6% $\bar{e}^{H4(3)}$ = 2.2%	$\bar{e}^{H1}$ = 19.8% $\bar{e}^{H4(2)}$ = 5.3% $\bar{e}^{H4(3)}$ = 4.6%	$\bar{e}^{H1}$ = 26.1% $\bar{e}^{H4(2)}$ = 7.1% $\bar{e}^{H4(3)}$ = 5.7%	$\bar{e}^{H1}$ = 34.7% $\bar{e}^{H4(2)}$ = 10.8% $\bar{e}^{H4(3)}$ = 9.3%
		λ_{max} = 2	$\bar{e}^{H1}$ = 9.7% $\bar{e}^{H4(2)}$ = 2.1% $\bar{e}^{H4(3)}$ = 1.8%	$\bar{e}^{H1}$ = 18.3% $\bar{e}^{H4(2)}$ = 5.1% $\bar{e}^{H4(3)}$ = 4.5%	$\bar{e}^{H1}$ = 25.6% $\bar{e}^{H4(2)}$ = 7.4% $\bar{e}^{H4(3)}$ = 6.3%	$\bar{e}^{H1}$ = 32.9% $\bar{e}^{H4(2)}$ = 9.1% $\bar{e}^{H4(3)}$ = 8.0%
		λ_{max} = 4	$\bar{e}^{H1}$ = 11.8% $\bar{e}^{H4(2)}$ = 3.5% $\bar{e}^{H4(3)}$ = 3.3%	$\bar{e}^{H1}$ = 13.4% $\bar{e}^{H4(2)}$ = 3.4% $\bar{e}^{H4(3)}$ = 2.9%	$\bar{e}^{H1}$ = 20.9% $\bar{e}^{H4(2)}$ = 5.0% $\bar{e}^{H4(3)}$ = 4.3%	$\bar{e}^{H1}$ = 31.6% $\bar{e}^{H4(2)}$ = 9.0% $\bar{e}^{H4(3)}$ = 8.2%
		λ_{max} = 8	$\bar{e}^{H1}$ = 2.8% $\bar{e}^{H4(2)}$ = 0.5% $\bar{e}^{H4(3)}$ = 0.3%	$\bar{e}^{H1}$ = 10.8% $\bar{e}^{H4(2)}$ = 2.7% $\bar{e}^{H4(3)}$ = 2.3%	$\bar{e}^{H1}$ = 21.6% $\bar{e}^{H4(2)}$ = 6.6% $\bar{e}^{H4(3)}$ = 5.9%	$\bar{e}^{H1}$ = 30.3% $\bar{e}^{H4(2)}$ = 9.0% $\bar{e}^{H4(3)}$ = 8.3%

Table 3. Computational results for K = 4

			n = 10	n = 20	n = 40	n = 80
K = 4	τ_{max} = 1	λ_{max} = 1	$\bar{e}^{H1}$ = 24.3% $\bar{e}^{H4(2)}$ = 7.9% $\bar{e}^{H4(3)}$ = 7.6%	$\bar{e}^{H1}$ = 31.8% $\bar{e}^{H4(2)}$ = 9.9% $\bar{e}^{H4(3)}$ = 9.6%	$\bar{e}^{H1}$ = 35.2% $\bar{e}^{H4(2)}$ = 11.3% $\bar{e}^{H4(3)}$ = 9.8%	$\bar{e}^{H1}$ = 37.5% $\bar{e}^{H4(2)}$ = 11.3% $\bar{e}^{H4(3)}$ = 10.6%
		λ_{max} = 2	$\bar{e}^{H1}$ = 20.5% $\bar{e}^{H4(2)}$ = 7.2% $\bar{e}^{H4(3)}$ = 7.0%	$\bar{e}^{H1}$ = 24.2% $\bar{e}^{H4(2)}$ = 7.8% $\bar{e}^{H4(3)}$ = 7.4%	$\bar{e}^{H1}$ = 35.4% $\bar{e}^{H4(2)}$ = 11.9% $\bar{e}^{H4(3)}$ = 11.3%	$\bar{e}^{H1}$ = 38.5% $\bar{e}^{H4(2)}$ = 12.1% $\bar{e}^{H4(3)}$ = 11.3%
		λ_{max} = 4	$\bar{e}^{H1}$ = 12.7% $\bar{e}^{H4(2)}$ = 3.2% $\bar{e}^{H4(3)}$ = 2.9%	$\bar{e}^{H1}$ = 21.9% $\bar{e}^{H4(2)}$ = 7.4% $\bar{e}^{H4(3)}$ = 6.7%	$\bar{e}^{H1}$ = 32.8% $\bar{e}^{H4(2)}$ = 9.3% $\bar{e}^{H4(3)}$ = 8.7%	$\bar{e}^{H1}$ = 37.1% $\bar{e}^{H4(2)}$ = 10.9% $\bar{e}^{H4(3)}$ = 10.3%
		λ_{max} = 8	$\bar{e}^{H1}$ = 13.3% $\bar{e}^{H4(2)}$ = 4.3% $\bar{e}^{H4(3)}$ = 3.9%	$\bar{e}^{H1}$ = 21.5% $\bar{e}^{H4(2)}$ = 6.0% $\bar{e}^{H4(3)}$ = 5.2%	$\bar{e}^{H1}$ = 29.5% $\bar{e}^{H4(2)}$ = 9.0% $\bar{e}^{H4(3)}$ = 8.5%	$\bar{e}^{H1}$ = 32.8% $\bar{e}^{H4(2)}$ = 10.1% $\bar{e}^{H4(3)}$ = 9.3%
	τ_{max} = 4	λ_{max} = 1	$\bar{e}^{H1}$ = 13.7% $\bar{e}^{H4(2)}$ = 4.8% $\bar{e}^{H4(3)}$ = 4.2%	$\bar{e}^{H1}$ = 22.9% $\bar{e}^{H4(2)}$ = 7.5% $\bar{e}^{H4(3)}$ = 6.7%	$\bar{e}^{H1}$ = 28.6% $\bar{e}^{H4(2)}$ = 8.6% $\bar{e}^{H4(3)}$ = 7.8%	$\bar{e}^{H1}$ = 30.1% $\bar{e}^{H4(2)}$ = 8.4% $\bar{e}^{H4(3)}$ = 7.1%
		λ_{max} = 2	$\bar{e}^{H1}$ = 12.8% $\bar{e}^{H4(2)}$ = 3.9% $\bar{e}^{H4(3)}$ = 3.6%	$\bar{e}^{H1}$ = 19.3% $\bar{e}^{H4(2)}$ = 5.0% $\bar{e}^{H4(3)}$ = 4.4%	$\bar{e}^{H1}$ = 27.8% $\bar{e}^{H4(2)}$ = 7.9% $\bar{e}^{H4(3)}$ = 7.1%	$\bar{e}^{H1}$ = 32.8% $\bar{e}^{H4(2)}$ = 9.2% $\bar{e}^{H4(3)}$ = 8.5%
		λ_{max} = 4	$\bar{e}^{H1}$ = 8.9% $\bar{e}^{H4(2)}$ = 2.3% $\bar{e}^{H4(3)}$ = 2.0%	$\bar{e}^{H1}$ = 18.7% $\bar{e}^{H4(2)}$ = 5.5% $\bar{e}^{H4(3)}$ = 4.7%	$\bar{e}^{H1}$ = 25.2% $\bar{e}^{H4(2)}$ = 7.3% $\bar{e}^{H4(3)}$ = 7.0%	$\bar{e}^{H1}$ = 32.1% $\bar{e}^{H4(2)}$ = 8.7% $\bar{e}^{H4(3)}$ = 8.3%
		λ_{max} = 8	$\bar{e}^{H1}$ = 6.3% $\bar{e}^{H4(2)}$ = 1.5% $\bar{e}^{H4(3)}$ = 1.4%	$\bar{e}^{H1}$ = 12.7% $\bar{e}^{H4(2)}$ = 3.4% $\bar{e}^{H4(3)}$ = 3.1%	$\bar{e}^{H1}$ = 21.7% $\bar{e}^{H4(2)}$ = 5.5% $\bar{e}^{H4(3)}$ = 4.9%	$\bar{e}^{H1}$ = 28.9% $\bar{e}^{H4(2)}$ = 8.3% $\bar{e}^{H4(3)}$ = 7.3%

Table 4. Computational results for K = 8

			n = 10	n = 20	n = 40	n = 80
K = 8	τ_{max} = 1	λ_{max} = 1	$\bar{e}^{H1}$ = 21.5% $\bar{e}^{H4(2)}$ = 8.5% $\bar{e}^{H4(3)}$ = 8.3%	$\bar{e}^{H1}$ = 29.8% $\bar{e}^{H4(2)}$ = 10.7% $\bar{e}^{H4(3)}$ = 10.0%	$\bar{e}^{H1}$ = 32.6% $\bar{e}^{H4(2)}$ = 9.8% $\bar{e}^{H4(3)}$ = 9.3%	$\bar{e}^{H1}$ = 37.2% $\bar{e}^{H4(2)}$ = 10.4% $\bar{e}^{H4(3)}$ = 9.2%
		λ_{max} = 2	$\bar{e}^{H1}$ = 19.9% $\bar{e}^{H4(2)}$ = 7.6% $\bar{e}^{H4(3)}$ = 7.4%	$\bar{e}^{H1}$ = 27.8% $\bar{e}^{H4(2)}$ = 8.9% $\bar{e}^{H4(3)}$ = 8.7%	$\bar{e}^{H1}$ = 33.7% $\bar{e}^{H4(2)}$ = 10.9% $\bar{e}^{H4(3)}$ = 10.3%	$\bar{e}^{H1}$ = 40.7% $\bar{e}^{H4(2)}$ = 12.8% $\bar{e}^{H4(3)}$ = 12.0%
		λ_{max} = 4	$\bar{e}^{H1}$ = 18.7% $\bar{e}^{H4(2)}$ = 8.2% $\bar{e}^{H4(3)}$ = 7.8%	$\bar{e}^{H1}$ = 29.7% $\bar{e}^{H4(2)}$ = 10.7% $\bar{e}^{H4(3)}$ = 10.6%	$\bar{e}^{H1}$ = 33.4% $\bar{e}^{H4(2)}$ = 10.3% $\bar{e}^{H4(3)}$ = 9.9%	$\bar{e}^{H1}$ = 41.5% $\bar{e}^{H4(2)}$ = 14.1% $\bar{e}^{H4(3)}$ = 13.3%
		λ_{max} = 8	$\bar{e}^{H1}$ = 13.4% $\bar{e}^{H4(2)}$ = 4.6% $\bar{e}^{H4(3)}$ = 4.5%	$\bar{e}^{H1}$ = 23.6% $\bar{e}^{H4(2)}$ = 8.9% $\bar{e}^{H4(3)}$ = 8.4%	$\bar{e}^{H1}$ = 27.4% $\bar{e}^{H4(2)}$ = 8.2% $\bar{e}^{H4(3)}$ = 7.6%	$\bar{e}^{H1}$ = 37.6% $\bar{e}^{H4(2)}$ = 11.7% $\bar{e}^{H4(3)}$ = 11.1%
	τ_{max} = 4	λ_{max} = 1	$\bar{e}^{H1}$ = 11.5% $\bar{e}^{H4(2)}$ = 4.5% $\bar{e}^{H4(3)}$ = 4.3%	$\bar{e}^{H1}$ = 20.5% $\bar{e}^{H4(2)}$ = 6.9% $\bar{e}^{H4(3)}$ = 6.7%	$\bar{e}^{H1}$ = 29.4% $\bar{e}^{H4(2)}$ = 9.0% $\bar{e}^{H4(3)}$ = 8.2%	$\bar{e}^{H1}$ = 31.9% $\bar{e}^{H4(2)}$ = 8.9% $\bar{e}^{H4(3)}$ = 8.2%
		λ_{max} = 2	$\bar{e}^{H1}$ = 11.0% $\bar{e}^{H4(2)}$ = 3.5% $\bar{e}^{H4(3)}$ = 3.5%	$\bar{e}^{H1}$ = 17.3% $\bar{e}^{H4(2)}$ = 4.8% $\bar{e}^{H4(3)}$ = 4.5%	$\bar{e}^{H1}$ = 30.6% $\bar{e}^{H4(2)}$ = 10.0% $\bar{e}^{H4(3)}$ = 9.7%	$\bar{e}^{H1}$ = 31.4% $\bar{e}^{H4(2)}$ = 9.1% $\bar{e}^{H4(3)}$ = 8.5%
		λ_{max} = 4	$\bar{e}^{H1}$ = 8.8% $\bar{e}^{H4(2)}$ = 3.2% $\bar{e}^{H4(3)}$ = 3.0%	$\bar{e}^{H1}$ = 17.7% $\bar{e}^{H4(2)}$ = 5.4% $\bar{e}^{H4(3)}$ = 5.0%	$\bar{e}^{H1}$ = 28.2% $\bar{e}^{H4(2)}$ = 9.9% $\bar{e}^{H4(3)}$ = 9.3%	$\bar{e}^{H1}$ = 34.1% $\bar{e}^{H4(2)}$ = 10.2% $\bar{e}^{H4(3)}$ = 9.6%
		λ_{max} = 8	$\bar{e}^{H1}$ = 7.7% $\bar{e}^{H4(2)}$ = 1.5% $\bar{e}^{H4(3)}$ = 1.4%	$\bar{e}^{H1}$ = 16.9% $\bar{e}^{H4(2)}$ = 5.3% $\bar{e}^{H4(3)}$ = 4.9%	$\bar{e}^{H1}$ = 23.6% $\bar{e}^{H4(2)}$ = 7.1% $\bar{e}^{H4(3)}$ = 6.7%	$\bar{e}^{H1}$ = 34.2% $\bar{e}^{H4(2)}$ = 10.8% $\bar{e}^{H4(3)}$ = 10.2%

Table 5. Percentage savings ($\bar{r}$) obtained from coordinating the two decentralized machines

			n = 10	n = 20	n = 40	n = 80
K = 1	τ_{max} = 1	λ_{max} = 1	11.6%	14.7%	17.4%	20.3%
		λ_{max} = 2	9.3%	11.4%	16.6%	18.7%
		λ_{max} = 4	8.3%	9.8%	14.1%	17.8%
		λ_{max} = 8	4.1%	10.8%	12.2%	16.4%
	τ_{max} = 4	λ_{max} = 1	5.0%	9.9%	15.1%	17.7%
		λ_{max} = 2	4.5%	8.8%	13.3%	17.3%
		λ_{max} = 4	2.7%	9.2%	11.3%	16.8%
		λ_{max} = 8	2.3%	6.4%	10.0%	14.8%
K = 2	τ_{max} = 1	λ_{max} = 1	12.4%	16.0%	18.3%	20.2%
		λ_{max} = 2	9.1%	13.5%	18.1%	19.8%
		λ_{max} = 4	8.6%	13.0%	16.9%	19.0%
		λ_{max} = 8	5.6%	10.2%	14.0%	17.6%
	τ_{max} = 4	λ_{max} = 1	5.5%	12.8%	15.7%	18.3%
		λ_{max} = 2	6.9%	11.5%	15.1%	18.5%
		λ_{max} = 4	7.8%	9.1%	13.7%	17.5%
		λ_{max} = 8	2.4%	7.4%	12.8%	16.9%
K = 4	τ_{max} = 1	λ_{max} = 1	13.7%	17.1%	18.2%	19.6%
		λ_{max} = 2	11.6%	13.8%	18.0%	19.7%
		λ_{max} = 4	9.2%	12.7%	18.0%	19.6%
		λ_{max} = 8	8.6%	13.2%	16.2%	17.6%
	τ_{max} = 4	λ_{max} = 1	8.9%	13.0%	15.7%	17.2%
		λ_{max} = 2	8.0%	12.9%	16.4%	17.1%
		λ_{max} = 4	6.3%	11.9%	14.5%	17.9%
		λ_{max} = 8	4.8%	8.9%	13.8%	16.7%
K = 8	τ_{max} = 1	λ_{max} = 1	11.8%	15.2%	17.7%	20.1%
		λ_{max} = 2	9.8%	15.4%	17.6%	20.4%
		λ_{max} = 4	10.0%	15.2%	18.1%	20.1%
		λ_{max} = 8	10.3%	12.6%	15.9%	19.4%
	τ_{max} = 4	λ_{max} = 1	7.3%	11.7%	16.5%	18.0%
		λ_{max} = 2	6.9%	11.3%	16.2%	17.6%
		λ_{max} = 4	6.2%	10.8%	15.0%	18.4%
		λ_{max} = 8	7.1%	10.4%	14.1%	18.0%

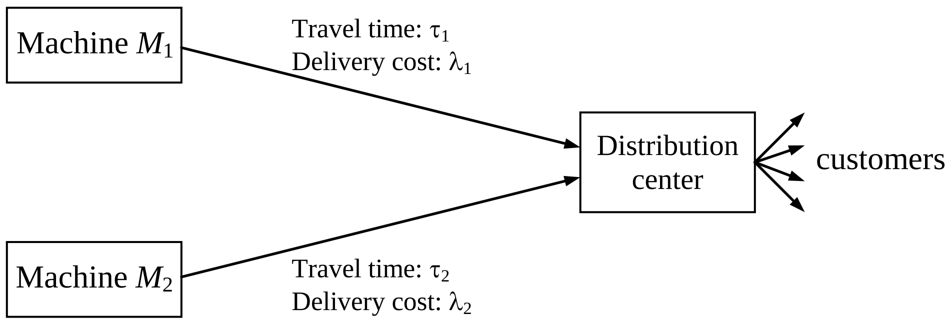

Figure 1. Machines and distribution center

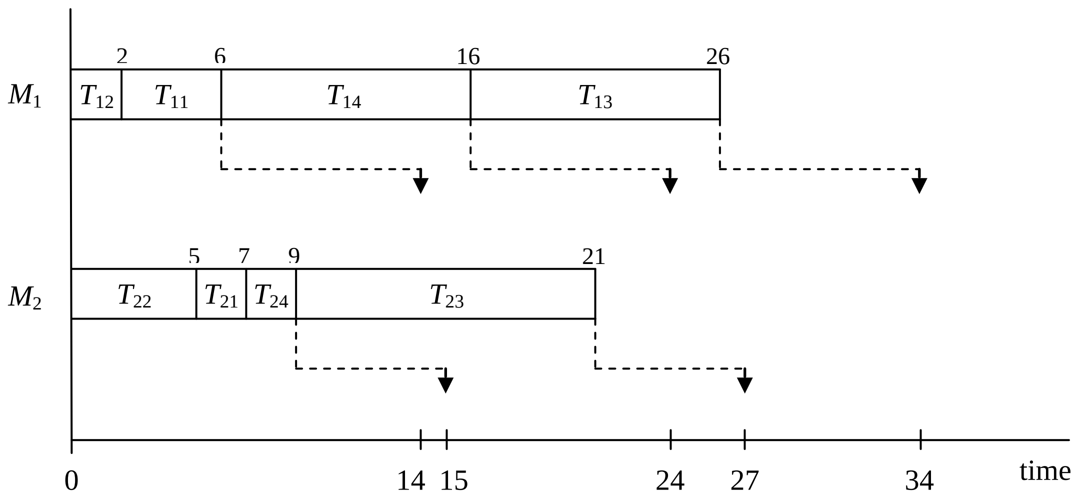

Figure 2. A numerical example

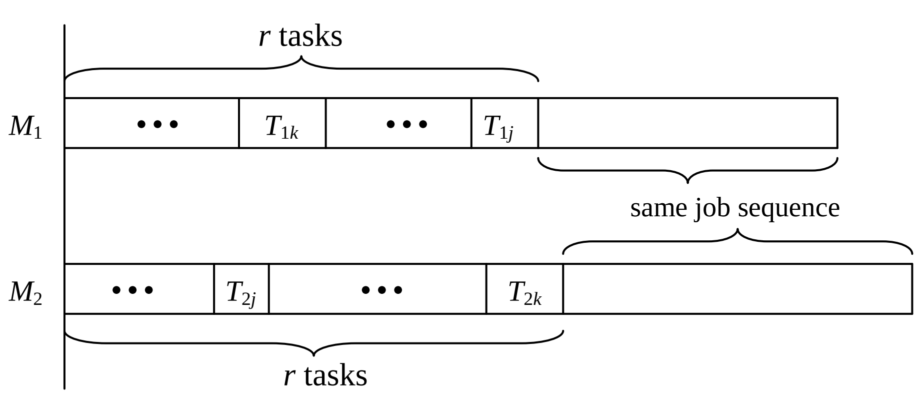

Figure 3. The schedule in the proof of property (i) of Lemma 2