Scheduling a Production–Distribution System To Optimize the Tradeoff between Delivery Tardiness and Distribution Cost

Guruprasad Pundoor, Zhi-Long Chen

Department of Decision and Information Technologies, Robert H. Smith School of Business, University of Maryland, College Park, Maryland 20742-1815

Received 17 June 2004; revised 26 January 2005; accepted 18 May 2005
DOI 10.1002/nav.20100
Published online 11 July 2005 in Wiley InterScience (www.interscience.wiley.com).

Abstract: We consider a make-to-order production–distribution system with one supplier and one or more customers. A set of orders with due dates needs to be processed by the supplier and delivered to the customers upon completion. The supplier can process one order at a time without preemption. Each customer is at a distinct location and only orders from the same customer can be batched together for delivery. Each delivery shipment has a capacity limit and incurs a distribution cost. The problem is to find a joint schedule of order processing at the supplier and order delivery from the supplier to the customers that optimizes an objective function involving the maximum delivery tardiness and the total distribution cost. We first study the solvability of various cases of the problem by either providing an efficient algorithm or proving the intractability of the problem. We then develop a fast heuristic for the general problem. We show that the heuristic is asymptotically optimal as the number of orders goes to infinity. We also evaluate the performance of the heuristic computationally by using lower bounds obtained by a column generation approach. Our results indicate that the heuristic is capable of generating near optimal solutions quickly. Finally, we study the value of production–distribution integration by comparing our integrated approach with two sequential approaches where scheduling decisions for order processing are made first, followed by order delivery decisions, with no or only partial integration of the two decisions. We show that in many cases, the integrated approach performs significantly better than the sequential approaches. © 2005 Wiley Periodicals, Inc. Naval Research Logistics 52: 571–589, 2005.

Keywords: production–distribution system; computational complexity; dynamic programming; heuristic; asymptotic analysis; value of production–distribution integration

1. INTRODUCTION

We consider a make-to-order production–distribution system consisting of one supplier and one or more customers. At the beginning of a planning horizon, each customer places a set of orders with the supplier. The supplier needs to process these orders and deliver the completed orders to the customers. Each order has a due date specified by the customer. Ideally, each customer wishes to receive her orders from the supplier by their respective due dates. However, since order deliveries incur distribution costs, the supplier wishes to consolidate the order delivery as much as possible to minimize the total distribution cost. Delivery consolidation implies that some completed orders may have to wait for other orders to be completed so that they can be delivered in the same shipment. Hence, some orders may be delivered to their customers after their due dates, resulting in a tradeoff between delivery timeliness and total distribution cost. The problem we consider in this paper is to find a joint schedule for order processing and delivery so that the tradeoff between the maximum delivery tardiness and total distribution cost is optimized.

This problem is often faced by manufacturers who make time-sensitive products such as fashion apparel and toys, which typically have large product types and sell only during specific seasons. Consider the production and distribution scheduling decisions such a manufacturer needs to make. The customers (e.g., distributors or retailers) often set due dates on the orders they place with the manufacturer and there is typically a penalty imposed on the manufacturer if the orders are not completed and delivered to the customers on time. Hence the manufacturer would like to meet the due dates as much as possible. Another factor the manufacturer has to consider is the total cost for order processing and delivery. Since the products are time-sensitive, orders are delivered shortly after their completion, and thus we

Correspondence to: Z.-L. Chen (zchen@rhsmith.umd.edu); G. Pundoor (pundoor@glue.umd.edu)

assume that little inventory cost is incurred. The manufacturer's total cost is mainly contributed by production and distribution operations. The total production cost for a fixed set of orders is normally fixed and independent of the production schedule used. Therefore, the manufacturer should focus on the distribution cost when considering the total cost. Since different orders may have different due dates, delivering more orders on time might require the manufacturer to make a larger number of shipments leading to higher total distribution cost. Therefore, the manufacturer has to find a production and distribution schedule that achieves some balance between delivery timeliness and total distribution cost. In practice, the *maximum tardiness* of orders and the *total tardiness* of orders are the two commonly used measurements for delivery timeliness. They represent the worst and average service level with respect to meeting order due dates, respectively. In this paper, we focus on the maximum tardiness as the measurement for delivery timeliness.

Scheduling problems with maximum tardiness related objectives have been studied extensively in the machine scheduling area (e.g., Pinedo [18]). However, most of the existing studies in this area consider only production operations. One of the earliest results is the EDD rule (Jackson [15]) that minimizes the maximum tardiness by scheduling the orders in the nondecreasing order of their due dates on a single machine. When we consider order delivery along with order processing, we may want to batch together a set of orders for shipping in order to reduce the total distribution cost. So batching becomes important. Even if an order in a batch is processed early, it has to wait till all the other orders in the batch are processed before getting delivered. Potts and Kovalyov [20] provide an extensive review of research in the area of scheduling with batching. However, many of the models described there differ from our model since batching in those models is done to take care of setup times between orders from different families instead of distribution. These problems deal only with the production part and do not consider production–delivery integration. For example, dynamic programming based and branch and bound based algorithms have been proposed to minimize the maximum tardiness for the single machine case with batch setup times (Ghosh and Gupta [8], Hariri and Potts [13]).

Some studies have tried to integrate production and distribution decisions (Cohen and Lee [5], Chandra and Fisher [1], Hahm and Yano [9], Fumero and Vercellis [6], Sarmiento and Nagi [21]). All of these studies address strategic or tactic levels of decisions, and do not study the model at the detailed scheduling level (Chen [3]). Some others that do study at the scheduling level differ from ours in the model structure and assumptions made. While there exist models that take into account both production and distribution performances into account, very few of them address the problem from a distribution cost and batching point of view. Even among such models, either the due dates are identical or it is assumed that order deliveries can be carried out instantaneously without any limit on the batch size. None of the models address the problem with general due dates, transportation times and costs, and delivery batch size limit simultaneously. Herrmann and Lee [14] consider the problem of minimizing the earliness–tardiness and batch delivery costs when the due dates are common. They assume that the cost per tardy delivery is fixed. Cheng, Gordon, and Kovalyov [4] consider a combined objective function for the single machine problem that takes into account the delivery costs and the sum of earliness of the orders. The earliness of an order is defined as the difference between the delivery time of the batch to which it belongs and its completion time on the machine. Chen [2] studies the problem of assigning a common due date to the orders to minimize a combined objective function that takes into account the earliness penalties, tardiness penalties, due date penalty, and delivery costs. Hall and Potts [10] consider various problems with an objective function combining a regular order delivery performance measure and delivery costs in the context of a two-stage supply chain. The last four papers also assume that delivery can be done instantaneously without any transportation time and that each delivery batch can carry any number of orders. Lee and Chen [16] study various problems of minimizing the maximum or total completion time of orders subject to the constraint that there are a limited number of transporters available for job delivery. Lee and Li [17] and Hall, Lesaoana, and Potts [12] study a similar model with the restriction that there are a fixed set of delivery dates at which the completed orders can be delivered. Studies such as Potts [19], Hall and Shmoys [11], and Woeginger [22, 23] take into account production and delivery times, but they do not consider the delivery costs. Hence each order can be shipped independently without increasing the cost function value.

In the following we describe the model to be studied in this paper. There are one supplier and m ($m \geq 1$) customers $M = \{1, \ldots, m\}$ located at different locations in a given production–distribution system. At the beginning of a planning horizon, the supplier receives n_i orders from each customer $i \in M$, requesting for processing. Let $n = n_1 + \cdots + n_m$ be the total number of orders. Let (i, j) denote the jth order from customer i, $N_i = \{(i, 1), \ldots, (i, n_i)\}$ be the set of the orders from customer i, and $N = N_1 \cup \cdots \cup N_m$ be the set of all the orders. All the orders are to be processed on a single production line at the supplier. Each order $(i, j) \in N$ has a processing time p_{ij} and a due date d_{ij}. Completed orders are delivered in batches to the customers. Due to the time sensitivity of the orders and the fact that each customer is located at a distinct location, direct shipping from the supplier to each customer is used. Therefore, only orders from the same customer can be batched together to form a delivery shipment and orders from different customers must be delivered separately. The delivery time and delivery cost from the supplier to customer $i \in M$ are t_i and f_i, respectively. The maximum allowed batch size (i.e., the maximum number of orders that can be shipped in a batch) is given by b. Let C_{ij} and D_{ij} denote the processing completion time and delivery time of order $(i, j) \in N$, respectively. We define $T_{ij} = \text{Max}\{0, D_{ij} - d_{ij}\}$ to be the tardiness of a particular order $(i, j) \in N$ and $T_{\max} = \max\{T_{ij}|(i, j) \in N\}$ to be the maximum tardiness of all orders. The total distribution cost for a given schedule is denoted as G, and $G = f_1x_1 + \cdots + f_mx_m$, where x_i is the number of shipments used to deliver the orders of customer $i \in M$.

The objective function should consider both $T_{\max}$ and G. In order to achieve this, we define a weighting factor α ($0 \le \alpha \le 1$), which is based on the manufacturer's relative preference on the two measurements $T_{\max}$ and G. The objective function is then defined as $\alpha T_{\max} + (1 - \alpha)G$. It can be seen that when α is close to 0, more emphasis is given to the total distribution cost. On the other hand, when α is close to 1, more emphasis is given to $T_{\max}$. In situations where the relative preference on the two measurements $T_{\max}$ and G is difficult to quantify, we can simply solve the problem multiple times with varying values of α and pick one of the resulting solutions with the right level of balance between the two measurements $T_{\max}$ and G.

The remainder of the paper is organized as follows. In Section 2, we analyze the computational complexity of the problem under various cases. We give efficient algorithms for the problem under several special cases and show that the problem under the general case is NP-hard and give efficient algorithms for other cases of the problem. In Section 3, we develop a quick heuristic for solving the general case of the problem. We show that the heuristic is asymptotically optimal with respect to the number of orders. To evaluate the performance of the heuristic, we develop a column generation based approach for generating lower bounds. Our computational experiment shows that the heuristic is capable of generating near optimal solutions. In Section 4, we compare the performance of the integrated production–distribution approach with two sequential approaches that treat order processing and order delivery independent of each other. Conclusions and scope for future work are given in Section 5.

2. ANALYSIS OF THE PROBLEM SOLVABILITY

In this section, we consider various cases of the problem. Since the problem with one customer, i.e., $m = 1$, has a different complexity from the problem with multiple customers, we consider these two cases separately. Another important case of the problem is when processing times and due dates of the orders are agreeable; i.e., if $p_{iu} \le p_{iv}$, then $d_{iu} \le d_{iv}$, for $1 \le u, v \le n_i$ and $i \in M$. This case arises in many practical environments where order due dates are set as a given multiple of the processing times. We define all the cases of the problem considered in this section as follows:

P1: The case where there is only one customer. In this case, for ease of presentation, we drop the customer index i from the problem parameters. Thus the n orders involved in the problem are $N = \{1, \ldots, n\}$, their processing times and due dates are $p_1, \ldots, p_n$, and $d_1, \ldots, d_n$ respectively, and the transportation time and transportation cost are t and f respectively.

P1A: The case P1 with agreeable processing times and due dates.

P2: The case where there are multiple customers, i.e., $m \ge 2$.

P2A: The case P2 with agreeable processing times and due dates.

Clearly, P1 is more general than P1A, and P2 is more general than the other three cases. We study the solvability of a problem by either providing an efficient algorithm for finding an optimal solution or proving that the problem is intractable. Problems P1A and P2A are studied in Sections 2.1. Problems P1 and P2 are studied in Sections 2.2 and 2.3 respectively.

Throughout the remainder of this paper, we say that a set of orders from the same customer are sequenced in *EDD order* (i.e., earliest due date first order) if they are sequenced in the nondecreasing order of their due dates. We require that, in case of a tie in due dates, the orders are sequenced in the nondecreasing order of their processing times and that if both the due dates and the processing times are the same amoung a set of orders, they are sequenced according to their indices. It can be seen that the above tie-breaking rule defines a unique EDD sequence for a given set of orders.

For all the problems, it is assumed without loss of generality that the orders belonging to each customer $i \in M$ have been indexed in the EDD order, i.e. $d_{i1} \le d_{i2} \le \cdots \le d_{in_i}$ for $i \in M$. Also, it is easily seen that there exists an optimal schedule where there is no idle time between the processing of orders at the supplier, and where the orders delivered in the same shipment are processed consecutively at the supplier.

2.1. P1A and P2A: The Problems with Agreeable Processing Times and Due Dates

We first present a property of problem P1A.

LEMMA 1: There exists an optimal solution for problem P1A where the orders are processed in EDD order.

PROOF: We prove this lemma by contradiction. Suppose that the lemma does not hold. Then there exist two orders u and v such that u is processed before v and $d_u > d_v$ (and hence $p_u > p_v$). Generate a new schedule by interchanging these two orders, keeping everything else the same as before. If these two orders belong to the same delivery batch in the earlier optimal solution, the value of $T_{\max}$ will remain unchanged, and the new schedule will be equivalent to the old one. Otherwise, the value of $T_{\max}$ will either decrease or remain unchanged because $d_u > d_v$ and $p_u > p_v$. The lemma follows immediately. □

By a similar argument as in the proof of Lemma 1, we can prove the following result. Hence it is stated without a proof.

LEMMA 2: There exists an optimal schedule for problem P2A where the orders from each customer are processed in their respective EDD order.

One of the problems considered by Hall and Potts [10] can be viewed as a special case of our problem P2 with $b = n$ (i.e., there is no batch size limit), and $t_i = 0$ for all $i \in M$ (i.e., there are no delivery times). They show that the result of Lemma 2 applies to their problem where processing times and due dates are not assumed to be agreeable. Based on this result, they propose an $O(n^{2m+1})$ dynamic programming algorithm for finding an optimal schedule for their problem. The idea of their algorithm is based on the following observation: If the number of delivery batches to each customer in the final schedule is given, then the total transportation cost is fixed. An optimal schedule can then be found by trying out all possible ways of splitting the orders of each customer (sequenced in their EDD order) into a desired number of batches. Their algorithm can be used to solve our problem P2A after it is modified by incorporating the batch size constraint and delivery times into their recursive relations. Since P1A can be viewed as a special case of P2A with $m = 1$, we can conclude that both problems P1A and P2A with a fixed number of customers m can be solved in polynomial time.

As we will see in the next section, when the processing times and due dates are not agreeable, the result of Lemma 1 does not hold for problem P1 and hence the result of Lemma 2 does not hold for problem P2 either. Consequently, the dynamic programming algorithm of Hall and Potts [10] does not work for our problems P1 or P2.

Next we consider the problem P2A with an arbitrary number of customers. The cases when $\alpha = 0$ or $\alpha = 1$ can be solved very easily. When $\alpha = 0$, the problem can be solved to optimality by minimizing the number of delivery batches for each customer. Any production schedule is optimal. To solve the problem when $\alpha = 1$, we define for each order $(i, j) \in N$, a *shipping due date* $d'_{ij} = d_{ij} - t_i$, which is the latest time the order (i, j) should leave the supplier in order to reach the customer without any tardiness. An optimal solution to this case is obtained by processing the orders in a nondecreasing sequence of the shipping due dates and delivering each order independently immediately after processing.

We show in the following that when $0 < \alpha < 1$, the problem P2A with an arbitrary m is NP-hard.

THEOREM 1: The problem P2A with $0 < \alpha < 1$ and an arbitrary number of customers is NP-hard.

PROOF: We prove this by reducing the Subset Sum problem, a known NP-hard problem, to P2A. The Subset Sum problem can be stated as follows (Garey and Johnson [7]): Given a set of $v + 1$ positive integers $a_1, \ldots, a_v$, and B, does there exist a subset $U \subseteq V = \{1, \ldots, v\}$ such that $\sum_{i \in U} a_i = B$?

Define $A = \sum_{i \in V} a_i$ and $H = \lceil 2v(1 - \alpha)/\alpha \rceil + v + B$. We construct a corresponding instance of problem P2A as follows:

Number of customers, $m = v$.
Number of orders from each customer, $n_i = 3$, for $i = 1, \ldots, m$.
Processing times: $p_{i1} = 1$, $p_{i2} = a_i$, $p_{i3} = Ha_i$, for $i = 1, \ldots, m$.
Due dates: $d_{i1} \equiv d_1 = m + B$, $d_{i2} \equiv d_2 = m + B + (H + 1)(A - B)$, and
$d_{i3} \equiv d_3 = m + (H + 1)A$, for $i = 1, \ldots, m$.
Transportation times and costs, $t_i = 0$, and $f_i = 1$, for $i = 1, \ldots, m$.
Maximum allowed batch size, $b = 3$.
Threshold cost, $F = 2m(1 - \alpha)$.

Clearly, the orders in this instance of P2A have agreeable processing times and due dates. For ease of presentation, we call orders $(i, 1)$, $(i, 2)$, and $(i, 3)$ from each customer $i \in M$ Type I, Type II, and Type III orders, respectively. We first prove the following properties: In a solution to this instance of P2A with the objective value not exceeding F, (a) Type I and Type III orders from each customer will be in different delivery batches, (b) there will be exactly $2m$ delivery batches, (c) there will be no tardy orders, and (d) orders of Type II will always be in a delivery batch of size 2. The proof for each is given next:

(a) If a Type I order is batched with a Type III order, the tardiness of the Type I order will be more than $H -$

d_1, and hence $\alpha T_{\max} > F$, which means that the objective value exceeds F.

(b) If there are more than $2m$ batches, the objective value will be greater than $F = (1 - \alpha)2m$. Since Type I and Type III orders cannot be put in the same batch, we need at least $2m$ total batches. Hence we have exactly $2m$ batches.

(c) Since $2m$ batches account for the entire threshold value, there can be no contribution from the tardiness part.

(d) This follows directly from parts (a) and (b).

Now we prove that there is a solution to the constructed instance of P2A with total cost not exceeding F if and only if there is a solution to the instance of the Subset Sum problem.

"→": Given a subset $U \subseteq V$ such that $\sum_{i \in U} a_i = B$, we construct a schedule for the instance of P2A as follows: First process all the Type I orders from customers $i \in V \backslash U$ and deliver each of them in a separate shipment. Next process the Type I and Type II orders from customers $i \in U$ and deliver them in batches of two orders. Next process the Type II and Type III orders from the customers $i \in V \backslash U$ and deliver them in batches of two orders. Finally process the orders of Type III from the customers $i \in U$ and deliver each of them separately. Let the cardinality of set U be k. The cardinality of $V \backslash U$ is $m - k$. The delivery time of the last batch with Type I and Type II orders is

$$D_1 = m + \sum_{i \in U} a_i = m + B = d_1.$$

Hence all the Type I orders are delivered before or on their due date. Similarly, the delivery time of the last batch with Type II and Type III orders is

$$D_2 = D_1 + (H + 1) \sum_{i \in V \backslash U} a_i = m + B + (H + 1)(A - B) = d_2.$$

Therefore, all the Type II orders are delivered before or on their due date. Similarly, the delivery time of the last Type III order in the schedule is

$$D_3 = D_2 + H \sum_{i \in U} a_i = m + B + (H + 1)(A - B) + HB = m + (H + 1)A = d_3.$$

We see that all the orders are delivered on time. The number of batches is given by: $(m - k) + k + (m - k) + k = 2m$. Hence the total cost is F.

"←": Let us assume that there are k batches that consist of orders of Type I and II. Let the indices of the corresponding Type II orders form set $U \subseteq V$. Since no tardy orders exist in the schedule [result (c) proved earlier], all these orders should be delivered no later than $d_1 = m + B$. Therefore, the maximum delivery time for orders of Type I is: $D_1 = m + \sum_{i \in U} a_i \leq m + B$, which means that

$$\sum_{i \in U} a_i \leq B. \tag{1}$$

Similarly, the delivery time D_2 of the last batch with Type II and Type III orders should not be greater than $d_2 = m + B + (H + 1)(A - B)$. Therefore,

$$D_2 = D_1 + (H + 1) \sum_{i \in V \backslash U} a_i = m + \sum_{i \in U} a_i + (H + 1) \sum_{i \in V \backslash U} a_i \leq m + B + (H + 1)(A - B).$$

This means that $m + A + H \sum_{i \in V \backslash U} a_i \leq m + HA - HB + A$, which further implies that $H \sum_{i \in V \backslash U} a_i \leq H(A - B)$. Therefore,

$$\sum_{i \in U} a_i \geq B. \tag{2}$$

From (1) and (2), we see that $\sum_{i \in U} a_i = B$ must hold. This means that set U is a solution to the instance of the Subset Sum problem.

Combining the "if" part and the "only if" part, we have proved the theorem. □

Theorem 1 implies that P2, the general case of the problem, is also NP-hard when $0 < \alpha < 1$ and the number of customers is arbitrary. When $\alpha = 0$ or $\alpha = 1$, P2 can be solved in polynomial time by adopting the same approach as the ones described earlier for P2A under those cases. The complexity of P2 with a fixed number of customers and $0 < \alpha < 1$ is an open problem.

2.2. P1: The Problem with One Customer and General Processing Times and Due Dates

We note that processing orders in their EDD order at the supplier is not necessarily optimal for this case of the problem. This is illustrated through the following example: Consider four orders with the following processing times and due dates: $p_1 = 1$, $p_2 = 5$, $p_3 = 1$, and $p_4 = 5$; $d_1 = 2$, $d_2 = 12$, $d_3 = 13$, and $d_4 = 14$. The transportation time $t = 0$ while the transportation cost $f = 10$. The maximum allowed batch size $b = 2$. Set $\alpha = 0.5$. Clearly, any solution to this problem will contain at least two delivery batches. Therefore, the objective value cannot be less than 10. We can obtain exactly 10 by putting the first and third orders in the first batch, and the rest in the second. Also, it can be seen that this is the only batch configuration that will give an objective value of 10. But this configuration violates the EDD rule. Hence we conclude that the EDD rule is not necessarily optimal for the general problem P1.

To solve P1, we first consider two related problems, called *auxiliary problem one* (AP1), and *auxiliary problem two* (AP2). We will solve our problem P1 by solving AP2 multiple times, where AP2 is solved by solving AP1 multiple times. Suppose that each order $j \in N$ has a deadline e_j. Problem AP1 is to schedule the production and distribution of the orders such that a minimum number of delivery shipments are used and all the orders are delivered to the customer before or at their deadlines. Problem AP2 is to schedule the production and distribution of the orders to minimize their maximum tardiness $T_{\max}$ subject to the constraint that the number of delivery shipments is no more than h for a given integer h. Later when we use AP1 to solve AP2, we will specify the deadline of each order e_j to be d_j plus some allowed tardiness. We focus on AP1 first. We propose the following algorithm to solve this problem. The algorithm schedules orders backwards and forms the delivery shipments from the last to the first.

Algorithm A1

Step 0: Let the set of unscheduled orders be $U = N$. Set the departure time of the current last shipment to be $Q = \sum_{j \in N} p_j$. Let $k = 1$.

Step 1: Find the subset of the orders that can be delivered in the kth last shipment without violating their deadlines, $S = \{j \in U | Q + t \le e_j\}$. If S is empty but U is not, then stop, and the problem is infeasible.

Step 2: If $|S| > b$, select the b orders with the largest processing times from S. Otherwise, select all the orders from S. Let X and P be the set of the selected orders and the total processing time of these orders, respectively. Process the selected orders consecutively without idle time in the time period $[Q - P, Q]$. Deliver them together in the kth last shipment with departure time Q. Update $Q = Q - P$, and $U = U \backslash X$. If U is empty, then stop, and we have a feasible schedule that uses exactly k batches. Otherwise, let $k = k + 1$, and go to Step 1.

LEMMA 3: Algorithm A1 finds an optimal solution to the problem AP1 in $O(n^2 \log n)$ time.

PROOF: We prove this by showing that any optimal solution π^* to the problem AP1 can be transformed to a feasible solution π generated by this algorithm without increasing the number of shipments. Suppose that for some integer $h \ge 0$, the kth last shipment in π^* is exactly the same as the kth last shipment in π, for $k = 1, \ldots, h$, but the $(h + 1)$st last shipment in π^* is different from the $(h + 1)$st last shipment in π. This implies that the set of the unscheduled orders U before the $(h + 1)$st last shipment is formed is the same in these two solutions. Also, the departure time Q of the $(h + 1)$st last shipment is the same in these two solutions. Let $S = \{j \in U | Q + t \le e_j\}$. There are two cases to consider as follows:

Case (i): $|S| > b$. In this case, the $(h + 1)$st shipment in π contains b orders. If there are less than b orders in the $(h + 1)$st shipment in π^*, we can move some orders in S that are scheduled in earlier shipments to the $(h + 1)$st last shipment so that this shipment contains b orders. If there are b orders in the $(h + 1)$st shipment in π^*, we can interchange some orders in this shipment with some other orders in S that are scheduled in earlier shipments but with larger processing times. It can be seen that in both cases the resulting new solution is still feasible and the number of shipments is not increased. Thus the $(h + 1)$st shipment in π^* can be transformed such that it becomes exactly the same as the $(h + 1)$st shipment in π.

Case (ii): $|S| \le b$. It can be similarly proved that in this case we can also transform the $(h + 1)$st shipment in π^* such that it becomes exactly the same as the $(h + 1)$st shipment in π without increasing the number of shipments.

This shows that the solution found by the algorithm is optimal. The algorithm carries out at most n iterations, each consisting of Steps 1 and 2. Since Step 1 takes at most $O(n)$ time and Step 2 takes at most $O(n \log n)$ time, the overall complexity of the algorithm is bounded by $O(n^2 \log n)$. □

Next we consider the second auxiliary problem AP2 which is to schedule the production and distribution of the orders to minimize their maximum tardiness $T_{\max}$ subject to the constraint that the number of delivery shipments is no more than h for a given integer h, where $\lceil n/b \rceil \le h \le n$. It can be seen that the value of $T_{\max}$ is nonincreasing with the value of h in the optimal solution of this problem. Based on this observation, we propose the following line search algorithm to find the optimal $T_{\max}$ given h.

Algorithm A2

Step 0: Let T^{LB} and T^{UB} denote a lower bound and an upper bound of the maximum delivery tardiness $T_{\max}$ of orders respectively. Initially, let T^{LB} be the maximum tardiness of orders if they are processed in the EDD order and each order is delivered separately, and let T^{UB} be the maximum tardiness of orders if they are processed in the EDD order and all are delivered in full shipments except possibly the last several orders which may be delivered in a partial shipment. Clearly, $T^{LB} \geq 0$ and $T^{UB} \leq t + P$, where $P = p_1 + \cdots + p_n$.

Step 1: Let $T^0 = (1/2)(T^{LB} + T^{UB})$. Define auxiliary problem one AP1 by imposing a deadline on each order $j \in N$, $e_j = d_j + T^0$. Solve this problem by Algorithm A1, and let the optimal number of shipments used be k.

Step 2: If $k > h$, let $T^{LB} = T^0$. Otherwise, let $T^{UB} = T^0$. If $T^{UB} - T^{LB} < 1$, stop. The only integer in the interval $[T^{LB}, T^{UB}]$ is adopted as the solution value of $T_{\max}$. Otherwise, go to Step 1.

LEMMA 4: Algorithm A2 finds an optimal solution to the problem AP2 in $O(n^2(\log n)(\log(P + t)))$ time, where $P = p_1 + \cdots + p_n$.

PROOF: As we observed earlier, the value of $T_{\max}$ is nonincreasing with the value of h in the optimal solution of this problem. Thus the solutions T^0 found in the line search involved in the algorithm are guaranteed to converge to the optimal solution if an infinitely many iterations are carried out. However, since the optimal value of $T_{\max}$ must be an integer, there is no need to carry out an infinite number of iterations. As soon as the gap between T^{LB} and T^{UB} becomes less than 1, there is at most one integer that can be contained in the interval $[T^{LB}, T^{UB}]$. Since this line search guarantees that the interval $[T^{LB}, T^{UB}]$ at each iteration contains the optimal solution, there must be an integer in this interval even when the width of this interval is less than 1. This shows that the algorithm does find the optimal solution.

The number of iterations in the line search is bounded by $O(\log(P + t))$. Since it takes $O(n^2 \log n)$ time to run Algorithm A1 in each iteration, the overall computational time is bounded by $O(n^2(\log n)(\log(P + t)))$. □

We propose the following algorithm based on A2 for solving our problem P1.

Algorithm A3

Step 1: For $h = \lceil n/b \rceil, \ldots, n$, do the following: Define an auxiliary problem AP2 with the number of delivery shipments no more than h. Solve the problem by applying Algorithm A2. Let π_h and $T_{\max}(\pi_h)$ be the optimal schedule and its maximum tardiness found by the algorithm.

Step 2: Find u such that $\alpha T_{\max}(\pi_u) + (1 - \alpha)uf = \min\{\alpha T_{\max}(\pi_h) + (1 - \alpha)hf | h = \lceil n/b \rceil, \ldots, n\}$. Then schedule π_u is optimal to problem P1 with the objective value $\alpha T_{\max}(\pi_u) + (1 - \alpha)uf$.

THEOREM 2: Algorithm A3 finds an optimal solution to the problem P1 in $O(n^3(\log n)(\log(P + t)))$ time, where $P = p_1 + \cdots + p_n$.

PROOF: By the definition of problem AP2, it can be seen that problem P1 is equivalent to finding the best h such that $\alpha T_{\max}(\pi_h) + (1 - \alpha)hf$ is minimum. This shows the optimality of the Algorithm A3. Since at most n auxiliary problems AP2 are solved in the algorithm, by Lemma 4, the overall complexity of A3 is bounded by $O(n^3(\log n)(\log(P + t)))$ time. □

Since the input size of our problem P1 is at least $\sum_{j \in N} (\lceil \log p_j \rceil + \lceil \log d_j \rceil) + \lceil \log t \rceil \geq n + \log(P + t)$, Algorithm A3 is polynomial.

3. A HEURISTIC FOR THE PROBLEM WITH MULTIPLE CUSTOMERS WHEN 0 < A < 1

In this section, we propose and evaluate a heuristic for the problem P2. Since P2 is a more general case of P2A, the heuristic is also applicable to P2A. We first prove an optimality property in Section 3.1. In Section 3.2, we develop the heuristic and prove it to be asymptotically optimal. In Section 3.3, we propose a linear programming based approach for obtaining tight lower bounds. We use column generation to solve the linear programming formulations to optimality. A set of computational experiments is carried out to evaluate the performance of the heuristic.

3.1. An Optimality Property

We define the *SEDD sequence* for a given set of orders as follows. Consider the shipping due dates of the orders as defined in Section 2.1. Arrange the orders in the nondecreasing order of their shipping due dates. In case of a tie, arrange the orders in the nondecreasing order of their processing times. If both the shipping due dates and the processing times are the same, arrange them by their customer index followed by their order index. It can be seen that the above tie-breaking rule defines a unique SEDD sequence for a given set of orders. Also, in the SEDD sequence, orders from the same customer are sequenced in their EDD order.

LEMMA 5: There exists an optimal schedule for P2, where:

(i) The orders that are delivered in the same batch are processed in their EDD sequence at the supplier.
(ii) The first orders of the batches form an SEDD sequence.
(iii) Let u denote the first order processed in a particular batch. All the orders that come before u in the SEDD sequence of all the orders of N are processed before this batch of orders.

PROOF: (i) Since all the orders in a batch are delivered at the same time, the tardiness of the batch is not influenced by the processing sequence of the orders in them. So we choose a sequence that is in the EDD order for the set of orders in the batch.

(ii) By (i), we can assume that the orders in each batch are processed in the EDD sequence. Let u be the first order in a batch. Suppose there exists an order v that is the first order in some earlier batch and has a shipping due date larger than that of u. Then we can move the batch containing order v to a position immediately after the batch containing order u without increasing the objective value. We can do this for every pair of batches that violates the lemma. In cases where two batches have their first orders with the same shipping due dates, the relative sequence of these two batches does not affect the tardiness value. Hence there exists an optimal solution where the first orders in the batches reflect the unique SEDD sequence.

(iii) This follows directly from (i) and (ii). □

3.2. The Heuristic

The heuristic first solves a single-customer *auxiliary problem* for each customer independently in such a way that the contribution due to the other customers is taken care of indirectly. Then it puts together the schedules for individual customers to obtain a combined schedule. We make use of Lemmas 2 and 5. That is, schedules are built such that the orders from each customer follow the EDD sequence and the set of first orders from every batch follows the SEDD sequence. Although Lemma 2 is not valid for the general problem P2, we will show that forcing the EDD sequence for each customer does not affect the results significantly when the number of orders is large.

Define C_{ij}^{SEDD} as the completion time of order $(i, j) \in N$ at the supplier when all of the orders from N are processed in the SEDD sequence. The single-customer auxiliary problem for customer $i \in M$, denoted as AUX_i, is defined as follows: Schedule the processing and delivery of the orders from N_i subject to the following two constraints: (i) the orders are processed in the EDD order at the supplier and (ii) the departure time of each delivery batch B containing (i, j) as the first order is required to be the sum of C_{ij}^{SEDD} and the total processing time of the remaining orders in the batch, i.e., $\sum_{(i,u)\in B} p_{iu} - p_{ij}$. The objective of problem AUX_i is to minimize the maximum delivery tardiness of the orders given that the orders are delivered in a given number of delivery batches. Due to constraint (ii), a feasible schedule to AUX_i may contain idle time between the processing of the last order in one batch and the first order in the next batch. We present the heuristic next.

Heuristic H1

Step 1: Create an auxiliary problem AUX_i, as described earlier, for each customer $i \in M$. Solve AUX_i, for $i \in M$, by the following dynamic programming algorithm, denoted as DP1, where the value function $F(j, k)$ is defined to be the minimum value of the maximum tardiness for the first j orders $\{(i, 1), (i, 2), \ldots, (i, j)\}$ when they are delivered in k batches.

DP1:

Initial values: $F(0, 0) = 0$.
Recursive relations: For $j = 1, \ldots, n_i$, and $k = \lceil j/b \rceil, \ldots, j$,

$$F(j, k) = \min_{1\le q\le \min\{b,j\}} \left\{ \max\left\{ F(j-q, k-1), \max\left\{ 0, C_{i,j-q+1}^{SEDD} + \sum_{u=j-q+2}^{j} p_{iu} + t_i - d_{i,j-q+1} \right\}\right\}\right\}. \quad (3)$$

Let $T_{\max}^i(k) = F(n_i, k)$ denote the maximum tardiness for customer i when the orders of customer i are delivered in k batches. Let $\Lambda_i(k)$ denote the corresponding batch configurations for customer i. Let $\Gamma_i = \{T_{\max}^i(k) | k = \lceil n_i/b \rceil, \ldots, n_i\}$, and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \cdots \cup \Gamma_m$. Clearly, $|\Gamma| \le \sum_{i=1}^{m} (n_i - \lceil n_i/b \rceil + 1)$.

Step 2: For each value $x \in \Gamma$, and each customer $i \in M$, define $k_i(x) = \min\{k \in \{\lceil n_i/b \rceil, \ldots, n_i\} | T_{\max}^i(k) \le x\}$ if there exists some $k \in \{\lceil n_i/b \rceil, \ldots, n_i\}$ with $T_{\max}^i(k) \le x$, and $k_i(x) = \infty$ otherwise. Find $x^* \in \Gamma$ such that

$$\alpha x^* + (1 - \alpha) \sum_{i \in M} f_i k_i(x^*) = \min_{x \in \Gamma} \left\{ \alpha x + (1 - \alpha) \sum_{i \in M} f_i k_i(x) \right\} \quad (4)$$

and the corresponding batch configurations $\Lambda_i(k_i(x^*))$ for each customer $i \in M$. Let π_i denote the schedule for customer i corresponding to the value function $F(n_i, k_i(x^*))$ [note that π_i is optimal to the problem AUX_i with $k_i(x^*)$ delivery batches].

Step 3: Sequence all the batches determined by the batch configurations $\{\Lambda_i(k_i(x^*)) | i \in M\}$ obtained in Step 2 such that the first orders of the batches form the SEDD sequence. This gives a feasible schedule π for the original problem. Calculate the objective value of π.

In the algorithm DP1, the recursive relation (3) enumerates all possible sizes q of the current last delivery batch. Hence DP1 solves AUX_i optimally for all possible number of delivery batches k. In Step 2, the selected value of maximum tardiness x^* optimizes the overall objective when each customer is considered separately.

Next we estimate the time complexity of the heuristic. For each customer $i \in M$, the time required by DP1 is $O(n_i^2 b)$. Thus the overall time needed in Step 1 of the heuristic is $O(n^2 b)$. In Step 2, there are no more than n values in the set Γ. For each value $x \in \Gamma$, $k_i(x)$ for customer $i \in M$ can be found by doing a line search for $T^i_{\max}(k)$ corresponding to values of k between $\lceil n_i/b \rceil$ and n_i. This takes $O(\log n_i)$ steps. Therefore, the total complexity of this for all the customers is $O(\Sigma_{i \in M} \log n_i)$ which is bounded by $O(m \log n)$. Step 3 requires $O(n \log n)$ steps. Hence the overall complexity of the heuristic is bounded by $O(n^2 b + nm \log n)$.

LEMMA 6: Denote the optimal objective value of the problem P2 as F^*. Then

$$F^* \geq \alpha x^* + (1 - \alpha) \sum_{i \in M} f_i k_i(x^*) - \alpha((b - 1)p_{\max} + 2t_{\max} - 2t_{\min}), \quad (5)$$

where x^* is as defined in (4), $p_{\max} = \max\{p_{ij} | (i, j) \in N\}$, $t_{\max} = \max\{t_i | i \in M\}$, and $t_{\min} = \min\{t_i | i \in M\}$.

PROOF: Given an optimal schedule S^* of the problem P2 that follows Lemma 5, we construct a schedule S' such that there are same number of delivery batches in S' as in S^*, and each batch in S' contains the same number of orders from the same customer as in the corresponding batch in S^*. But in S', the orders from each customer are scheduled in their EDD sequence. So the actual set of orders in any batch in S' may be different from that in the corresponding batch in S^*. Each batch in S' is shipped at a time that is the sum of the completion time C^{SEDD}_{ij} of the first order (i, j) in the batch and the total processing time of the other orders in the batch. Note that schedule S' may not be feasible to P2 because there may be overlap between batches of orders from different customers. Also, note that schedule S' gets enumerated implicitly in Step 1 of the heuristic. We can easily see that the total distribution cost in S' is the same as that in S^*.

Let $D_{ij}(S')$ and $D_{ij}(S^*)$ denote the delivery time of order (i, j) in S' and S^* respectively. Similarly, let $T_{ij}(S') = \max\{0, D_{ij}(S') - d_{ij}\}$ and $T_{ij}(S^*) = \max\{0, D_{ij}(S^*) - d_{ij}\}$ denote the tardiness of order (i, j) in S' and S^*, respectively. The maximum tardiness of orders in S' is determined solely based on the first order in each batch. Consider the first order, denoted as (i, u) in a particular batch of orders from customer $i \in M$ in schedule S'. Let τ denote the sum of processing times of all the orders except (i, u) in the batch. Evidently, $\tau \leq (b - 1)p_{\max}$. Let Q denote the set of all orders up to and including order (i, u) in the SEDD sequence of N. Clearly, $C^{SEDD}_{iu} = \Sigma_{(i,j) \in Q} p_{ij}$. Thus we have

$$D_{iu}(S') = C^{SEDD}_{iu} + \tau + t_i = \sum_{(i,j) \in Q} p_{ij} + \tau + t_i \leq \sum_{(i,j) \in Q} p_{ij} + (b - 1)p_{\max} + t_{\max}. \quad (6)$$

Now consider the last batch in S^* that contains an order from Q. Suppose that this batch belongs to customer $k \in M$. Denote the first order in this batch as (k, ν). Note that (k, ν) belongs to Q. We have the following:

$$D_{k\nu}(S^*) \geq \sum_{(i,j) \in Q} p_{ij} + t_{\min}. \quad (7)$$

From (6) and (7), we have

$$D_{iu}(S') - D_{k\nu}(S^*) \leq (b - 1)p_{\max} + t_{\max} - t_{\min}. \quad (8)$$

Since (i, u) is the last order in Q, the shipping due dates follow the relation: $d'_{iu} \geq d'_{k\nu}$. Therefore, $d_{iu} - t_i \geq d_{k\nu} - t_k$, which implies that $d_{iu} - d_{k\nu} \geq t_{\min} - t_{\max}$. This, along with (8), implies that

$$(D_{iu}(S') - d_{iu}) - (D_{k\nu}(S^*) - d_{k\nu}) \leq (b - 1)p_{\max} + 2t_{\max} - 2t_{\min}. \quad (9)$$

Inequality (9) implies that

$$T_{iu}(S') - T_{k\nu}(S^*) \leq (b-1)p_{\max} + 2t_{\max} - 2t_{\min}. \tag{10}$$

Inequalities (8)–(10) are valid for the first order (i, u) of any batch in S'. DP1 in the heuristic H1 considers all choices of batch configurations at every customer including the configurations that appear in schedule S'. In Step 2 of H1, $\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*)$ is obtained by putting together these values in such a way that the combined value for all the customers is minimum. Therefore, this value will not be greater than the one obtained by combining the values at individual customers in S'. Hence,

$$\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*) \leq F^* + \alpha((b-1)p_{\max} + 2t_{\max} - 2t_{\min}). \tag{11}$$

This completes the proof. □

LEMMA 7: Let F^{H1} represent the objective value of the schedule π obtained by the heuristic H1. Then,

$$F^{\mathrm{H1}} \leq \alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*) + \alpha(b-1)(m-1)p_{\max}, \tag{12}$$

where x^* and $k_i(x^*)$ are defined in Step 2 of the heuristic.

PROOF: Let us consider an arbitrary batch ω of customer $i \in M$ in the schedule π generated in Step 3 of the heuristic. Denote the first order in the batch as (i, u). Let $D_{iu}(\pi_i)$ and $D_{iu}(\pi)$ denote the delivery times of this batch in the schedule π_i generated in Step 2 and in the schedule π, respectively. By the definition of problem AUX_i and the fact that π_i is optimal to the problem AUX_i with $k_i(x^*)$ delivery batches, we have

$$D_{iu}(\pi_i) = C_{iu}^{SEDD} + \sum_{(i,j)\in\omega} p_{ij} - p_{iu} + t_i. \tag{13}$$

Since, in schedule π, the first orders of the batches form SEDD sequence and the orders from each customer are sequenced in EDD order, there can be at most one batch scheduled before ω from every customer other than i, that contains orders which come after (i, u) in the SEDD sequence of all the orders of N. Even in those batches, there should be at least one order in each batch that comes before order (i, u) in the SEDD sequence of all the orders of N. Therefore, we have

$$D_{iu}(\pi) \leq \left(C_{iu}^{SEDD} + \sum_{(i,j)\in\omega} p_{ij} - p_{iu} + t_i\right) + (b-1)(m-1)p_{\max}. \tag{14}$$

Let $T_{iu}(\pi_i)$ and $T_{iu}(\pi)$ denote the tardiness of the batch ω in schedules π_i and π, respectively. Then by (13) and (14), we have

$$T_{iu}(\pi) - T_{iu}(\pi_i) \leq D_{iu}(\pi) - D_{iu}(\pi_i) \leq (b-1)(m-1)p_{\max}. \tag{15}$$

Here the first relation is not an equality to take into account cases where the batch is delivered before its due date. Relation (15) is valid for all the batches and hence $(b-1)(m-1)p_{\max}$ gives an upper bound on the difference in maximum tardiness possible between the schedules π_i and π, for every $i \in M$. Since in the schedule π, the batch configurations $\Lambda_i(k_i(x^*))$ generated in Step 2 is used for each customer $i \in M$, the total distribution cost incurred by the orders of N_i in π is exactly $f_i k_i(x^*)$, which is the same as in schedule π_i. Since the sum of the objective values of the schedules π_i over all $i \in M$ is $\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*)$, the objective value F^{H1} of the schedule π satisfies (12). This completes the proof. □

THEOREM 3: If order processing times p_{ij}, delivery times t_i, and delivery costs f_i are drawn from distributions over finite intervals $[L_1, U_1]$, $[L_2, U_2]$, and $[L_3, U_3]$, respectively, with $0 < L_1 \leq U_1 < \infty$, $0 < L_2 \leq U_2 < \infty$, and $0 < L_3 \leq U_3 < \infty$, then the solution generated by the heuristic H1 is asymptotically optimal for problem P2 with $0 < \alpha < 1$ when n goes to infinity, with m and b fixed.

PROOF: As in the proofs of Lemmas 6 and 7, let F^* and F^{H1} represent the optimal objective value of the problem P2, and the objective value of the schedule π generated by the heuristic respectively. Combining the inequalities (5) and (12), we have

$$F^{\mathrm{H1}} \leq F^* + \alpha((b-1)p_{\max} + 2t_{\max} - 2t_{\min}) + \alpha(b-1) \times (m-1)p_{\max} = F^* + \alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min}),$$

which means that

$$\frac{F^{\mathrm{H1}} - F^*}{F^*} \leq \frac{\alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min})}{F^*}. \tag{16}$$

For fixed b, the total distribution cost, and therefore F^*, increase to infinity as the number of orders n increases to infinity. Since $p_{\max} < \infty$, $t_{\max} < \infty$, $\alpha < 1$, and m and b are fixed, $\alpha((b-1)mp_{\max}+2t_{\max}-2t_{\min})$ is finite. Hence, by (16), we have

$$\lim_{n\to\infty}\frac{F^{\mathrm{H1}}-F^*}{F^*}\le\lim_{n\to\infty}\frac{\alpha((b-1)mp_{\max}+2t_{\max}-2t_{\min})}{F^*}=0.$$

This shows the theorem. □

When a batch is being formed for a particular customer, heuristic H1 ignores the effect due to the batches formed for the orders of all other customers. This leads to an increase in the tardiness value when we move from Step 2 to Step 3. One way to limit this increase is to reduce the maximum allowed batch size for a few customers and then run the heuristic again. Doing this may lead to an increase in the number of delivery batches for these customers. But, on the other hand, it may also lead to smaller batches being formed, which helps reduce the maximum tardiness value. If we reduce the maximum allowed batch size for those customers that have low transportation costs, the reduction in the maximum tardiness may outweigh the increase in the total distribution cost. Another way to improve an existing solution is to replace the order completion times (C_{ij}^{SEDD}) in Step 1 with the actual order completion times obtained using the heuristic. In doing so, we replace the hypothetical completion time values of Step 1 with something that is more realistic and accounts for the batching of orders. These values can be used to obtain the sequences for Step 2 and subsequently Step 3. This approach favors the formation of smaller batches whenever the delay due to batching starts to accumulate. We include these two procedures as improvement schemes while implementing the heuristic.

In the next section, we describe how to obtain a tight lower bound for the problems P2 and P2A using column generation.

3.3. Evaluating the Heuristic

To evaluate the performance of heuristic H1, we need to obtain tight lower bounds. Though Lemma 6 provides a valid lower bound of the optimal objective value of problem P2, our computational tests show that this lower bound is very loose. In this section, we present a linear programming based procedure for obtaining tight lower bounds. We will give an IP formulation for a problem closely related to P2 and describe a procedure for obtaining valid lower bounds using the LP relaxation of this IP formulation.

3.3.1. A Sequential Search Approach for Obtaining Lower Bounds

We first consider a closely related problem, denoted as CRP, which is to minimize the total distribution cost subject to the constraint that the maximum tardiness of the orders, $T_{\max}$, is no more than a given value T_0. We will see later that a lower bound of the optimal objective value of the problem P2 can be obtained by utilizing lower bounds of the optimal objective values of the problem CRP with various values of T_0.

We first formulate CRP as an IP problem. Let Ω_i be the set of all feasible schedules for *a single batch of orders* from customer i, for $i\in M$. A feasible schedule $\omega\in\Omega_i$ for a batch of orders from customer $i\in M$ specifies which orders are in the batch, the starting time of the first order, the processing completion time of the last order, and the time these orders are delivered to the customer. All feasible schedules for a batch satisfy the constraint that the maximum delivery tardiness of the orders in the batch is no more than the given value T_0. We define the following parameters:

$$Q=\text{total processing times of all the orders}=\sum_{(i,j)\in N}p_{ij},$$

$$g_\omega=\text{transportation cost of schedule }\omega\in\Omega_i=f_i,$$

$$a_{j\omega}=1\quad\text{if order }(i,j)\text{ is covered in schedule }\omega\in\Omega_i\text{ and 0 otherwise,}$$

$$\tau_{t\omega}=1\quad\text{if time interval }[t,t+1]\text{ is covered by schedule }\omega\in\Omega_i\text{ and 0 otherwise.}$$

Also, we define a variable x_ω to be 1 if schedule $\omega\in\Omega_i$ is used and 0 otherwise. Then CRP can be formulated as the following set partitioning type binary IP formulation:

$$[\mathrm{SP}]\qquad \operatorname{Min}\sum_{i\in M}\sum_{\omega\in\Omega_i}g_\omega x_\omega \tag{17}$$

subject to

$$\sum_{\omega\in\Omega_i}a_{j\omega}x_\omega=1,\qquad i\in M\text{ and }j=1,\ldots,n_i \tag{18}$$

$$\sum_{i\in M}\sum_{\omega\in\Omega_i}\tau_{t\omega}x_\omega=1,\qquad t=0,1,\ldots,Q-1. \tag{19}$$

$$x_\omega\in\{0,1\},\qquad \omega\in\bigcup_{i\in M}\Omega_i. \tag{20}$$

In [SP], the objective function is to minimize the total distribution cost. Equation (18) ensures that each order gets covered exactly once by some schedule. Equation (19) ensures that each time slot in the interval $[0, Q]$ is covered exactly once.

We denote the LP relaxation of [SP] as [LSP] where the constraint (20) is replaced by "$x_\omega \geq 0$." Clearly, the optimal objective value of [LSP] is a lower bound of that of CRP. We will develop a column generation based algorithm to solve [LSP] in Section 3.3.2.

Next we describe how to get a lower bound for problem P2 by solving [LSP]. Let Ψ denote the set of all possible values of $T_{\max}$ in a feasible schedule of problem P2. Since the order processing times and the due dates are integer valued, there is only a finite number of values in the set Ψ. The minimum value of $T_{\max}$, denoted as $T_{\max}^{\min}$, is obtained when the orders are processed in their SEDD order and then shipped individually. The maximum value possible for $T_{\max}$, denoted as $T_{\max}^{\max}$, is given by the maximum tardiness of orders if they are processed in their SEDD order and all are delivered in full batches except possibly last several orders that are delivered in a partial batch. Let $LB_{\mathrm{CRP}}(T_0)$ denote the optimal objective value of [LSP] with $T_{\max}$ no more than T_0. Then it can be seen that $LB_{\mathrm{P2}} = \min\{\alpha T_0 + (1 - \alpha) LB_{\mathrm{CRP}}(T_0) | T_0 \in \Psi\}$ is a valid lower bound of P2. However, it may not be necessary to solve [LSP] for each value of T_0 in Ψ.

Next we give a procedure for getting a lower bound for P2 based on the above observations.

Algorithm A4

Step 0: Set the lower bound, $LB = \infty$. Set $T_0 = T_{\max}^{\max}$.
Step 1: Solve [LSP]. Let the optimal objective value be Z^*.
Step 2: Obtain the actual maximum tardiness value corresponding to the current optimal solution of [LSP], denoted as $T_{\max}^a$, which is defined to be the maximum tardiness among all the batches $\omega \in \Omega_i$, $i \in M$, whose corresponding variable in [LSP] has a positive value. If $\alpha T_{\max}^a + (1 - \alpha) Z^* < LB$, set $LB = \alpha T_{\max}^a + (1 - \alpha) Z^*$.
Step 3: Set $T_0 = T_{\max}^a - 1$. If $T_0 \geq T_{\max}^{\min}$, go to Step 1. Otherwise, STOP and LB gives a lower bound for P2.

It should be noted that due to the way we reduce T_0 in Step 3, we need not solve the [LSP] for each and every integer value of $T_{\max}$ between $T_{\max}^{\min}$ and $T_{\max}^{\max}$.

3.3.2. *Column Generation for Solving [LSP]*

Due to the large number of columns in the formulation [LSP], it is impractical to solve it directly. We resort to a column generation approach. In each iteration of the column generation approach, we first solve a *master problem,* which is [LSP] with only a subset of the columns (i.e., single-batch schedules) from each set Ω_i. Then we use the dual variable values of the master problem to form a *subproblem* corresponding to each customer to find schedules $\omega \in \Omega_i$ with a negative reduced cost. We add to the master problem several columns with negative reduced costs generated while solving these subproblems. Then we solve the master problem again. We repeat this till there are no more columns that can be generated by solving the subproblems that give negative reduced costs. At that point, we have the optimal solution to [LSP].

An initial set of columns for [LSP] can be generated by processing all the orders in the SEDD sequence and delivering them individually. We use ρ_{ij} and γ_t to denote the dual variable value corresponding to the constraint set (18) and (19) of [LSP], respectively. Then the reduced cost r_ω of a column corresponding to $\omega \in \Omega_i$, $i \in M$, is given by $r_\omega = g_\omega - \sum_{j\in\omega} a_{j\omega}\rho_{ij} - \sum_{t\in\omega} \tau_{t\omega}\gamma_t$. The ith subproblem (for customer $i \in M$) is to select columns $\omega \in \Omega_i$ with the minimum value of r_ω. Before we present an algorithm for solving this subproblem, we first note that the following are true for each column $\omega \in \Omega_1 \cup \cdots \cup \Omega_m$:

a. All the orders in the column belong to the same customer.
b. There are no more than b orders in the column.
c. The maximum tardiness for the orders in the column is not greater than the given value T_0.
d. The completion time of the last order in the column must be no more than Q, the total processing times of all the orders in N.

By Lemma 2, for the special case of the problem P2A, in addition to a–d above, the orders in a batch will be consecutive orders from the EDD sequence for the customer. For the general case of the problem P2, the orders in a batch of a customer may not be consecutive in the EDD sequence for that customer. Hence the number of potential columns is much higher in the case of P2.

In the following, we develop a dynamic programming algorithm for solving the ith subproblem, for any $i \in M$. The algorithm is presented for the case of problem P2, and can be simplified slightly for the case of P2A.

Algorithm DP2

Define the value function $F(u, v, q, t)$ as the minimum reduced cost of a schedule of a batch with q orders that contains order (i, u) as the first order and (i, v) as the last order which is completed at time t, where $v \geq u$, $q \leq b$, and $t \leq \min\{Q, d_{iu} + T_0 - t_i\}$.

Initial values

$$F(u, u, 1, t) = f_i - \rho_{iu} - \sum_{h=t-p_{iu}}^{t-1} \gamma_h,$$

$$\text{for } u \in \{1, \ldots, n_i\}, t \in \{p_{iu}, \ldots, \min\{Q, d_{iu} + T_0 - t_i\}\},$$

$$F(u, u, q, t) = \infty, \qquad \text{if } q > 1$$
$$\text{or } t \notin \{p_{iu}, \ldots, \min\{Q, d_{iu} + T_0 - t_i\}\}.$$

Recursive relations

For $u \in \{1, \ldots, n_i\}$, $\nu \in \{u, \ldots, n_i\}$, $q \in \{1, \ldots, b\}$, and $t \in \{p_{iu}, \ldots, \min\{Q, d_{iu} + T_0 - t_i\}\}$,

$$F(u, \nu, q, t) = \min\left\{F(u, k, q-1, t - p_{i\nu}) - \rho_{i\nu} - \sum_{h=t-p_{i\nu}}^{t-1} \gamma_h \middle| k = u, \ldots, \nu - 1\right\} \quad (21)$$

Optimal solution

For a fixed $u \in \{1, \ldots, n_i\}$, an optimal schedule with order (i, u) as the first order is found by minimizing $F(u, \nu, q, t)$ over all possible (ν, q, t) satisfying: $\nu \geq u$, $1 \leq q \leq b$, and $p_{iu} \leq t \leq \min\{Q, d_{iu} + T_0 - t_i\}$. Among the n_i such schedules found, the one with the minimum F is optimal to subproblem i.

LEMMA 8: Algorithm DP2 solves the ith subproblem optimally in time $O(n_i^3 bQ)$.

PROOF: The term $-\rho_{i\nu} - \sum_{h=t-p_{i\nu}}^{t-1} \gamma_h$ in the recursive relation (21) is the total contribution to the reduced cost made by order (i, ν) which is scheduled in the interval $[t - p_{i\nu}, t)$. The recursive relation enumerates all possible orders (i, k) that can be scheduled before the current last order (i, ν). Thus the optimality is guaranteed. There are a total of $O(n_i^2 bQ)$ states in the dynamic program, and it takes $O(n_i)$ time to compute the value for each state; thus the overall time needed by the algorithm is $O(n_i^3 bQ)$. □

Algorithm DP2 can be made more efficient for the ith subproblem in the case of problem P2A. As we pointed out earlier, for problem P2A, each delivery batch for a customer consists of consecutive orders from the EDD sequence of the orders of that customer. Thus, $q = \nu - u + 1$ in each state of the dynamic program, which means that we can actually drop q from each state and the recursive relation (21) can be modified as $F(u, \nu, t) = F(u, \nu - 1, t - p_{i\nu}) - \rho_{i\nu} - \sum_{h=t-p_{i\nu}}^{t-1} \gamma_h$. The time complexity of the DP becomes $O(n_i bQ)$.

In solving [LSP], some techniques can be used to speed up the algorithm. For example, it is not necessary to run DP2 for every choice of the first order (i, u) in each iteration of the column generation. We use a cyclic scheme for running DP2 in which we start with order $(i, 1)$ as the first order in the first iteration of the column generation and continue with $(i, 2)$ as the first order and so on until we generate a certain number of columns with a negative reduced cost, and then in the next iteration of the column generation we start from the last order considered in the last iteration as the first order in the batch.

3.3.3. Computational Results

In this section, we describe the computational experiment to evaluate the performance of the heuristic H1. The results obtained from the heuristic are compared with the lower bound generated using the column generation approach described in Sections 3.3.1 and 3.3.2. Test problems are randomly generated as follows:

a. Total number of orders $n \in \{20, 30, 40\}$; number of customers $m \in \{2, 4\}$; shipment capacity $b \in \{2, 4\}$. The orders are assigned to customers randomly, with each order having an equal probability for getting assigned to a particular customer.
b. Order processing times p_{ij} are independently generated from a uniform distribution $U[1, 10]$.
c. Transportation times t_i are independently generated from a uniform distribution $U[10, 100]$; transportation cost per delivery shipment f_i is set equal to the transportation times.
d. Order due dates d_{ij} are independently generated from a uniform distribution $U[p_{\min} + t_{\min}, \lambda((p_{\min} + p_{\max})n/2 + (t_{\min} + t_{\max})/2)]$, where $p_{\min}$ and $p_{\max}$ are the minimum and maximum order processing times respectively, and $t_{\min}$ and $t_{\max}$ are the minimum and maximum transportation times. Value of λ determines how tight the due dates are. We test three different levels: $\lambda \in \{0.5, 1, 1.5\}$. Two types of due dates are considered: special (corresponding to problem P2A) and general (corresponding to problem P2). For the special case, due dates are made agreeable with the processing times. To ensure this, the same random seed is used to generate both the processing time and the due date for an order. For the general case, the due dates are generated independent of the processing times.
e. Weighting parameter in the objective function $\alpha \in \{0.5, 0.75, 0.9\}$. The values for α are chosen so that we are able to demonstrate the effect of varying weights on the production and distribution part. The

Table 1. Computational results for heuristic H1.

Problem				P2A		P2		Overall	
n	m	b	α	Avg gap	Max gap	Avg gap	Max gap	Avg gap	Max gap
			0.5	0.00%	0.00%	3.58%	8.97%	1.79%	8.97%
20	2	2	0.75	0.00%	0.00%	3.10%	7.09%	1.55%	7.09%
			0.9	0.11%	1.66%	1.88%	4.26%	0.99%	4.26%
			0.5	0.09%	1.28%	9.49%	17.97%	4.79%	17.97%
20	2	4	0.75	0.37%	3.06%	6.61%	15.13%	3.49%	15.13%
			0.9	0.74%	4.61%	4.19%	9.25%	2.46%	9.25%
			0.5	0.03%	0.41%	9.15%	21.18%	4.59%	21.18%
20	4	2	0.75	0.07%	1.03%	6.88%	17.48%	3.47%	17.48%
			0.9	0.53%	6.00%	4.83%	11.47%	2.68%	11.47%
			0.5	0.18%	1.29%	9.65%	14.29%	4.91%	14.29%
20	4	4	0.75	0.39%	2.75%	3.47%	9.04%	1.93%	9.04%
			0.9	1.18%	4.96%	2.50%	7.79%	1.84%	7.79%
			0.5	0.00%	0.00%	5.90%	6.34%	2.95%	6.34%
30	2	2	0.75	0.00%	0.00%	4.88%	6.21%	2.44%	6.21%
			0.9	0.00%	0.00%	3.20%	6.00%	1.60%	6.00%
			0.5	0.15%	1.42%	9.53%	17.07%	4.84%	17.07%
30	2	4	0.75	0.37%	3.61%	7.03%	12.96%	3.70%	12.96%
			0.9	0.98%	4.93%	5.14%	19.01%	3.06%	19.01%
			0.5	0.00%	0.00%	4.14%	6.09%	2.07%	6.09%
30	4	2	0.75	0.00%	0.00%	3.74%	6.75%	1.87%	6.75%
			0.9	0.20%	1.28%	3.86%	7.94%	2.03%	7.94%
			0.5	0.42%	2.11%	11.84%	18.66%	6.13%	18.66%
30	4	4	0.75	1.29%	5.26%	7.58%	11.23%	4.44%	11.23%
			0.9	2.45%	10.75%	5.94%	11.26%	4.19%	11.26%
			0.5	0.00%	0.00%	0.18%	0.90%	0.09%	0.90%
40	2	2	0.75	0.00%	0.00%	0.56%	2.57%	0.28%	2.57%
			0.9	0.00%	0.00%	1.30%	6.52%	0.65%	6.52%
			0.5	0.27%	2.04%	2.67%	10.02%	1.47%	10.02%
40	2	4	0.75	0.62%	5.41%	3.72%	9.35%	2.17%	9.35%
			0.9	1.59%	14.58%	4.63%	12.08%	3.11%	14.58%
			0.5	0.03%	0.32%	5.54%	7.30%	2.78%	7.30%
40	4	2	0.75	0.08%	0.89%	4.88%	6.16%	2.48%	6.16%
			0.9	0.61%	3.91%	4.08%	6.86%	2.34%	6.86%
			0.5	0.44%	3.05%	11.74%	16.69%	6.09%	16.69%
40	4	4	0.75	1.64%	6.50%	8.40%	13.22%	5.02%	13.22%
			0.9	3.69%	19.94%	5.89%	8.61%	4.79%	19.94%

contribution due to the production part is small compared to that of the distribution part when $\alpha = 0.5$, the two are comparable when $\alpha = 0.75$, and when $\alpha = 0.9$, the production part dominates.

For each of the 108 combinations of (n, m, b, λ, α), we test ten different randomly generated problem instances. Five of these are with special due dates (i.e., for problem P2A) while the remaining five are assigned general due dates (i.e., for problem P2). Hence we test a total of 1080 problem instances. The programs were written in C, and all LP problems involved were solved by calling the LP solver of CPLEX 8.0. The code was run on a PC with a 1.5-GHz Pentium IV processor and 512-MB memory. Every problem instance was successfully solved by the heuristic with no more than 1 CPU second. On the other hand, the computational time for the column generation procedure was observed to increase at an exponential rate. For problem instances with general due dates, it took around 45 CPU minutes per instance when the number of orders was set at 40. Moreover, the computer ran out of memory when the number of orders was increased beyond 40 for shipment capacity 4.

Table 1 reports both average and maximum relative gaps between the objective values Z_{H1} of the solutions generated by the heuristic H1 and the lower bound LB_{P2} generated by solving [LSP] by the column generation approach. The relative gap is defined as $[(Z_{H1} - LB_{P2})/LB_{P2}] \times 100\%$. Clearly, the relative gap defined here is an upper bound of the actual relative gap between the heuristics solutions and the optimal solution. Each entry in the columns "Avg Gap" of Table 1 is the average relative gap over the random test problems with the corresponding (n, m, b, α) combination. Note that the results corresponding to different values of λ have been put together for ease of presentation. In order to account for this, we have presented the maximum gap values in each category along with the average.

These results demonstrate that the heuristic is capable of generating near optimal solutions for most problems tested. Due to the excessive computational time needed for getting lower bounds by the column generation approach for larger problems, we did not test on larger problems. However, by the asymptotic optimality of the heuristic (Theorem 3), it can be expected that the heuristic will also perform well for larger problems.

Some other conclusions can be made based on the results in Table 1. It can be seen that in general, for a given number of orders n, the performance of the heuristic deteriorates as the maximum allowed batch size b increases. This can be explained by the fact that when we increase the maximum allowed batch size b, the schedules $\pi_1, \ldots, \pi_m$ generated from individual customers in Step 2 are more likely to overlap with one another and hence the final combined schedule π generated in Step 3 is more different from these individual schedules, which leads to a negative effect on the performance of the heuristic. The heuristic performs considerably better when the due dates are proportional to the processing times. This is expected since this heuristic is developed based on this special case. In general, when the number of orders n is high compared to the number of customers m or the maximum allowed batch size b, the heuristic is more likely to generate near optimal solutions.

4. VALUE OF PRODUCTION–DISTRIBUTION INTEGRATION

The problems we have studied integrate order processing and order delivery decisions in order to optimize a combined objective function. However, production and distribution decisions are often treated separately and sequentially in the literature. Most production scheduling models consider order processing only, whereas most distribution models assume that orders to be delivered have been processed and are only concerned with the total distribution cost.

In this section, we analyze the value of such integration. We compare the integrated scheduling approach considered in this paper with two typical sequential approaches that treat order processing and order delivery sequentially with no or only partial integration. In both the sequential approaches, the production part assumes that each order $(i, j) \in N$ once completed processing is delivered to its customer immediately without considering the possibility of delivery consolidation with other orders, i.e., $D_{ij} = C_{ij} + t_i$, and tries to minimize the maximum tardiness of orders $T_{\max}$. Clearly, scheduling the orders in the SEDD order is optimal in this part. The distribution part of the first sequential approach tries to minimize the distribution cost G only, given the SEDD processing sequence of orders. As a result, the orders completed in the production part are delivered to the customers using a minimum possible number of shipments. Thus, for each customer $i \in M$, the orders $(i, (k-1)b+1), \ldots, (i, kb)$ are delivered together as the kth shipment for $k = 1, \ldots, \lfloor n_i/b \rfloor$, and the remaining orders as the last shipment. The total overall cost $\alpha T_{\max} + (1-\alpha)G$ of this approach can be calculated accordingly. In this sequential approach, production and distribution are treated totally separately and there is no integration at all. We call this the *sequential approach without integration.*

In the second sequential approach, given the SEDD processing sequence of the orders, the distribution part tries to minimize the integrated objective function $\alpha T_{\max} + (1-\alpha)G$. Since the production part does not consider this overall objective, production and distribution is only partially integrated in this sequential approach. In the distribution part of this approach, an optimal distribution schedule can be obtained by applying the first two steps of heuristic H1 to the given SEDD production sequence of the orders with the following two modifications: (i) in the single-customer auxiliary problem AUX_i for customer $i \in M$, the departure time of a delivery batch B is redefined simply as the completion time C_{ij}^{SEDD} of the last order (i, j) in B; (ii) the recursive relation (3) of DP1 is replaced by the following:

$$F(j, k) = \min_{1 \le q \le \min\{b, j\}} \{\max\{F(j-q, k-1), \max\{0, C_{ij}^{SEDD} + t_i - d_{i,j-q+1}\}\}\}.$$

Then the total cost of this approach is given by (4). We call this the *sequential approach with partial integration.*

We conduct a computational experiment to evaluate the possible improvement that can be achieved for the integrated objective function, $\alpha T_{\max} + (1-\alpha)G$, from the two sequential approaches to the integrated approach. More specifically, we calculate the relative gap of the objective value of the solution generated by a sequential approach and that generated by the heuristic H1: $[(Z_{SEQ} - Z_{H1})/Z_{SEQ}] \times 100\%$, where Z_{SEQ} and Z_{H1} represent the objective values of the solutions found by a sequential approach and the heuristic H1, respectively. Since the heuristic solution is used

Table 2. Relative improvement from the sequential approach without integration to the integrated approach.

Problem				P2A		P2		Overall	
n	m	b	α	Avg gap	Max gap	Avg gap	Max gap	Avg gap	Max gap
			0.5	0.55%	2.35%	0.56%	2.63%	0.56%	2.63%
25	2	2	0.75	1.45%	6.36%	1.46%	7.33%	1.45%	7.33%
			0.9	3.26%	14.80%	3.37%	18.18%	3.32%	18.18%
			0.5	2.56%	7.56%	3.15%	8.92%	2.86%	8.92%
25	2	4	0.75	5.70%	16.79%	7.09%	18.91%	6.40%	18.91%
			0.9	10.97%	37.56%	13.38%	30.19%	12.17%	37.56%
			0.5	4.41%	17.43%	3.86%	8.06%	4.14%	17.43%
25	4	2	0.75	9.94%	36.49%	8.99%	18.91%	9.46%	36.49%
			0.9	17.37%	57.42%	16.31%	34.28%	16.84%	57.42%
			0.5	10.40%	31.35%	10.05%	15.19%	10.23%	31.35%
25	4	4	0.75	19.67%	53.02%	19.59%	29.88%	19.63%	53.02%
			0.9	30.46%	72.32%	30.62%	47.79%	30.54%	72.32%
			0.5	0.16%	0.90%	0.37%	2.92%	0.27%	2.92%
50	2	2	0.75	0.46%	2.60%	0.99%	7.72%	0.72%	7.72%
			0.9	1.22%	6.90%	2.39%	17.05%	1.80%	17.05%
			0.5	0.84%	3.29%	1.75%	5.26%	1.29%	5.26%
50	2	4	0.75	2.38%	8.93%	4.32%	12.38%	3.35%	12.38%
			0.9	5.49%	20.83%	9.90%	25.54%	7.70%	25.54%
			0.5	0.73%	5.32%	2.18%	8.15%	1.45%	8.15%
50	4	2	0.75	1.94%	13.85%	5.38%	19.57%	3.66%	19.57%
			0.9	4.46%	29.84%	10.66%	36.72%	7.56%	36.72%
			0.5	6.23%	19.03%	7.30%	15.53%	6.77%	19.03%
50	4	4	0.75	13.41%	37.20%	15.16%	31.73%	14.28%	37.20%
			0.9	23.81%	63.20%	25.10%	52.02%	24.46%	63.20%
			0.5	0.26%	1.62%	0.18%	0.63%	0.22%	1.62%
100	2	2	0.75	0.71%	4.42%	0.51%	1.79%	0.61%	4.42%
			0.9	1.69%	10.46%	1.49%	4.79%	1.59%	10.46%
			0.5	0.58%	3.27%	0.96%	2.99%	0.77%	3.27%
100	2	4	0.75	1.56%	8.26%	2.38%	7.63%	1.97%	8.26%
			0.9	4.04%	15.61%	6.01%	15.83%	5.03%	15.83%
			0.5	0.93%	4.04%	1.30%	2.73%	1.11%	4.04%
100	4	2	0.75	2.44%	10.81%	3.47%	7.40%	2.96%	10.81%
			0.9	5.54%	24.48%	8.03%	17.18%	6.78%	24.48%
			0.5	2.54%	5.83%	4.95%	10.72%	3.75%	10.72%
100	4	4	0.75	6.39%	15.25%	11.32%	23.14%	8.85%	23.14%
			0.9	14.49%	34.47%	20.10%	41.97%	17.29%	41.97%

instead of the optimal solution for the integrated approach, this relative gap is a lower bound of the relative gap between the sequential approaches and the optimal integrated approach. This gap gives an indication of the percentage savings that we can obtain by resorting to an integrated approach.

Test problems are generated exactly the same way as in Section 3.3.3 except that the number of orders $n \in \{25, 50, 100\}$. Tables 2 and 3 report the average and maximum gap values for the test problems between the two sequential approaches and the integrated approach. Over all the test problems, the average gap between the first sequential approach and the integrated approach is 6.08% for P2A and 7.35% for P2, whereas that between the second sequential approach and the integrated approach is 1.82% for P2A and 2.20% for P2. This means that the sequential approach with partial integration provides much closer solutions to optimal solutions than the sequential approach without integration. This shows that even just partial integration enhances overall solutions significantly. The results show that the gaps could be as high as 72.32% for the first sequential approach and 21.22% for the second. We also note that since the heuristic is not guaranteed to give the optimal solution for the integrated problem, theoretically it is possible for the sequential approaches to beat the heuristic. But as the results show, this does not happen very often. Out of 1080 in-

Table 3. Relative improvement from the sequential approach with partial integration to the integrated approach.

Problem				P2A		P2		Overall	
n	m	b	α	Avg gap	Max gap	Avg gap	Max gap	Avg gap	Max gap
			0.5	0.34%	2.35%	0.33%	2.17%	0.33%	2.35%
25	2	2	0.75	0.72%	4.57%	0.87%	5.71%	0.79%	5.71%
			0.9	0.38%	2.16%	1.40%	7.80%	0.89%	7.80%
			0.5	1.45%	7.56%	1.23%	3.80%	1.34%	7.56%
25	2	4	0.75	1.42%	10.66%	2.38%	5.73%	1.90%	10.66%
			0.9	1.14%	6.69%	0.64%	4.21%	0.89%	6.69%
			0.5	3.21%	7.22%	1.23%	3.96%	2.22%	7.22%
25	4	2	0.75	3.96%	7.58%	2.41%	8.82%	3.18%	8.82%
			0.9	3.60%	9.18%	2.23%	9.70%	2.92%	9.70%
			0.5	6.10%	21.22%	7.07%	15.19%	6.59%	21.22%
25	4	4	0.75	8.74%	19.85%	8.78%	20.33%	8.76%	20.33%
			0.9	8.21%	18.60%	5.20%	13.78%	6.71%	18.60%
			0.5	0.07%	0.50%	0.35%	2.92%	0.21%	2.92%
50	2	2	0.75	0.18%	1.39%	0.78%	5.64%	0.48%	5.64%
			0.9	0.47%	3.46%	0.93%	4.52%	0.70%	4.52%
			0.5	0.42%	2.65%	1.11%	3.87%	0.76%	3.87%
50	2	4	0.75	0.93%	4.38%	1.31%	4.09%	1.12%	4.38%
			0.9	0.88%	6.54%	0.99%	7.70%	0.94%	7.70%
			0.5	0.56%	3.96%	1.18%	3.50%	0.87%	3.96%
50	4	2	0.75	1.02%	4.39%	2.86%	8.81%	1.94%	8.81%
			0.9	1.33%	4.25%	3.06%	7.05%	2.19%	7.05%
			0.5	3.88%	10.32%	3.79%	8.72%	3.83%	10.32%
50	4	4	0.75	5.17%	13.21%	5.86%	12.05%	5.51%	13.21%
			0.9	4.55%	11.77%	5.47%	11.93%	5.01%	11.93%
			0.5	0.04%	0.43%	0.13%	0.59%	0.08%	0.59%
100	2	2	0.75	0.12%	1.22%	0.30%	1.69%	0.21%	1.69%
			0.9	0.09%	1.92%	0.43%	1.92%	0.26%	1.92%
			0.5	0.37%	2.19%	0.41%	1.25%	0.39%	2.19%
100	2	4	0.75	0.45%	2.68%	0.81%	2.50%	0.63%	2.68%
			0.9	0.41%	3.07%	1.20%	4.20%	0.81%	4.20%
			0.5	0.57%	2.19%	0.93%	2.53%	0.75%	2.53%
100	4	2	0.75	0.73%	2.84%	2.16%	5.27%	1.45%	5.27%
			0.9	0.30%	2.72%	2.29%	5.49%	1.29%	5.49%
			0.5	1.46%	4.32%	2.08%	5.66%	1.77%	5.66%
100	4	4	0.75	1.78%	6.35%	3.71%	9.11%	2.75%	9.11%
			0.9	0.65%	8.35%	3.43%	9.34%	2.04%	9.34%

stances tested, the first sequential approach beat the heuristic in just four instances. For the second sequential approach, this happened 48 times.

We can also see that in both tables, the gap increases in direct proportion to the maximum allowed batch size b and the number of customers m. This is expected since the effect due to batching becomes more prominent when the number of customers or the maximum allowed batch size is increased. It may also be noted that the value of α plays an important role. At low values of α, the integrated approach is not significantly better even compared to the first sequential approach since we are laying emphasis on the distribution cost and the first sequential approach minimizes this. But as the value of α increases, we see a significant increase in the gap. When everything else is kept constant, increasing the number of orders n leads to a decrease in the gap. This is explained by the fact that the heuristic H1 is essentially a local perturbation around the SEDD sequence. When the number of orders is increased, the change in tardiness value through this local perturbation does not increase proportionately. Hence in general, when the number of orders is very high compared to the maximum allowed batch size or the number of customers, the performance of heuristic H1 is not significantly better than that of the sequential approach. The integrated approach leads to good improvements in performance under cases where the contribution due to the maximum tardiness is significant in the objective function value.

5. CONCLUSIONS

In this paper, we have studied the production–distribution system with one supplier and one or more customers. Our goal was to optimize a combined objective function that considered both the maximum tardiness and total distribution cost. It was seen that for an arbitrary number of customers, the problem is NP-hard even in the special case where the processing times and the due dates are agreeable. A fast heuristic has been proposed that is asymptotically optimal when the number of orders goes to infinity. Computational tests show that the heuristic is capable of generating near optimal solutions. We have also demonstrated that there is distinct advantage of using the integrated production–distribution approach as compared to the two sequential approaches that try to optimize production and delivery sequentially with no or only partial integration.

It should be noted that, though we have assumed b, the maximum allowed batch size, to be the same for every customer, it is not difficult to extend the heuristic and all the other algorithms to the case where the maximum allowed batch size is dependent on the customers. All the results presented in the paper still hold. We have shown that in the case when the processing times and due dates are agreeable, there exists a procedure that is polynomial in the number of orders that can solve the problem optimally. The complexity of the case with general processing times and due dates and a fixed number of customers is left as an open problem.

In the case when there is no batch size limit, i.e., $b = n$, the problems P2A and P2 with an arbitrary number of customers are still NP-hard because the same NP-hardness proof given in Section 2.1 still works for this case. On the other hand, when $b = n$, both problems P1 and P2 with a fixed number of customers can be solved in polynomial time by the $O(n^{2m+1})$ dynamic programming algorithm of Hall and Potts [10] mentioned in Section 2.1 after it is slightly modified to take transportation times t_i into account. The $O(n^3(\log n)(\log(P + t)))$ algorithm given in Section 2.2 still works for problem P1 with $n = b$. However, the algorithm of Hall and Potts has a lower time complexity.

In this paper, we have not considered shipments that can serve more than one customer. Such a problem would include routing decisions for each shipment. When sharing of shipments across different customers is allowed, the shortest route may not always be the best since we have to take into account the tardiness for the orders delivered at each customer. Consequently, we cannot define the shipping due dates any more as the orders of a customer may be routed through some other customers before getting delivered. New algorithms would be needed to solve such a problem. We believe that the value of production–distribution integration would be even greater in this case because it would require a closer production–distribution linkage in order to fully take advantage of order consolidation across different customers.

ACKNOWLEDGMENT

This research was supported in part by the National Science Foundation under Grants DMI-0196536 and DMI-0421637. The authors are grateful to the helpful comments and constructive suggestions of the two anonymous referees and the associate editor.

REFERENCES

[1] P. Chandra and M.L. Fisher, Coordination of production and distribution planning, European J Oper Res 72 (1994), 503–517.

[2] Z.-L. Chen, Scheduling and common due date assignment with earliness-tardiness penalties and batch delivery costs, European J Oper Res 93 (1996), 49–60.

[3] Z.-L. Chen, "Integrated production and distribution operations: Taxonomy, models, and review," Handbook of quantitative supply chain analysis: Modeling in the e-business era, D. Simchi-Levi, S.D. Wu, and Z.-J. Shen (Editors), Kluwer Academic, New York, 2004, pp. 711–745.

[4] T.C.E. Cheng, V.S. Gordon, and M.Y. Kovalyov, Single machine scheduling with batch deliveries, European J Oper Res 94 (1996), 277–283.

[5] M.A. Cohen and H.L. Lee, Strategic analysis of integrated production-distribution systems: Models and methods, Oper Res 36 (1988), 212–228.

[6] F. Fumero and C. Vercellis, Synchronized development of production, inventory, and distribution schedules, Transportation Sci 33 (1999), 330–340.

[7] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, Freeman, San Francisco, 1979.

[8] J.B. Ghosh and J.N.D. Gupta, Batch scheduling to minimize maximum lateness, Oper Res Lett 21 (1997), 77–80.

[9] J. Hahm and C.A. Yano, Economic lot and delivery scheduling problem: Powers of two policies, Transportation Sci 29 (1995), 222–241.

[10] N.G. Hall and C.N. Potts, Supply chain scheduling: Batching and delivery, Oper Res 51 (2003), 566–584.

[11] L.A. Hall and D.B. Shmoys, Jackson's rule for single-machine scheduling: Making a good heuristic better, Math Oper Res 17 (1992), 22–35.

[12] N.G. Hall, M. Lesaoana, and C.N. Potts, Scheduling with fixed delivery dates, Oper Res 49 (2001), 134–144.

[13] A.M.A. Hariri and C.N. Potts, Single machine scheduling with batch set-up times to minimize maximum lateness, Ann Oper Res 70 (1997), 75–92.

[14] J. Herrmann and C.-Y. Lee, On scheduling to minimize earliness-tardiness and batch delivery costs with a common due date, European J Oper Res 70 (1993), 272–288.

[15] J.R. Jackson, Scheduling a production line to minimize maximum tardiness, Research Report 43, Management Science Research Project, University of California, Los Angeles, 1955.
[16] C.-Y. Lee and Z.-L. Chen, Machine scheduling with transportation considerations, J Scheduling 4 (2001), 3–24.
[17] C.-Y. Lee and C.-L. Li, On the fixed interval due-date scheduling problem, Discrete Appl Math 68 (1996), 101–117.
[18] M. Pinedo, Scheduling: Theory, algorithms, and systems, 2nd edition, Prentice Hall, Upper Saddle River, NJ, 2002.
[19] C.N. Potts, Analysis of a heuristic for one machine sequencing with release dates and delivery times, Oper Res 28 (1980), 1436–1441.
[20] C.N. Potts and M.Y. Kovalyov, Scheduling with batching: A review, European J Oper Res 120 (2000), 228–249.
[21] A.M. Sarmiento and R. Nagi, A review of integrated analysis of production-distribution systems, IIE Trans 31 (1999), 1061–1074.
[22] G.J. Woeginger, Heuristics for parallel machine scheduling with delivery times, Acta Inform 31 (1994), 503–512.
[23] G.J. Woeginger, A polynomial-time approximation scheme for single-machine sequencing with delivery times and sequence-independent batch set-up times, J Scheduling 1 (1998), 79–87.