# Freight Consolidation on a Single Lane

Yue Geng, Diego Klabjan
Department of Industrial Engineering and Management Sciences
Northwestern University
yue-geng@northwestern.edu, d-klabjan@northwestern.edu

October 21, 2014

**Abstract**

We consider freight consolidation problems on a single lane where cost savings can be achieved by switching from less-than-truck to full-truck loads. We first formulate the problem as an integer program and show that it is NP-hard. For the constant truck capacity case, we derive the convex hull of all solutions. In addition, if the per-truck per-day price is constant, we present a simple optimal policy and show the worst case ratio of 2 for the send-when-full policy, which is commonly used in practice. A real-world case study is carried out, which shows that the send-when-full policy can be efficient in practical situations. A two-mode transportation case, where an accelerated transportation mode is available by paying a higher cost, is also discussed. We show that several special cases of the two-mode consolidation problem are polynomially solvable.

# 1 Introduction

In long term shipping contracts, full truck loads are used. They typically cover stable repetitive shipments. Companies with a lot of different items and no patterns in demand, e.g., spare part providers, tend to use less-than-truckload (LTL). However, on a high demand lane they could potentially aggregate demand through a local consolidation point and then use milk runs for delivering from the consolidation point to the end customers, which would lead to full-truck loads. Our study is about whether using such truckload operations is cost effective. Attention also needs to be paid to service level requirements. Consolidation implies that shipment of select items have to be delayed and thus this must not happen too often. In such a setting, there is no correlation among lanes and thus it can be considered on a lane-by-lane basis. We study this problem as part of a collaboration with a large US firm that faced this exact problem.

As a motivating example, consider a large hardware distributor which delivers products from distribution centers all over the country. Currently they use less-than-truckload operations and occasionally parcel. By observing a high volume of orders on one of its major lanes, the company considers using dedicated truckload to serve the lane. Under the requirement that the existing service level should be maintained (i.e., the dedicated truckload should be able to deliver the orders no later than the current LTL operations), the company seeks transportation consolidation strategies to save the cost.

In this paper we study the consolidation problem where there is a single link composed of a single supply location and destination. The orders are dynamic, i.e., there is daily demand over a time horizon, typically a year. Thus the switch from LTL operations to TL would be a strategic decision. In such a strategic decision model, the orders are assumed to be deterministic. Each day, orders are coming in, and each order is associated with weight, and a delivery due date. Since each truck has a fixed cost and a capacity that cannot be exceeded, the problem is to find the most economic way to aggregate orders. We impose strict requirements on the service level, that is, no order can violate its delivery deadline.

The decision is on which days in the time horizon to use a truck and what orders should each truck carry.

In order to obtain the optimal consolidation policy, we use integer programming (IP) techniques. Usually the solutions provided by an IP are not intuitive and are hard to follow by decision makers. Therefore, we also study more practical heuristics. These include send-when-deadline policy and send-when-full policy. We show the optimality of the former policy and give the worse case bound of the latter policy. We also use real world data from our partner to analyze the performance of the different policies.

Lastly, a commonly used transportation service, expedited service, is also studied. In such a two-mode transportation case, an additional fast transportation mode is provided as an option, with a team of gangs operating the truck, thus leading to shorter transportation time but with a higher cost. This additional option opens more possibilities for consolidation. We show that two cases can be solved in polynomial time.

The contributions of this study are as follows. We study the problem of freight consolidation where service level requirements are taken into account explicitly. Although economies of scale in transportation have already been extensively studied, the literature focuses on the trade-off between inventory and transportation costs. To the best of our knowledge, this is the first work to study the minimization of the transportation cost under a service level requirement in a transportation consolidation setting. Second, for the equal truck capacity single-mode consolidation case, we provide the convex hull of the solutions and study the approximation ratio of two commonly used policies. The practical performance of the two polices is also investigated by means of a case study. Third, for the two-mode transportation consolidation case, we provide two special cases that are solved in polynomial time.

The rest of the paper is organized as follows. We review the literature in Section 2 and provide mathematical formulations in Section 3. Section 4 discusses the equal capacity case. We show the convex hull of the solutions, the worse case ratio of the send-when-full policy and optimality of the send-when-deadline policy. In Section 5, we provide a case study to show the practical performance of the two proposed policies. The two-mode transportation consolidation problem is introduced in Section 6, where we focus on two polynomially solvable cases.

## 2 Literature Review

To have a general understanding of consolidation in logistics and supply chain, we refer the readers to Hall (1987). The paper examines different aspects of consolidation: in inventory, in vehicles and in terminals. Possible trade-offs between a lower transportation and higher operations cost, such as the inventory cost, are also discussed. Consolidation papers at network levels provide ILP formulations and solution methodologies, while we focus on analytical studies for the single lane case. For this reason we summarize single lane prior work.

Speranza and Ukovich (1994) consider the problem of finding shipping frequencies, in order to minimize the sum of the transportation and inventory costs for several products on a single lane. Constant demand and supply rates are assumed, which is different from our dynamic order setting. They develop ILP models and show properties of optimal solutions. In the same setting as Speranza and Ukovich (1994), Bertazzi et al. (2007) present worst-case ratios for the zero inventory ordering policies and the frequency-based periodic shipping policies, and Bertazzi and Speranza (2005) study the full load policy, in which a vehicle is shipped from the origin to the destination as soon as the vehicle capacity is saturated.

The problem we study is different from the above problems since dynamic orders are involved. Our problem considers in-vehicle consolidations where there is no inventory holding cost or in-terminal consolidation. With respect to the problem structure, the trade-off appears between the required service level and the transportation cost, instead of between the inventory holding cost and transportation cost, which most literature studies.

From a mathematical modeling perspective, the two most closely related well-studied problems are the capacitated lot-sizing problem and the capacitated set cover problem. In a capacitated lot-sizing problem (see Karimi et al. (2003) for a review), the decisions are when to produce and how much to produce. This is comparable to our decisions, which are when to dispatch a truck and how many orders to put on each truck, however, there are notable differences. For lot-sizing problems the trade-off is between the production and inventory holding cost, and the variables denoting whether to produce or not are binary variables. In our problem, we have different trade-offs, and the variables denoting the number of trucks to dispatch on each day can be any nonnegative integer.

Our problem can also be viewed as a special type of capacitated set cover problem, where orders are elements to cover and trucks serve as covers. We notice that a special capacitated set cover problem, called the rectangle stabbing problem (Even et al. (2008)), is the most relevant to our study. In the one dimensional rectangle stabbing problem with hard capacities, the input consists of a set $\mathcal{U}$ of horizontal intervals (orders in our context) and a set $\mathcal{S}$ of vertical lines (trucks) with capacity $c(S)$ and weight $w(S)$, for each $S \in \mathcal{S}$. The objective is to find the minimum weighted cover from $\mathcal{S}$ to stab or cover each element in $\mathcal{U}$. Even et al. (2008) develop a dynamic programming algorithm for this problem, which has a time complexity of $O(|\mathcal{U}|^2|\mathcal{S}|^2(|\mathcal{U}|+|\mathcal{S}|))$. They also consider one dimensional rectangle stabbing with soft capacities, where each $S \in \mathcal{S}$ can be used unlimited times. This problem can be transformed to the hard-capacity case by duplicating each element in $\mathcal{S}$ for $\lceil \frac{|U(S)|}{c(S)} \rceil$ times, where $U(S)$ denotes the set of segments in $\mathcal{U}$ that intersects with $S$. Note that after using such a transformation, the algorithm is no longer polynomial. We point out that this problem is essentially the same as our single-mode consolidation problem with unequal truck capacities, which we show is NP-hard. To observe the connection of our problem to the rectangle stabbing problem, we imagine the orders with the same delivery time window as a horizontal segment (elements to cover), with the length equal to the length of the time window. Trucks dispatched on a specific day correspond to a vertical line stabbing the segments, given that this specific day falls into the delivery time window of the orders.

We point out that researchers also use stochastic models to study consolidation problems. For example, Bookbinder and Higginson (2002) study a consolidation cycle and assume Poisson process for order arriving and a gamma distribution for weight per order. The cost includes transportation cost and inventory holding cost. Higginson and Bookbinder (1995) use Markov decision processes to minimize the cost per hundred weight per unit time. They seek the optimal action to take for each system state. Çetinkaya and Bookbinder (2003) study a situation where full truckloads are used to ship orders. In such a situation consolidation is important in order to achieve economies of scale in transportation. This setting is similar to ours. They apply the renewal process to find out optimal parameters to use under the time and quantity policy.

## 3 Problem Setting and Formulations

The freight consolidation problem can be described as follows. We have a number of orders to deliver. Each order has a weight, an order placement date at the origin and an order delivery deadline at the destination. An order must be sent at the origin after its placement date and must be received at the destination before its receiving deadline. A truck can only carry a certain number of orders. Note that one order can be split among several trucks. There is a cost associated with dispatching a truck. Our goal is to minimize the total cost of using trucks.

Below we define the notation that is used in the paper.

- **T**: the set of days, ranging from 1 to $T$
- $l_k$: transportation lead time in days for orders sent out on day $k$
- $\bar{d}_{ij}$: weight of orders placed on day $i$, with the delivery deadline at the destination on day $j$ ($j \geq i + l_i$)
- $d_{ij}$: weight of orders placed on day $i$, with sending-out deadline on day $j$
- $D_{lu}$: weight of orders that have to be sent out during days $l$ and $u$

- $C_k$: truck capacity on day $k$, i.e. the weight of orders one truck can accommodate
- $e_k$: cost per truck on day $k$
- $I$: an input instance, which specifies the weight, the placement date (within $[1, T]$) and the receiving deadline (within $[1, T]$) for each order

We assume that orders do not cross in time, i.e., $k+l_k > j+l_j$ if $k > j$. Later we remove this assumption. Note that with the non-crossing order condition, there is a one-to-one correspondence between $\bar{d}_{ij}$ and $d_{ij}$. If an order is placed on day $i$ and has delivery due date on day $j$, then it has to be shipped out on any day $k \geq i$ with $k + l_k \leq j$. We call any such day a sending out day, while the latest such day is called the sending-out deadline. Even if the fleet is homogeneous, lead time $l$ depends on day $k$ due to non-working weekends. The decision variables are introduced as follows.

- $x_{ikj}$: weight of orders sent out on day $k$, with order placement day $i$ and sending-out deadline on day $j$
- $y_k$: number of trucks sent out on day $k$

## 3.1 Models

To address the problem mathematically, we propose the following model, which assumes a single origin and a single destination. We assume that all the subscripts are within the time interval $[1, T]$; otherwise that quantity is not present. We first introduce the straightforward three-index model. The model reads as follows:

$$\min \sum_{k=1}^{T} e_k y_k \tag{1}$$

$$\sum_{k=i}^{j} x_{ikj} = d_{ij} \qquad 1 \leq i \leq j \leq T \tag{2}$$

$$C_k y_k \geq \sum_{i=1}^{k} \sum_{j=k}^{T} x_{ikj} \qquad k = 1, ..., T \tag{3}$$

$$\mathbf{x}, \mathbf{y} \qquad \text{nonnegative integers.} \tag{4}$$

The objective function is to minimize the total transportation cost. Constraint (2) states that all of the demand must be met on time. Constraint (3) requires that truck capacity cannot be exceeded.

To derive a more effective formulation with only the $\mathbf{y}$ variables, we introduce a covering-type model. We begin with a theorem covering a more general consolidation case, where order $i$ with weight $d_i$ has to be sent out on a given set of days $\mathbf{S}_i$ (not necessarily consecutive as is the case in our original statement).

**Theorem 1.** *Let $n_\mathbf{S}$ denote the total weight of orders that have to be sent out over set of days $\mathbf{S}$. The following set of constraints are sufficient to describe the feasible region of the consolidation problem:*

$$\sum_{k \in \mathbf{S}} C_k y_k \geq n_\mathbf{S} \qquad \mathbf{S} \subseteq \mathbf{T}. \tag{5}$$

Note that in Theorem 1, we do not require the set of days $\mathbf{S}$ to contain consecutive days, thus it applies to a more general consolidation setting and there is an exponential number ($2^T$) of constraints included in (5). We call this set of constraints the *covering constraints* and the models with this set of constraints the *covering-type models*.

*Proof.* We construct a network flow graph with source node $s$ and sink node $t$. In addition to source node $s$ and sink node $t$, there are three sets of nodes in the graph. Node set $M$

contains nodes $m_i$ ($1 \leq i \leq n$), where each node $m_i$ corresponds to demand $d_i$. Node set $N$ contains nodes $n_k$ ($1 \leq k \leq T$), where each node $n_k$ corresponds to a possible order-dispatching day, ranging from 1 to $T$. Node set $P$ contains nodes $p_k$, where each node $p_k$ corresponds to a possible order dispatching day, ranging from 1 to $T$. There are arcs from $s$ to each $m_i$ and the capacity of one such arc is set to $d_i$. There is an arc from $m_i$ to $n_k$ if and only if order $d_i$ can be sent out on day $k$, and the capacity of the arc is unlimited. There are arcs from $n_k$ to $p_k$ for each $k$ and the capacity of each such arc is set to $Cy_k$. There are arcs from each node in set $P$ to $t$ and the capacities of these arcs are unlimited.

As an example, Figure 1 shows such a network flow graph with 5 general orders and $T = 3$, where order 1 can be sent out on day 1, order 2 can be sent out on either day 1 or day 2, order 3 can be sent out on either day 1 or day 3, etc. Note that $\mathbf{S}_1 = \{1\}, \mathbf{S}_2 = \{1, 2\}, \mathbf{S}_3 = \{1, 3\}$.

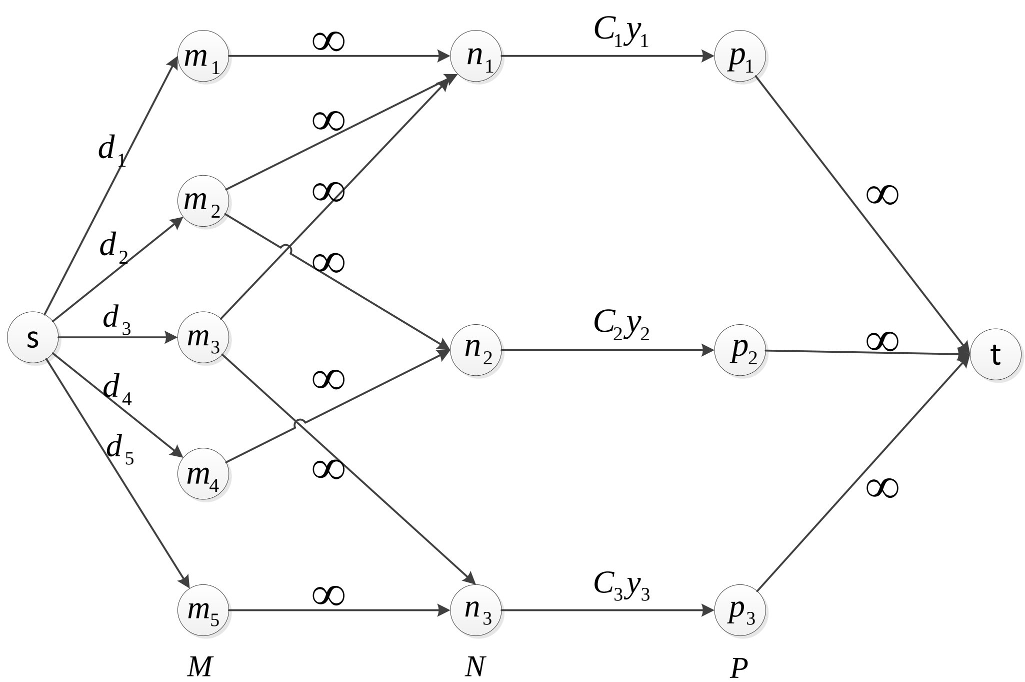


Figure 1: Illustration of the constructed graph with 5 general orders and $T = 3$.

By noting that any feasible solution $(\mathbf{x}, \mathbf{y})$ for the general consolidation problem corresponds to a maximum flow, we can apply the max-flow min-cut theorem on this graph to prove Theorem 1. □

The technique used to obtain this theorem has been applied in the lot-sizing problem with production time windows (see Wolsey (2006), for proof see Cezik and Gunluk (2004)). By applying Theorem 1 to our specific consolidation problem, we obtain the following compact formulation:

$$\min \sum_{k=1}^{T} e_k y_k \tag{6}$$

$$\sum_{k=l}^{u} C_k y_k \geq D_{lu} \qquad 1 \leq l \leq u \leq T \tag{7}$$

$$\mathbf{y} \qquad \text{nonnegative integer} \tag{8}$$

where $D_{lu}$ denotes the weight of orders that need to be sent out between days $l$ and $u$, i.e., $D_{lu} = \sum_{l \leq i \leq j \leq u} d_{ij}$. This reformulation technique has also been applied to lot-sizing problems (see Pochet and Wolsey (2006) for examples). The correctness of this formulation is shown in Appendix A. Therefore, we have shown that the three-index model and the covering-type model are equivalent, which gives the following corollary.

**Corollary 1.** *The projection of the feasible region of the model (1) - (4) into the* $\mathbf{y}$*-space is the same as the feasible region of the model (6) - (8).*

Even with the compact formulation, our consolidation problem is hard to solve. Actually the problem is NP-hard. Considering the case where the only demand is $d_{1T}$, our consolidation problem reduces to the minimum cost unbounded knapsack problem.

# 4 Equal Capacity Case

In practice a carrier may provide trucks of different capacities, however, a contracted shipper usually sticks to the same size truck for its daily operations. In this section, we show that if each truck has the same capacity $C$, the consolidation problem can be solved to optimality in polynomial time. We also discuss two consolidation policies.

The next theorem states that the convex hull of integer solutions can be obtained for the equal capacity case.

**Theorem 2.** *The convex hull of integer* $\mathbf{y}$ *solutions is given by linear constraints*

$$\sum_{k=l}^{u} y_k \geq \lceil \frac{D_{lu}}{C} \rceil \qquad 1 \leq l \leq u \leq T. \tag{9}$$

*Proof.* It is straightforward to see that constraints (9) contain the same integer points as constraints (7). Next we argue that constraints (9) provide the convex set that contains these integer points. Notice that the coefficient matrix of $\mathbf{y}$ in (9) has the structure of an interval matrix, i.e., in each row the 1's appear consecutively, thus it is totally unimodular (TU) (Nemhauser and Wolsey (1988)). It follows that constraints (9) provide an integral polyhedron. This proves the theorem. □

Note that for this case, the constraint matrix is also a network matrix, where a network simplex algorithm can be applied to obtain optimal solutions (Nemhauser and Wolsey (1988)).

## 4.1 Constant cost

Under the assumption that the cost for dispatching a truck is constant over time, we discuss the properties of two commonly used policies.

The first policy is commonly used in practice due to its simplicity and ease of implementation. The rule is that we dispatch a truck as soon as it is filled up to its capacity or there is an impeding due date.

As an example, assume we have 4 days and the truck capacity is 10. A truck needs one day to deliver. There are two orders, in which the first order is placed on day 1, with receiving deadline on day 4 and weight of 15, and the second order is placed on day 2, with receiving deadline on day 3 and weight of 5. Following this policy, we dispatch one truck on day 1 to take 10 units from the first order, and dispatch the second truck on day 2 to take the rest of the first order plus the second order.

More formally, we call a policy *send-when-full* if a truck is dispatched either because we have accumulated enough orders to fill the truck, or because some orders must be sent out due to the deadline constraints. A formal description for this policy is presented in Algorithm 1 in Appendix B.

The second policy is to wait until the deadline approaches. Using the same example as above, under this policy, we do not act on day 1. The first truck is dispatched on day 2 to take the second order and a weight of 5 from the first order. The second truck is dispatched on day 3 with the remains of the first order.

More formally, we call a policy *send-when-deadline* if a truck is dispatched only because some orders have to be sent out due to the deadline constraints. When a truck has to be dispatched, the capacity is taken into account and orders with earlier deadlines are placed first on trucks. A formal description for this policy is presented in Algorithm 2 in Appendix B.

Note that when implementing both policies, we place as many orders as possible in the trucks. The orders with earlier deadlines are placed first. Both algorithms require the orders to be non-crossing since we identify the orders which have to be sent out based on the following rule: on a given day $i$, those orders with deadline on day $j \in J$ have to be sent out, where $J := \{j | i + l_i \leq j \text{ and } (i+1) + l_{i+1} > j\}$.

To assist the analysis of the policies, we introduce $tr(I, P, t)$, which is the number of trucks dispatched for a given input instance $I$ and consolidation policy $P$ at the end of day $t$ ($t \in \{1, ..., T\}$). A policy $P$ is a rule to determine a feasible schedule for dispatching these orders. Under the constant truck cost assumption, the number of trucks used at the end of time horizon $tr(I, P, T)$ represents the objective function under policy $P$.

#### 4.1.1 Optimality of the send-when-deadline policy

The send-when-deadline policy turns out to be optimal, as shown by the next theorem.

**Theorem 3.** *The send-when-deadline policy provides the minimum cost.*

Intuitively as we postpone sending out orders, additional order information is gathered, which is helpful for decision making.

Using the $tr(I, P, t)$ notation, Theorem 3 shows that $tr(I, D, T) \leq tr(I, P, T)$, where $I$ is any instance, $D$ denotes the *send-when-deadline* policy and $P$ is any feasible policy. We may wonder if there exists a time in the horizon when the send-when-deadline policy is suboptimal. The following result shows that no such time point exists.

**Theorem 4.** *The send-when-deadline policy is optimal at any point of time, i.e., for any* $t \in \{1, ..., T\}$

$$tr(I, D, t) \leq tr(I, P, t)$$

*where* $I$ *is any given instance and* $P$ *is any feasible policy.*

The proof is presented in Appendix C. The proof can also easily be adjusted to the more general case of $e_1 \geq e_2 ... \geq e_T$.

**Corollary 2.** *Send-when-deadline policy is optimal if* $e_1 \geq e_2 ... \geq e_T$.

#### 4.1.2 Worst case ratio of the send-when-full policy

The send-when-full policy is commonly used in practice. However, in the worst case, it can incur double the minimum cost.

**Theorem 5.** *The send-when-full policy has a worst case ratio of 2.*

*Proof.* Let $F$ denote the send-when-full policy, and let $D$ denote the send-when-deadline (optimal policy). Let $I$ denote any input instance. We are to show

$$tr(I, F, T) < 2tr(I, D, T). \tag{10}$$

For any given instance $I$, we divide the trucks into two types under policy $F$: type I truck is dispatched only because the truck is full, i.e., each order on it can be delayed; type II truck contains at least one order that cannot be delayed. Correspondingly, the orders can also be divided into two types: type I orders are sent out on type I trucks under policy $F$, and type II orders are sent out on type II trucks under policy $F$. Note that an order may be partitioned into two or more orders if it is split into two or more trucks, under which case the part of the order on type I truck is called type I order, and the part of the order on type II truck belongs to type II order.

We generate instance $I'$ from $I$ by dropping the deadline constraints for all type I orders, and change their order placement to the first day, i.e., type I orders can be sent out on any day without violating their deadlines. Then we have

$$tr(I', D, T) \leq tr(I, D, T).$$

This is because any schedule generated for instance $I$ is feasible for instance $I'$, and $D$ is the optimal policy for instance $I'$. Thus to show (10), it is sufficient to show

$$tr(I, F, T) < 2 \cdot tr(I', D, T) \tag{11}$$

On one hand, for instance $I$, let the total number of type I trucks be $y_f$, and let the total number of type II trucks be $y_d$, then

$$tr(I, F, T) = y_f + y_d.$$

On the other hand, for instance $I'$, following policy $D$, we still need $y_d$ trucks for the orders with deadline constraints. For those $Cy_f$ orders without deadline constraints, there are two ways to send them out: load $Cy_f$ weight of orders on those $y_d$ trucks if there are remaining capacities, or put them on new trucks other than those $y_d$ trucks. Following optimal policy $D$ (Algorithm 2), we must put as many orders as possible on those $y_d$ trucks unless all $y_d$

trucks are full. Let $r$ denote the total capacity remaining on those $y_d$ trucks. Three cases can happen depending on whether the remaining capacity can accommodate those orders with weight $Cy_f$ .

Case 1: $Cy_f = r$. We have $tr(I', D, T) = y_d$, then

$$\frac{tr(I, F, T)}{tr(I', D, T)} = \frac{y_f + y_d}{y_d} = 1 + \frac{r}{Cy_d} < 1 + \frac{Cy_d}{Cy_d} = 2.$$

Case 2: $Cy_f > r$. Let $Cy_f = r + a$, where $a > 0$. We have

$$tr(I', D, T) = y_d + \lceil \frac{a}{C} \rceil.$$

Then we have

$$tr(I, F, T) = y_f + y_d = \frac{r + a}{C} + y_d < \frac{Cy_d + a}{C} + y_d < 2(y_d + \lceil \frac{a}{C} \rceil) = 2tr(I', D, T).$$

Case 3: $Cy_f < r$. We have $tr(I', D, T) = y_d$, then

$$\frac{tr(I, F, T)}{tr(I', D, T)} = \frac{y_f + y_d}{y_d} < 1 + \frac{r}{Cy_d} < 1 + \frac{Cy_d}{Cy_d} = 2.$$

Hence, we always have $tr(I, F, T) < 2tr(I', D, T)$. Therefore, (10) always holds. □

We next show that the ratio of 2 is best possible by means of an example. Let $a$ be a multiple of $C - 1$ and $T = 1 + \frac{aC}{C-1}$. Let the sequence of order placements from day 1 to $T$ be $aC, 1, 1, ..., 1$, where an order of weight $aC$ is placed on day 1 and can be sent out on any day without violating its deadline, and all of the other orders are of weight 1 and must be sent out on the same day as its placement day. The number of orders of weight 1 is $\frac{aC}{C-1}$, which is an integer by choice of $C$. By the send-when-full policy, the number of trucks needed is $a + \frac{aC}{C-1}$. Under the optimal policy, we do not send out a truck on day 1, and send out one full truck on each of the following days. This requires $\frac{aC}{C-1}$ trucks. Thus the ratio of the number of trucks used by the send-when-full policy to that of the send-when-deadline policy equals

$$\frac{a + \frac{aC}{C-1}}{\frac{aC}{C-1}} = 2 - \frac{1}{C}.$$

As $C$ approaches infinity, this ratio approaches 2.

### 4.1.3 The case when orders can be crossed

In all the above, we considered the order non-crossing case ($k + l_k > j + l_j$ if $k > j$). Here we show that we can handle the order crossing case by a simple transformation.

To develop the analysis, notice that certain days are dominated by other days. Specifically, for any day $i$, if there exists a day $j$ such that $j > i$ and $j + l_j < i + l_i$, then day $i$ is dominated by day $j$. Obviously we can postpone orders and trucks sent out on day $i$ until day $j$, without violating the receiving deadlines and without increasing the objective. Therefore, we can transform the order crossing case to the order non-crossing case by ignoring all the days that are dominated. This procedure is described in Algorithm 3 of Appendix D.

## 5 Case Study

To compare the practical performance of the send-when-full and send-when-deadline policies, we carry out a case study based on the data obtained from a large industrial distributor. This partner was very interested in the send-when-full policy since it is easy to understand and implement.

The distributor has its distribution network across North America. We focus our study on one of its major lanes. This major lane goes across several states starting from a major distribution center and ships orders to a region on the west coast. Facing a high volume of orders, the distributor is interested in investigating the potential benefit of shifting from LTL operations, which have been used for years, to dedicated TL operations. The distributor wants to maintain the current service level, i.e., after adopting TL operations, the orders should not be delivered later than the current LTL delivery time. Under their contracts, TL operates on weekends while LTL does not, which allows certain orders to stay at the distribution location for more days before being sent out. Under this service level requirement, the cost of using TL under certain scenarios is computed in order to compare it with the current LTL cost. The possible scenarios include different truck capacities and different lead times. The former refers to the weight of orders a truck can possibly accommodate, while the latter refers to different operational modes. Under standard operations, usually one or two drivers are involved in a single team. However, the distributor has leverage to reach an agreement with certain carriers to take advantage of a fast transportation mode, where teams of drivers are involved. This fast mode can save several days on the road.

We carry out what-if analyses under these scenarios. For each scenario, e.g., different truck capacity under the standard mode or fast mode, both the send-when-full and send-when-deadline policies were implemented and evaluated. The data has been obtained from the distributor for the most recent year. We compute the service level, cost, and truck utilization under each scenario. The service level is defined as the number of days that the actual delivery day is ahead of the delivery deadline. Figures 2 and 3 show these key metrics for the standard mode operation. Figures 4 and 5 give these metrics for the fast mode operation. In all bar charts, the left data point is obtained by assuming the minimum truck capacity, the middle data point with 1.2 times the minimum capacity and the right data point with 1.4 times the minimum capacity. Figures 3 and 5 show the performance of the send-when-deadline policy.

It is not surprising to observe that as truck capacity increases, both cost and truck utilization decrease. Also, as the lead time decreases, more consolidation opportunities are present, which explains the higher utilization rate under the fast mode. An important observation is that no matter whether we use the send-when-full policy or the send-when-deadline policy, the cost and utilization remain the same. This is true under both the standard and fast mode operations. This observation indicates that although in theory the commonly used send-when-full policy might double the cost, in practice it usually achieves the minimum cost. The reason is that in practical situations, orders are usually not allowed to wait too long before being delivered, thus, the send-when-deadline policy cannot explore more order aggregation opportunities than the send-when-full policy. Based on Figures 2 and 4, the send-when-full policy also provides a better service level than the send-when-deadline policy. Therefore, in our industrial distributor case, where orders are dense and the time windows allowed to deliver orders are less than a week, the send-when-full policy has an advantage.

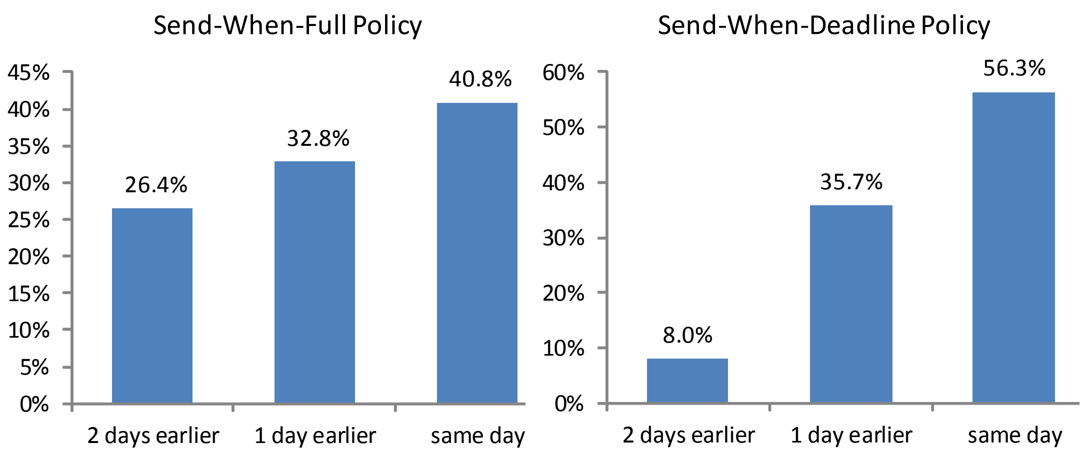


Figure 2: Comparison of Service Levels for the Two Policies under Standard Operation

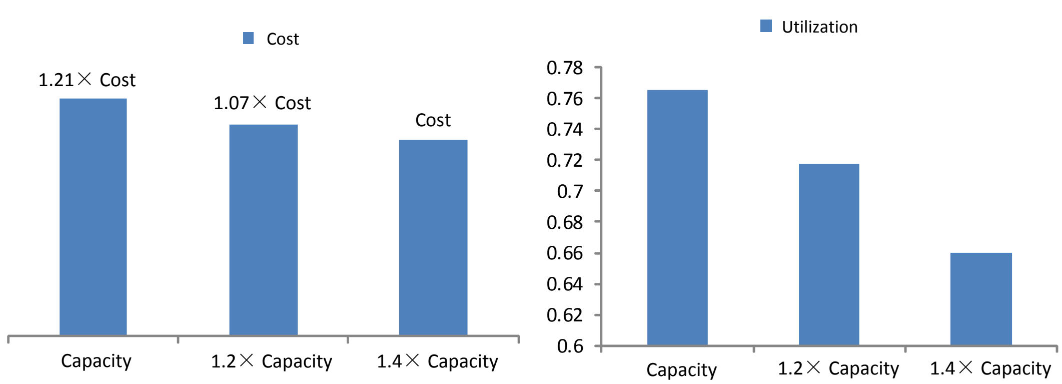


Figure 3: Changes in Cost and Utilization as truck Capacity Increases under Standard Operation

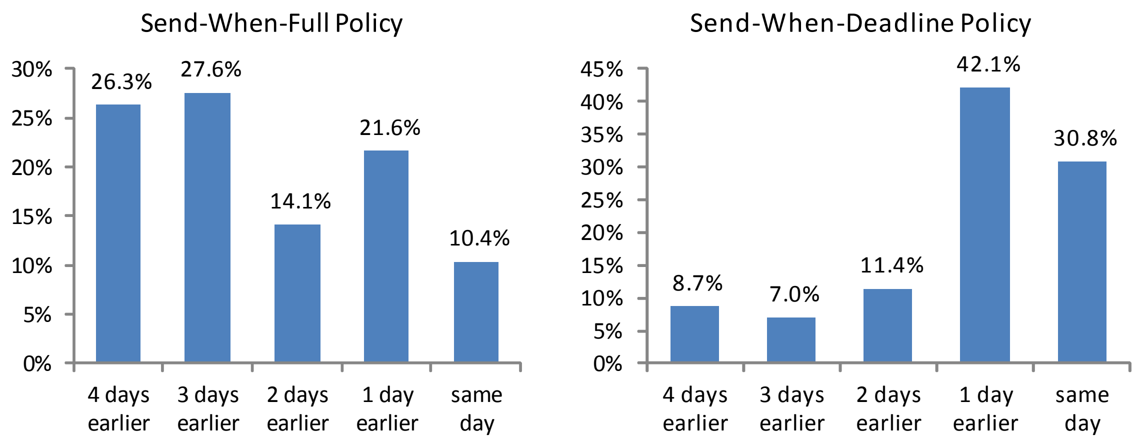


Figure 4: Comparison of Service Levels for the Two Policies under Fast Operation

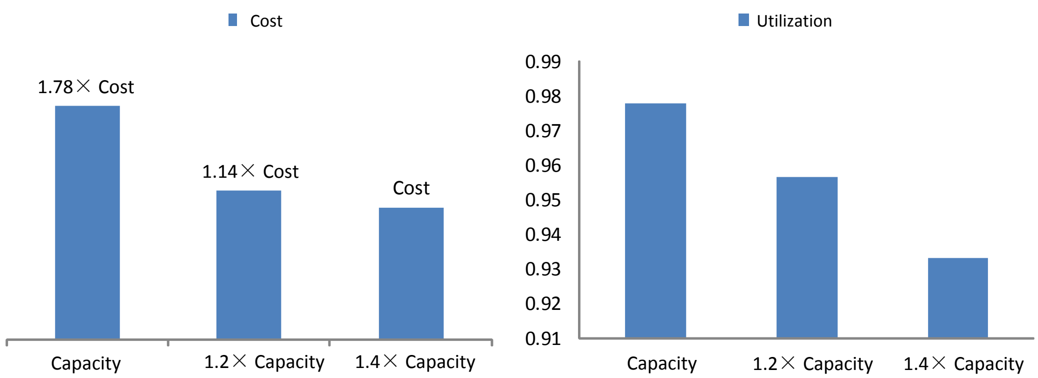


Figure 5: Changes in Cost and Utilization as truck Capacity Increases under Fast Operation

# 6 Consolidation with Two Modes

In this section, an extension of the consolidation problem, where an accelerated mode of transportation is available, is discussed. Consider the case where a team of two or more drivers can operate on different weekly shifts. This operation leads to shorter lead times but at a higher price. In logistics, this type of expedited operations are commonly used. In the following, we refer to the team operation as the fast mode and the normal single-driver operation as the standard mode. Throughout this section, we assume that all the trucks have the same capacity $C$. All the parameters are adapted from the single mode case by adding superscript "F," which denotes the fast mode, or superscript "S," which denotes the standard mode. Similarly, decision variables $x_{ikj}$ and $y_k$ from the single-mode case are adapted by adding superscripts “F” or “S.”

## 6.1 Models

Similar to the single-mode case, we present the three-index model and covering-type models in this section.

We first present the three-index model for the problem, which is similar to the one given in Section 3. Let $t_j^F = max\{k|k + l_k^F \leq j\}$, $t_j^S = max\{k|k + l_k^S \leq j\}$. The model reads as follows.

$$\min \quad \sum_{k=1}^{T}(e_k^F y_k^F + e_k^S y_k^S) \tag{12}$$

$$\sum_{k=i}^{t_j^F} \bar{x}_{ikj}^F + \sum_{k=i}^{t_j^S} \bar{x}_{ikj}^S = \bar{d}_{ij} \qquad i = 1, ..., T \qquad j = i + l_i^F, ..., T \tag{13}$$

$$Cy_k^F \geq \sum_{i=1}^{k} \sum_{j=k+l_k^F}^{T} \bar{x}_{ikj}^F \qquad k = 1, ..., T \tag{14}$$

$$Cy_k^S \geq \sum_{i=1}^{k} \sum_{j=k+l_k^S}^{T} \bar{x}_{ikj}^S \qquad k = 1, ..., T \tag{15}$$

$$\bar{\mathbf{x}}, \mathbf{y} \text{ non-negative integer vectors} \tag{16}$$

In the above formulation, constraint (13) requires that all the demands are satisfied. Constraints (14) and (15) guarantee that the truck capacities are not exceeded.

As we have seen in the covering-type model for the single mode case, we can develop the covering-type models by partitioning the weights of orders $\bar{d}_{ij}$ into portions of orders sent by the fast mode trucks, denoted by $\bar{d}_{ij}^F$, and portions of orders sent by the standard mode trucks, denoted by $\bar{d}_{ij}^S$. Note that in this case both $\bar{d}_{ij}^F$ and $\bar{d}_{ij}^S$ are decision variables. Applying Corollary 1, we have

$$\min \quad \sum_{k=1}^{T}(e_k^F y_k^F + e_k^S y_k^S) \tag{17}$$

$$\sum_{k=l}^{u} C y_k^F \geq D_{lu}^F \qquad 1 \leq l \leq u \leq T \tag{18}$$

$$\sum_{k=l}^{u} C y_k^S \geq D_{lu}^S \qquad 1 \leq l \leq u \leq T \tag{19}$$

$$\bar{d}_{ij} = \bar{d}_{ij}^F + \bar{d}_{ij}^S \qquad i = 1, ..., T \quad j = i + l_i^F, ..., T \tag{20}$$

$$\bar{d}_{ij}^S = 0 \qquad i = 1, ..., T \quad j < i + l_i^S \tag{21}$$

$$D_{lu}^F = \sum_{\substack{i \geq l \\ j \geq u + l_u^F \\ j < (u+1) + l_{u+1}^F}} \bar{d}_{ij}^F \qquad 1 \leq l \leq u \leq T \tag{22}$$

$$D_{lu}^S = \sum_{\substack{i \geq l \\ j \geq u + l_u^S \\ j < (u+1) + l_{u+1}^S}} \bar{d}_{ij}^S \qquad 1 \leq l \leq u \leq T \tag{23}$$

$$\mathbf{D}^F, \mathbf{D}^S, \bar{\mathbf{d}}^F, \bar{\mathbf{d}}^S \geq 0 \qquad \mathbf{y} \text{ non-negative integer vector} \tag{24}$$

In the above formulation, aggregated demand $D_{lu}^F$ and $D_{lu}^S$ denote the weight of orders that have to be sent out between day $l$ and day $u$ using fast mode and standard mode, respectively, provided that the type of transportation mode to use for each individual order is determined. Constraints (18) and (19) state that the truck capacities cannot be exceeded. Constraints (20) and (21) partition the orders into orders sent by fast mode trucks and orders sent by standard mode trucks. By noting that constraints (18) - (19) are equivalent to the following set of constraints,

$$\sum_{k=l}^{u} y_k^F \geq \lceil \frac{D_{lu}^F}{C} \rceil \qquad 1 \leq l \leq u \leq T,$$

$$\sum_{k=l}^{u} y_k^S \geq \lceil \frac{D_{lu}^S}{C} \rceil \qquad 1 \leq l \leq u \leq T,$$

we can simplify the above model as follows.

$$\min \quad \sum_{k=1}^{T}(e_k^F y_k^F + e_k^S y_k^S) \tag{25}$$

$$\sum_{k=l}^{u} y_k^F \geq \lceil \frac{\sum_{i\geq l, j\geq u+l_u^F, j<(u+1)+l_{u+1}^F} \bar{d}_{ij}^F}{C} \rceil \qquad 1 \leq l \leq u \leq T \tag{26}$$

$$\sum_{k=l}^{u} y_k^S \geq \lceil \frac{\sum_{i\geq l, j\geq u+l_u^S, j<(u+1)+l_{u+1}^S} \bar{d}_{ij}^S}{C} \rceil \qquad 1 \leq l \leq u \leq T \tag{27}$$

$$\bar{d}_{ij} = \bar{d}_{ij}^F + \bar{d}_{ij}^S \qquad i = 1, ..., T \quad j = i + l_i^F, ..., T \tag{28}$$

$$\bar{d}_{ij}^S = 0 \qquad i = 1, ..., T \quad j < i + l_i^S \tag{29}$$

$$\bar{\mathbf{d}}^F, \bar{\mathbf{d}}^S \geq 0 \qquad \mathbf{y} \text{ integer vector.} \tag{30}$$

Next we show mathematically that this simplified nonlinear formulation is equivalent to the three-index model.

**Lemma 1.** *The* $\mathbf{y}$*-space projections of the feasible regions of the two models (12) - (15) and (25) - (29) are the same.*

*Proof.* Any feasible partition of $\bar{d}_{ij}$ into $\bar{d}_{ij}^F$ and $\bar{d}_{ij}^S$ corresponds to a feasible partition of $\bar{d}_{ij}$ into $\sum_{k=i}^{t_j^F} \bar{x}_{ikj}^F$ and $\sum_{k=i}^{t_j^S} \bar{x}_{ikj}^S$ and vice versa. For each partition, by applying Corollary 1 to the fast and standard mode separately, we know that the projected feasible region into the $\mathbf{y}^F$- and $\mathbf{y}^S$-space are the same for the two formulations. Therefore, by aggregating all these partitions, the two formulations provide the same projected $\mathbf{y}$-space. □

The issue with the above covering-type model is that $\bar{d}_{ij}^F$ and $\bar{d}_{ij}^S$ are unknown variables, and thus the model is nonlinear. To circumvent this, we develop a model that only involves $\mathbf{y}^F$ and $\mathbf{y}^S$ as decision variables by applying Theorem 1. The model reads as follows:

$$\min \sum_{k=1}^{T} (e_k^F y_k^F + e_k^S y_k^S) \tag{31}$$

$$\sum_{k \in U} C y_k^F + \sum_{k \in V} C y_k^S \geq \sum_{(i,j) \in W} \bar{d}_{ij}$$

$$W = \{(i,j) | [i, j - l_j^F] \subseteq U \text{ and } [i, j - l_j^S] \subseteq V\}, \quad U \subseteq [1,T], V \subseteq [1,T] \tag{32}$$

$$\mathbf{y}^F, \mathbf{y}^S \text{ non-negative integer vectors.} \tag{33}$$

Note that constraint set (32) states that on any set of days, the total capacity provided by the trucks must be no less than the orders that have to be sent out on those days. Unfortunately, this constraint set contains an exponential number ($2^{2T} - 1$) of constraints.

## 6.2 Polynomially solvable cases

In this section, we show that several cases of the two-mode transportation consolidation problem can be formulated by using a totally unimodular (TU) constraint matrix and thus can be solved in polynomial time. We refer the readers to Nemhauser and Wolsey (1988) for details on TU matrices. The complexity of the general two-mode transportation consolidation case is unknown. In the following we assume the fast and standard modes have the same lead time $l^F$ and $l^S$ on different days, respectively. We begin with the case where a consecutive-one (i.e., in each row the 1's appear consecutively, abbreviated as CO1) constraint matrix is sufficient to describe the feasible region. It is known that CO1 matrices are TU.

Note that although in formulation (31) - (33) the constraint matrix is not CO1, we can find dominance relations to transform the constraint matrix into a CO1 matrix under certain conditions. To this end, we make a connection between the feasible solutions of the two-mode consolidation problem and maximum flow on an appropriately constructed network, as demonstrated in Figure 6. The network is constructed in a similar manner as the single-mode case, where vertical layer "M" is composed of order nodes. The difference is that layers "N" and "P" each have $2T$ nodes, where $T$ nodes represent the days to send out a standard-mode truck, and the other $T$ nodes represent the days to send out a fast-mode truck. Note that any feasible solution to the two-mode consolidation problem corresponds to a maximum integer flow on this constructed network flow graph and vice versa (see the proof of Theorem 1). The following lemma states the condition to have a CO1 constraint matrix.

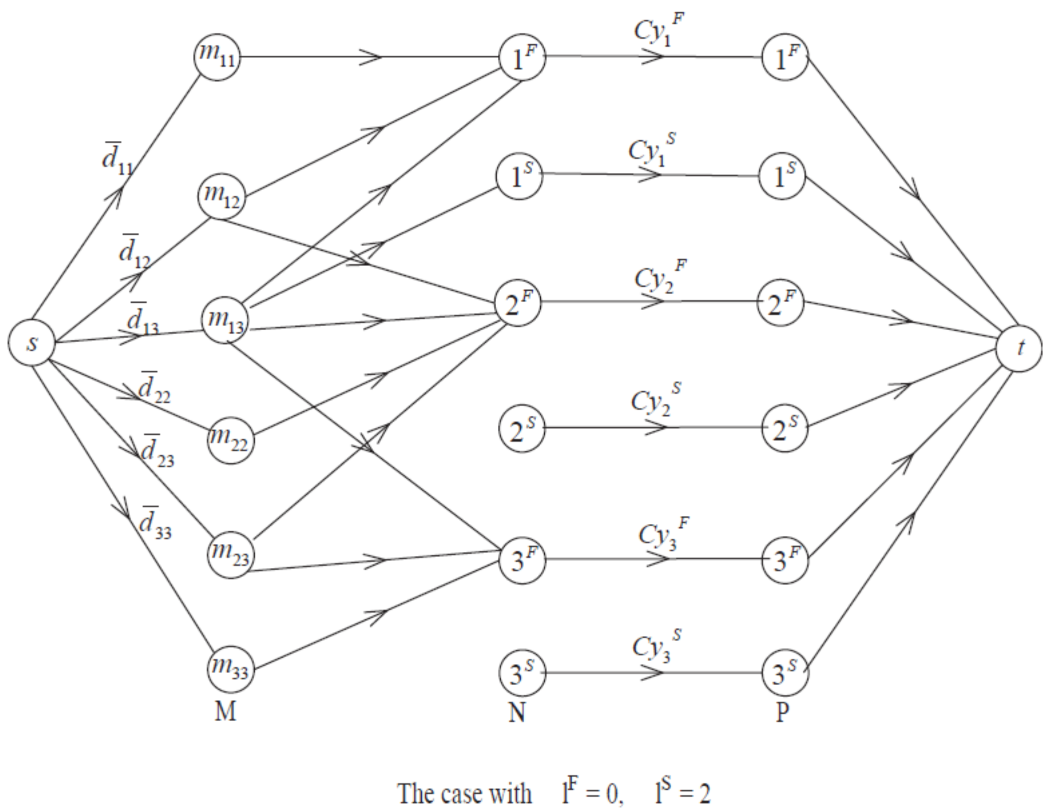


Figure 6: Illustration of the Constructed Graph, with $T = 3$

**Lemma 2.** *A CO1 constraint matrix describes the feasible region of the two-mode transportation problem if and only if there exists an order of the sending-out day nodes (nodes in layer "N") such that each order node (nodes in layer "M") links to consecutive sending-out day nodes.*

As an example, in Figure 6, order nodes do not always connect to consecutive nodes in layer "N." In fact, nodes $m_{12}$, $m_{13}$ and $m_{23}$ all link to non-consecutive sending-out day nodes (for example, node $m_{12}$ links to $1^F$ and $2^F$, which are non-consecutive under the current order). However, if we re-order the nodes in layer "N" as shown in Figure 7, no such cases exist. Therefore, this example can be formulated using a CO1 constraint matrix in the same manner as the single-mode case. We next prove this lemma.

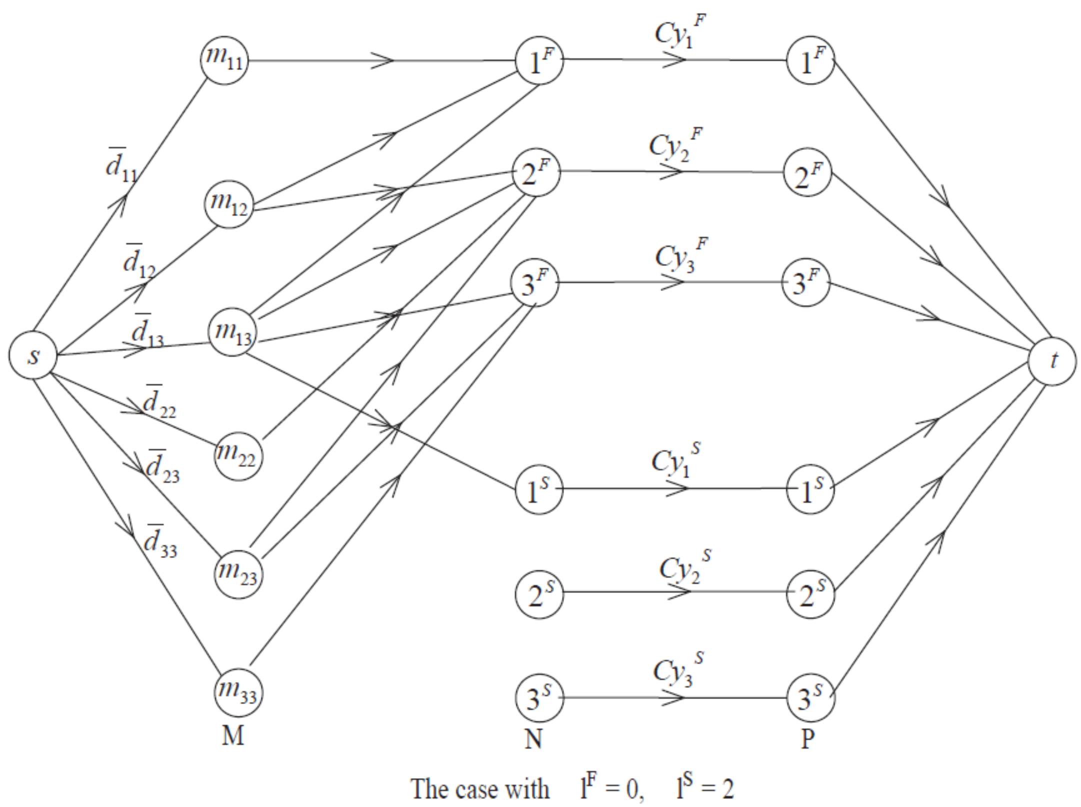


Figure 7: Illustration of the Constructed Graph, with $T = 3$

*Proof.* Let us assume first that the sending-out day nodes are ordered in a way such that each order node links to consecutive sending-out day nodes. Let $n_1$, $n_2$ ,..., $n_{2T}$ denote the send-out day nodes in this order. We can then use the following set of constraints to describe the feasible region:

$$\sum_{k=l}^{u} y_k \geq \lceil \frac{D_{lu}}{C} \rceil \qquad n_1 \leq l \leq u \leq n_{2T},$$

where $D_{lu}$ denotes the weight of orders that have to be sent out between node $l$ and node $u$.

If the linkage from a layer "M" node $\bar{d}_{i'j'}$ to the nodes in layer "N" is split into two sections of nodes $[l_1, u_1]$ and $[l_2, u_2]$, where $l_2 > u_1 + 1$, then the constraint

$$(Cy_{l_1} + ... + Cy_{u_1}) + (Cy_{l_2} + ... + Cy_{u_2}) \geq \sum_{[i,j] \subset [l_1, u_1]} d_{ij} + \sum_{[i,j] \subset [l_2, u_2]} d_{ij} + \bar{d}_{i'j'}$$

cannot be obtained from a consecutive one (CO1) constraint, where $y_1, y_2, ..., y_{2T}$ is any given order of layer "N." □

Note that Lemma 2 is a generalization of the single-mode transportation consolidation case, where we can use a CO1 constraint matrix to describe the feasible region.

The next lemma provides a case where such an order of nodes in Lemma 2 exists so that a CO1 constraint matrix is sufficient to describe the feasible region.

**Lemma 3.** *If $l^S = l^F + 1$, a CO1 matrix describes the feasible region of the two-mode transportation problem.*

*Proof.* We can order the sending-out day nodes in the order of $1^F, 1^S, 2^F, 2^S, ..., T^F, T^S$. Based on Lemma 2, a CO1 constraint matrix is sufficient to describe the feasible region. □

With a CO1 constraint matrix, the problem is polynomially solvable. Other than this CO1 constraint matrix case, there are other situations where the constraint matrix can be transformed into a TU matrix. Next we introduce a simple decomposition of the constraint matrix, and then show some useful properties of the constraint matrix based on this decomposition.

As shown in Figure 7, we can order the nodes in layer $N$ in the order of $y_1^F, ..., y_T^F, y_1^S, ..., y_T^S$. Correspondingly, we can write the constraints in the following form by applying Theorem 1:

$$C \left[ \begin{array}{c|c} \mathbf{A}^F & \mathbf{A}^S \end{array} \right] \begin{bmatrix} \mathbf{y}^F \\ \mathbf{y}^S \end{bmatrix} \geq \mathbf{D}, \tag{34}$$

where $\mathbf{A}^F$ and $\mathbf{A}^S$ are $2^{2T}$ by $T$ matrices, and $\mathbf{D}$ is a $2^{2T}$ dimensional vector representing the weight of orders that have to be sent out on the selected days specified by the matrix $\mathbf{A} = [\mathbf{A}^F|\mathbf{A}^S]$. Note that each element of $\mathbf{A}$ is either 0 or 1. If its value is 1, the corresponding $y$ variable (i.e., day and mode combination) is selected. A key observation is that certain rows of matrix $\mathbf{A}$ are dominated under certain conditions. Next we show that if the truck capacity $C$ is infinite, the dominating rows of matrix $\mathbf{A}$ form a TU matrix. To start, we need the following definitions.

**Definition 1.** Original orders with the same placement day and the same delivery deadline captured by $\bar{d}_{ij}$ are called a *single order.* In model (31) - (33), the constraint stating that the weight of a single order must be less than or equal to the capacity provided on possible sending-out days is called a *single-order constraint.* It corresponds to $W = \{(i,j)\}$ with no other orders required to be sent out in $[i,j]$.

**Definition 2.** At least two single orders combined together are called a *multiple-order.* In model (31) - (33), the constraint stating that the weight of a multiple-order must be less than or equal to the capacity provided on the possible sending-out days is called a *multiple-order constraint.* It corresponds to $W$ that comprises of several single orders.

Example: Assume the standard mode takes 2 days and the fast mode takes 0 days. The set of days is $\{1,2,3,4\}$. We have three single orders: $\bar{d}_{13}$ with weight 1, $\bar{d}_{24}$ with weight 2 and $\bar{d}_{34}$ with weight 4. Let $C$ denote the truck capacity. Then the constraint matrix $\mathbf{A}$ (rows with zero left-hand-side values are omitted) and the right-hand side are:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{pmatrix} 1 \\ 2 \\ 4 \\ 3 \\ 5 \\ 6 \\ 7 \end{pmatrix}$$

where single-order constraints correspond to the first three rows, and the rest of the rows denote multiple-order constraints. Note that every single-order constraint cannot be dominated by other constraints. Multiple-order constraints may be dominated by other constraints. To separate single-order constraints and multiple-order constraints, we write $\mathbf{A}$ in the form of $\left(\begin{smallmatrix} \mathbf{A}_1^F & \mathbf{A}_1^S \\ \mathbf{A}_2^F & \mathbf{A}_2^S \end{smallmatrix}\right)$, where $\mathbf{A}_1 = \begin{pmatrix} \mathbf{A}_1^F & \mathbf{A}_1^S \end{pmatrix}$ and $\mathbf{A}_2 = \begin{pmatrix} \mathbf{A}_2^F & \mathbf{A}_2^S \end{pmatrix}$ represent the single-order and multiple-order constraint matrix, respectively. To prove the TU property of the single-order constraint matrix $\mathbf{A}_1$, we make the following observations.

**Observation 1.** *$\mathbf{A}_1^F$ and $\mathbf{A}_1^S$ are both CO1 matrices.*

**Observation 2.** *In the same row of $\mathbf{A}_1^F$ and $\mathbf{A}_1^S$, the consecutive ones start at the same position (if at least one exists in a row of $\mathbf{A}_1^S$).*

With these observations, we can prove that the single-order constraint matrix is TU.

**Theorem 6.** *For the two-mode transportation case, the single-order constraint matrix $\mathbf{A}_1$ is TU.*

*Proof.* Let us consider odd columns in matrix $\mathbf{A}_1^F$, which we denote by $N$ and the remaining columns by $\bar{N}$. Let us also consider even columns for matrix $\mathbf{A}_1^S$, denoted by $M$, and let $\bar{M}$ be the remaining columns. Let $a_{ij}$ denote the element at row $i$ and column $j$ of matrix $\mathbf{A}_1$.

Let us assume consecutive ones start at an even position in $\mathbf{A}_1^S$. We have $\sum_{j \in N} a_{ij} - \sum_{j \in \bar{N}} a_{ij} \in \{0, 1\}$. Since consecutive ones also start at an even position for $\mathbf{A}_1^F$, we have

$\sum_{j\in M} a_{ij} - \sum_{j\in \bar{M}} a_{ij} \in \{0, -1\}$. Therefore, we have $|(\sum_{j\in N} a_{ij} + \sum_{j\in M} a_{ij}) - (\sum_{j\in \bar{N}} a_{ij} + \sum_{j\in \bar{M}} a_{ij})| \leq 1$ for any row $i$.

Similarly, if consecutive ones start at an odd position in $\mathbf{A}_1^S$, it follows that $|(\sum_{j\in N} a_{ij} + \sum_{j\in M} a_{ij}) - (\sum_{j\in \bar{N}} a_{ij} + \sum_{j\in \bar{M}} a_{ij})| \leq 1$ for any row $i$.

Therefore, we can always pick column set $M \cup N$ such that $|(\sum_{j\in N} a_{ij} + \sum_{j\in M} a_{ij}) - (\sum_{j\in \bar{N}} a_{ij} + \sum_{j\in \bar{M}} a_{ij})| \leq 1$ for any row $i$. This proves that $\mathbf{A}_1$ is a TU matrix (based on Theorem 2.7 of Nemhauser and Wolsey (1988)). □

The next lemma states that if all trucks have a big enough capacity, the constraint matrix $\mathbf{A}$ can be replaced by the single-order constraint matrix $\mathbf{A}_1$. We call the case with $C \geq \sum_i \sum_j d_{ij}$ the uncapacitated case since the fixed cost of sending a truck is still occurred, but there is no restriction on how many orders to include on a single truck.

**Lemma 4.** *For the uncapacitated two-mode transportation case, all of the multiple-order constraints are dominated by single-order constraints.*

*Proof.* Suppose a multiple-order is composed of single orders $\bar{d}_{i_1,j_1}, ..., \bar{d}_{i_n,j_n}$. The corresponding multiple-order constraint holds if one of the variables $y^S_{i_1}, ..., y^S_{j_1-l^S}, y^F_{i_1}, ..., y^F_{j_1-l^F}, ..., y^S_{i_n}, ..., y^S_{j_n-l^S}, y^F_{i_n}, ..., y^F_{j_n-l^F}$ equals 1, which must be true if the single-order constraints for single orders $\bar{d}_{i_1,j_1}, ..., \bar{d}_{i_n,j_n}$ are satisfied. □

We have shown that if the truck capacity can accommodate all the orders at once, the multiple-order constraints are dominated by the single-order constraints. This leads to a strong formulation for the uncapacitated two-mode problem.

**Theorem 7.** *The uncapacitated two-mode transportation case is polynomially solvable.*

*Proof.* Based on Theorem 6 and Lemma 4, the constraint matrix is TU. Thus the conclusion holds. □

Note that for a more general consolidation case, under the condition that a truck has unlimited capacity, we cannot draw the same conclusion. This is stated in the next theorem.

**Theorem 8.** *Given a set of orders $\{d_1, d_2, ..., d_m\}$ of arbitrary integer weight, a set of days $\{1, 2, ..., T\}$, and let order $d_i$ be sent out on set of days $S_i$. On each day the truck capacity is unlimited and the cost for sending out a truck on day $i$ is $e_i$, then the problem of finding the minimum cost schedule is NP-hard.*

*Proof.* This problem is equivalent to the minimum set cover problem, where orders can be perceived as the elements of the ground set and each day can be perceived as a set to cover the elements. □


## Acknowledgment

We thank W.W. Grainger for providing the motivating problem and data for our study.


## A Proof of Correctness of Formulation (6) - (8)

*Proof.* We list all the covering-type constraints as shown in (5). These constraints can be divided into the following two cases.

Case 1: The chosen subset of $\mathbf{T}$ consists of consecutive days. Then the constraints can be written as

$$\sum_{k=l}^{u} C_k y_k \geq D_{lu} \qquad 1 \leq l \leq u \leq T,$$

which is the same as constraint (7).

Case 2: The chosen subset $\mathbf{T}$ consists of non-consecutive days. These non-consecutive days can be divided into $q$ ranges of consecutive days, $[n_{l_1}, n_{u_1}], ..., [n_{l_q}, n_{u_q}]$. We have

$$\sum_{k=l_1}^{u_1} C_k y_k + ... + \sum_{k=l_q}^{u_q} C_k y_k \geq D_{l_1 u_1} + ... + D_{l_q u_q},$$

which is dominated by the inequalities obtained from case 1 and thus is redundant. Therefore, we have shown that constraint set (5) is equivalent to constraint set (7) for the defined consolidation problem. □

# B Algorithms for the Send-When-Full Policy and the Send-When-Deadline Policy

To have a strict description of the policies, we introduce the following notation.

- $r$: at the current time point, the total weight of orders remaining to be sent
- $r_j$: at the current time point, the weight of orders remaining to be sent with a receiving deadline day $j$
- $t^*$: at the current time point, the earliest receiving deadline of the remaining orders

Algorithm 1 provides a detailed description of the send-when-full policy. In Steps 3 and 4, we add the newly placed orders on day $i$ to the remaining orders. Step 5 updates the earliest receiving deadline among the remaining orders. In Step 6, we assess if there are enough orders to fill a full truck. If yes, we dispatch trucks in Step 7. Steps 8 and 9 update the number of remaining orders, while step 10 updates the earliest deadline among the remaining orders. Step 12 checks if there are orders that have to be sent out on day $i$. If yes, we dispatch one more truck to accommodate these orders as shown in Step 13, update the number of remaining orders as shown in Step 14, and update the earliest deadline of the remaining orders to $T$ as shown in Step 15 (actually there are no remaining orders).

Algorithm 2 provides a description of the send-when-deadline policy. Step 3 updates the numbers of remaining orders after orders are placed on day $i$. In Step 4, we dispatch $y_i$ trucks. Note that $\sum_{j=i+l_i}^{i+l_{i+1}} r_j$ denotes the weight of orders that have to be sent out on day $i$ in order to meet the deadline. Step 5 updates the numbers of the remaining orders. Step 6 computes the remaining truck capacities after accommodating for deadline orders. In Step 7, we try to utilize the remaining truck capacities by inserting the remaining orders, with the earliest deadline orders first. In Step 8, we check if the remaining capacity is larger than the weight of the earliest deadline orders. If yes, in Step 9 we put these orders in the truck; if not, we use part of these orders to take up all the remaining truck capacity, as shown in Step 11.

let $r_j = 0$ for $j = 1, ..., T$
**for** $i = 1$ to $T$ **do**
  $r_j = r_j + d_{ij}$ for $j = i, ..., T$
  $r = \sum_{j=i}^{T} r_j$
  $t^* = \min\{j | r_j > 0\}$
  **if** $r \geq C$ **then**
    $y_i = \lfloor \frac{r}{C} \rfloor$
    $r = r - C\lfloor \frac{r}{C} \rfloor$ (orders with the latest deadlines are left)
    $r_l = 0$ for $l = i, ..., k-1$ and $r_k = r - (r_{k+1} + ... + r_T)$
    $t^* = k$
  **end if**
  **if** $(i+1) + l_{i+1} > t^*$ **then**
    $y_i = y_i + 1$
    $r = 0$ and $r_j = 0$ for $j = i, ..., T$
    $t^* = T$
  **end if**
**end for**

**Algorithm 1:** Send-when-full policy

# C Proofs of Optimality of the Send-When-Deadline Policy

To prove Theorem 3, we use $((y_1, X_1), (y_2, X_2), ..., (y_T, X_T))$ to denote a schedule, where $X_k = (z_{k1}, z_{k2}, ..., z_{kT})$ is a vector, and $z_{kj} = \sum_{i=1}^{T} x_{ikj}$ denotes the weight of orders sent out on day $k$ with receiving deadline on day $j$. Obviously, in any feasible schedule, $x_{ikj} = 0$ if the condition $i \leq k$ and $k + l_k \leq j$ does not hold.

*Proof.* Suppose we have an optimal schedule $((y_1^*, X_1^*), (y_2^*, X_2^*), ..., (y_T^*, X_T^*))$ and a schedule $((y_1, X_1), (y_2, X_2), ..., (y_T, X_T))$ generated by Algorithm 2. It is straightforward to check that $((y_1, X_1)$, $(y_2, X_2)$,..., $(y_T, X_T))$ is a feasible schedule by following the algorithm (all deadlines are met and on each day truck capacity is not exceeded). Next we show that the schedule $((y_1, X_1)$, $(y_2, X_2)$,..., $(y_T, X_T))$ provides the same objective value as the optimal schedule.

Suppose day $i$ is the first day that the two schedules differ, that is, $(y_i^*, X_i^*) \neq (y_i, X_i)$ and $(y_e^*, X_e^*) = (y_e, X_e)$ for all $e < i$. We have at the beginning of day $i$ (Step 3 in Algorithm 2), $r_j = r_j^*$ for $j = 1, ..., T$, where $r_j$ and $r_j^*$ denote the weight of orders remaining to be sent

1: let $r_j = 0$ for $j = 1, ..., T$
2: **for** $i = 1$ to $T$ **do**
3: $\quad r_j = r_j + d_{ij}$ for $j = i, ..., T$
4: $\quad y_i = \lceil \frac{\sum_{j=i+l_i}^{i+l_{i+1}} r_j}{C} \rceil$
5: $\quad r_j = 0$ for $j = i + l_i, ..., i + l_{i+1}$
6: $\quad$ define the remaining truck capacity $C_r = \lceil \frac{\sum_{j=i+l_i}^{i+l_{i+1}} r_j}{C} \rceil C - \sum_{j=i+l_i}^{i+l_{i+1}} r_j$
7: $\quad$ **for** $j = (i+1) + l_{i+1}$ to $T$ **do**
8: $\quad\quad$ **if** $C_r > r_j$ **then**
9: $\quad\quad\quad C_r = C_r - r_j$ and $r_j = 0$
10: $\quad\quad$ **else**
11: $\quad\quad\quad C_r = 0$ and $r_j = r_j - C_r$
12: $\quad\quad\quad$ break
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: **end for**

**Algorithm 2:** Send-when-deadline policy

with deadline on day $j$, for the schedule generated by the algorithm and for the optimal schedule, respectively. We distinguish two cases.

Case 1: The number of deadline orders $\sum_{j=i+l_i}^{i+l_{i+1}} r_j = 0$. Then we have $y_i = 0$ and $X_i = \mathbf{0}$. We adjust the optimal schedule as follows: $y_{i+1}^* = y_{i+1}^* + y_i^*$, $X_{i+1}^* = X_{i+1}^* + X_i^*$, $y_i^* = 0, X_i^* = \mathbf{0}$. Then the new schedule $((y_1^*, X_1^*), (y_2^*, X_2^*), ..., (y_T^*, X_T^*))$ is still optimal but we have $(y_i^*, X_i^*) = (y_i, X_i)$.

Case 2: The number of deadline orders $\sum_{j=i+l_i}^{i+l_{i+1}} r_j \neq 0$. Following Algorithm 2, the number of trucks used is $y_i = \lceil \frac{\sum_{j=i+l_i}^{i+l_{i+1}} r_j}{C} \rceil$. Since orders $r_{i+l_i}, ..., r_{i+l_{i+1}}$ have to be sent out on day $i$, $y_i^* \geq y_i$ must hold. There are two sub-cases.

Case 2.1: Every truck used on day $i$ fully utilizes its capacity (i.e., Step 12 is executed). We have $y_i C = \sum_{j=i+l_i}^{i+l_{i+1}} r_j + r'$, where $r'$ denotes the weight of orders that are inserted to fully take up the truck capacity. Note that these orders have the earliest deadlines among the remaining orders.

Case 2.1.1: In the optimal schedule, the orders included in $r'$ are already on the trucks used on day $i$. We can set $y_i^* = y_i$, $X_i^* = X_i$, $y_{i+1}^* = y_{i+1}^* + (y_i^* - y_i)$, $X_{i+1}^* = X_{i+1}^* + (X_i^* - X_i)$. Again we have a new optimal schedule with $(y_i^*, X_i^*) = (y_i, X_i)$.

Case 2.1.2: In the optimal schedule, some or all of the orders (denoted by $\hat{r}$) included in $r'$ are not on the trucks used on day $i$. We move these orders to the trucks used for day $i$ and keep $y^*_{i+1}, ..., y^*_T$ unchanged. If capacity $Cy^*_i$ is not exceeded, we set $y^*_i = y_i$, $X^*_i = X_i$, $y^*_{i+1} = y^*_{i+1} + (y^*_i - y_i)$, $X^*_{i+1} = X^*_{i+1} + (X^*_i - X_i)$, which yields a new optimal schedule with $(y^*_i, X^*_i) = (y_i, X_i)$. If capacity $Cy^*_i$ is exceeded, we move those orders with the latest deadlines out until capacity $Cy^*_i$ is not exceeded, and insert them to the space previously occupied by $\hat{r}$ (note that $y^*_1, ..., y^*_T$ remain unchanged). By letting $y^*_i = y_i$, $X^*_i = X_i$, $y^*_{i+1} = y^*_{i+1} + (y^*_i - y_i)$, $X^*_{i+1} = X^*_{i+1} + (X^*_i - X_i)$, we have a new optimal schedule with $(y^*_i, X^*_i) = (y_i, X_i)$.

Case 2.2: At the end of day $i$, all the orders are sent out (i.e., Step 12 is not executed). We have $X_i \geq X^*_i$ coordinate-wise. Note that in this case $y^*_i = y_i$ ($y^*_i > y_i$ cannot hold since empty trucks cannot appear in an optimal schedule). Setting $X^*_i = X_i$ and $(X^*_{i+1} + ... + X^*_T) = (X^*_{i+1} + ... + X^*_T) - (X_i - X^*_i)$, and keeping $y^*_{i+1}, ..., y^*_T$ unchanged, yields a new optimal schedule with $(y^*_i, X^*_i) = (y_i, X_i)$.

This completes all cases. We conclude that we can always adjust the current optimal schedule to obtain $(y^*_i, X^*_i) = (y_i, X_i)$ without increasing the objective value. Repeating this procedure shows that $((y_1, X_1), (y_2, X_2), ..., (y_T, X_T))$ generated by Algorithm 2 provides an optimal solution. □

Next we prove Theorem 4. Note that $tr(I, P, t)$ denotes the number of trucks dispatched for a given consolidation instance $I$ and consolidation policy $P$ at the end of day $t$ ($t \in \{1, ..., T\}$).

*Proof.* Assume under the send-when-deadline policy $D$, we dispatch trucks on days $a_1, ..., a_n$ ($a_1 < ... < a_n$). It is sufficient to show that $tr(I, D, a_i) \leq tr(I, P, a_i)$ for every $i \in \{1, ..., n\}$.

To show this, for any $i \in \{1, ..., n\}$ we generate instance $I_i$ from $I$ as follows. The new order set $I_i$ is composed of those orders that have to be sent out by day $a_i$. For any policy $P$ we have

$$tr(I_i, P, a_i) \leq tr(I, P, a_i) \qquad i \in \{1, ..., n\}$$

Applying Theorem 3, we have

$$tr(I_i, D, a_i) = tr(I_i, D, T) \leq tr(I_i, P, T) = tr(I_i, P, a_i) \qquad i \in \{1, ..., n\}.$$

Note the fact that

$$tr(I_i, D, a_i) = tr(I, D, a_i) \qquad i \in \{1, ..., n\},$$

therefore, we obtain $tr(I, D, a_i) \leq tr(I, P, a_i) \qquad i \in \{1, ..., n\}$. □

We provide the proof of Corollary 2 as follows.

*Proof.* Note that the total cost is expressed as

$$\sum_{k=1}^{T} e_k y_k = (e_1 - e_2) y_1 + (e_2 - e_3)(y_1 + y_2) + (e_3 - e_4)(y_1 + y_2 + y_3) + ... + (e_{T-1} - e_T) \sum_{i=1}^{T-1} y_i + e_T \sum_{i=1}^{T} y_i.$$

By Theorem 4, $y_1, y_1 + y_2, ..., y_1 + y_2 + ... + y_T$ are all minimized by the *send-when-deadline* policy and thus $\sum_{k=1}^{T} e_k y_k$ is also minimized by the *send-when-deadline* policy. □

# D Handling Order Crossing Case

We use Algorithm 3 to transform the order crossing case into the order non-crossing case. The generated order non-crossing case has a set of new orders $d_{ij}$ with specific sending out days in set $\Gamma$, where $\Gamma$ is the set of non-dominated days. In Steps 3 and 4, we assess if the current day is dominated by a later day $j$. If yes, we delete the current day from set $\Gamma$, add the orders placed on the current day to the next day, and delete these orders from the current day, as shown in Steps 5, 6 and 7.

```
Let the set Γ = {1, ..., T}
for i = 1 to T do
    for j = i + 1 to T do
        if j + l_j < i + l_i then
            Γ = Γ\{i}
            let d_{i+1,j} = d_{i+1,j} + d_{i,j} for any j
            let d_{i,j} = 0 for any j
            break
        end if
    end for
end for
return Γ and all orders d_{ij}
```

**Algorithm 3:** Transform crossing order case to non-crossing order case

# References


Bertazzi, Luca, Lap Mui Ann Chan, Maria G. Speranza. 2007. Analysis of practical policies for a single link distribution system. *Naval Research Logistics* **54** 497–509.

Bertazzi, Luca, Maria G. Speranza. 2005. Worst-case analysis of the full load policy in the single link shipping problem. *International Journal of Production Economics* **93-94** 217–224.

Bookbinder, James H., James K. Higginson. 2002. Probablistic modeling of freight consolidation by private carriage. *Transportation Research Part E* **38** 305–318.

Çetinkaya, Sila, James H. Bookbinder. 2003. Stochastic models for the dispatch of consolidated shipments. *Transportation Research Part B: Methodological* **37** 747–768.

Cezik, Tolga, Oktay Gunluk. 2004. Reformulating linear programs with transportation constraints - with applications to workforce scheduling. *Naval Research Logistics* **51** 275 – 296.

Even, Guy, Retsef Levi, Dror Rawitz, Baruch Schieber, Shimon Shahar, Maxim Sviridenko. 2008. Algorithms for capacitated rectangle stabbing and lot sizing with joint set-up costs. *ACM Transactions on Algorithms* **4** Article No. 34.

Hall, Randolph W. 1987. Consolidation strategy: Inventory, vehicles and terminals. *Journal of Business Logistics* **8** 57–73.

Higginson, James K., James H. Bookbinder. 1995. Markovian decision processes in shipment consolidation. *Transportation Science* **29** 242–255.

Karimi, Behrooz, S.M.T. Fatemi Ghomi, John Wilson. 2003. The capacitated lot sizing problem: a review of models and algorithms. *Omega* **31** 365 – 378.

Nemhauser, George L., Laurence A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley.

Pochet, Yves, Laurence A. Wolsey. 2006. *Production Planning by Mixed Integer Programming*. Springer.

Speranza, Maria G., Walter Ukovich. 1994. Minimizing transportation and inventory costs for several products on a single link. *Operations Research* **42** 879–894.

Wolsey, Laurence A. 2006. Lot-sizing with production and delivery time windows. *Mathematical Programming* **A 107** 471 – 489.