Final Report

to the

Center for Multimodal Solutions for Congestion Mitigation (CMS)

CMS Project Number: 2010-018

CMS Project Title: Impacts of Efficient Transportation Capacity Utilization via Multi-Product Consolidation on Transportation Network Usage and Congestion

for period 9/16/10 to 13/31/11

Joseph Geunes and Dincer Konur
Industrial & Systems Engineering
303 Weil Hall, PO Box 116596
Gainesville, FL 32611
352-392-1464, Ext. 2012
geunes@ise.ufl.edu

Date prepared: 8/6/12

TABLE OF CONTENTS

Acknowledgment of Sponsorship
This work was sponsored by a grant from the Center for Multimodal Solutions for Congestion Mitigation, a U.S. DOT Tier-1 grant-funded University Transportation Center.

LIST OF TABLES

Abstract

We study multi-item inventory problems that explicitly account for realistic transportation cost structures and constraints, including a per-truck capacity and per-truck cost. We analyze shipment consolidation and coordination policies under these conditions. A set partitioning problem is formulated to determine the best consolidation policy. We first use a branch-and-price method to solve the resulting set partitioning problem. Since the pricing problem for the general case is NP-hard, two heuristic methods are proposed to generate new columns. We also show that the pricing problem can be solved in polynomial time for a practical special case. Furthermore, we develop two heuristic methods as alternatives to the branch-and-price method. Numerical studies are conducted to demonstrate the efficiency of the heuristic column generators, heuristic methods to the set partitioning problem, and how the modeling approach helps mitigate truck density on transportation networks while resulting in higher truck utilization and lower total costs.

Executive Summary

When multiple items share fixed costs and/or resource capacity, independent inventory control for the items typically leads to solutions that are far from optimal. For example, when the shared resource corresponds to transportation capacity, independent inventory decisions may maximize the number of shipments required for delivery (as a result of shipping items separately). The supply chain management literature has, therefore, intensely focused on integrated inventory and transportation problems and many policies have been proposed to deal with such problems.

This research project examines a multi-item inventory problem that explicitly accounts for practical transportation cost structures and constraints, including individual truck capacities and shipment costs. That is, we consider truckload transportation costs. In particular, truckload transportation cost modeling implies that the additional fixed cost paid for each shipment is a step function of the shipment quantity, whereas, less-than-truckload transportation cost modeling assumes per unit transportation costs. In this study, we focus on a multi-item inventory control problem. In particular, we consider shipment consolidation opportunity as different items can share the same trucks for their shipment requirements.

The problem we study has similarities with the classical joint replenishment problem, although there are also significant differences, as we do not aim to find a basic cycle length which specifies shipment frequencies. Instead, our aim is to find a partition of a set of items such that each subset in the partition corresponds to a subset of items which are consolidated and shipped together. Each subset of consolidated items may therefore have different replenishment cycle lengths. We model this problem as a set partitioning problem and propose a column generation based solution method as well as two heuristic solution approaches. To the best of our knowledge, this model has not been studied in the supply chain literature with the explicit consideration of truckload cost structures. This work therefore contributes to literature by modeling this problem, studying and revealing its underlying structural properties, and providing efficient solution methods.

The consolidation policy analyzed in this study leads to decreased number of trucks required to ship the same amount of commodities, i.e., reduced truck density and increased truck capacity utilization. Considering the need for low CO2 emissions in transportation, this study ideally is able to propose policies for green transportation in supply chains. Furthermore, these policies lead to less truck congestion on the distribution network.

1 Background

When multiple items share fixed costs and/or resource capacity, independent inventory control for the items typically leads to solutions that are far from optimal. For example, when the shared resource corresponds to transportation capacity, independent inventory decisions may maximize the number of shipments required for delivery (as a result of shipping items separately). The supply chain management literature has, therefore, intensely focused on integrated inventory and transportation problems, and numerous policies have been proposed to deal with such problems (see, e.g., Silver et al., 1998).

This paper examines a multi-item inventory problem that explicitly accounts for practical transportation cost structures and constraints, including individual truck capacities and shipment costs. That is, we consider truckload transportation costs, where an additional setup cost is incurred for each truck. This results in an order cost component that is a step function of the shipment quantity, whereas less-than-truckload (LTL) transportation cost modeling often assumes only per unit transportation costs. As Toptal and Çetinkaya (2006) show for a single-item case, explicit consideration of truck transportation cost structures often results in using fewer total shipments for transportation. We note that similar cost structures have been considered in the literature for different types of single-item inventory control problems to model truckload transportation costs (see, e.g., Aucamp, 1982, Lee, 1986, Hwang et al., 1990, Çetinkaya and Lee, 2002, Lee et al., 2003, Toptal et al., 2003, Zhao et al., 2004, Toptal and Çetinkaya, 2006, Mendoza and Ventura, 2008, Toptal, 2009, Zhang et al., 2009). In this study, we focus on a multi-item inventory control problem. In particular, we consider shipment consolidation opportunities across items, as different items can share trucks for their shipment requirements.

The majority of past studies on shipment consolidation considers stochastic demand environments. In particular, these studies seek to determine a *time*-, *quantity*-, or *time-and-quantity*-based consolidation policy for customers with stochastic arrivals of shipping requirements. In a time-based policy, the question of interest is when to ship customer demands that accumulate over time, whereas a quantity-based policy specifies how much to accumulate before dispatching a truck (for examples of modeling studies focusing on time- and quantity-based consolidation policies, see, e.g., Higgison, 1995, Higgison and Bookbinder, 1995, Çetinkaya and Lee, 2000, Çetinkaya and Bookbinder, 2003). A time-and-quantity policy defines the time to release a truck, unless a specified quantity is accumulated before that time (see, e.g., Bookbinder and Higgison, 2002, Ching and Tai, 2005, Çetinkaya et al., 2006, Mutlu et al., 2010). Higgison and Bookbinder (1994) compare these three policy approaches using simulation, and Chen et al. (2005) compare time- and quantity-based policies under Vendor-Managed-Inventory. We refer the reader to Çetinkaya (2005) for a detailed discussion of integrated inventory control and transportation studies with multi-item consolidation.

In contrast, we analyze freight consolidation with explicit transportation costs in a multi-item Economic-Order-Quantity (EOQ) model, i.e., we consider deterministic demand. The *joint replenishment problem* considers multi-item inventory systems with deterministic demand, where each shipment involves a (major) setup cost and each item has an individual (minor) setup cost if it is included in a shipment. Goyal (1974, 1975), Goyal (1974, 1975), Silver (1975, 1976), Jackson et al. (1985), Viswanathan (1996, 2002), Wildeman et al. (1997), and Moon et al. (2010) study joint replenishment problems and their solution methods. Due to the problem's complexity, heuristic methods are commonly used for solving the resulting models. One may refer to Khouja and Goyal (2008) for a review of various joint replenishment problems studied in the literature. A common solution approach for joint replenishment problems is to determine a base cycle length along with an integer multiple for each item. This integer multiple denotes how frequently the item is replenished (for instance, a multiple of one may indicate that the inventory is replenished every week, while a multiple of three corresponds to replenishment every three weeks). Since the groups of items that share truck capacity are an indirect byproduct of the solution strategy, this approach is referred to as indirect grouping strategy (Khouja and Goyal, 2008). In a direct grouping strategy, on the other hand, the groups of items that will be jointly shipped are directly determined. In this paper, we adopt a direct grouping strategy for the problem of interest. That is, we do not restrict individual items to be replenished at integer multiples of a base cycle length. Instead, our aim is to find a partition of a set of items such that each subset in the partition corresponds to a subset of items which are consolidated and are always shipped together. Each subset of consolidated items may therefore have different replenishment cycle lengths.

Çetinkaya and Lee (2002) determine an inventory replenishment cycle length for the supply to a warehouse of a single item with deterministic demand, as well as an integer number of *consolidation cycles* within a replenishment cycle, where a consolidation cycle represents the time between consecutive shipment releases made to the market from the warehouse. As noted by Çetinkaya and Lee (2002), time- and quantity-based consolidation policies are identical in the case of deterministic demand. Moon et al. (2010) extend the consolidation problem of Çetinkaya and Lee (2002) to a multi-item inventory system. They determine a base cycle length, an integer value that specifies each individual item's inventory replenishment cycle length, and the number of consolidation cycles for each item within each of its replenishment cycles. However, while Çetinkaya and Lee (2002) model explicit transportation costs, which implies the setup cost paid for each replenishment is a nonlinear function of the replenishment quantity, Moon et al. (2010) consider fixed major and minor setup costs, which are constants, as in the joint replenishment problem.

The problem we study has similarities with the classical joint replenishment problem, although there are also significant differences. Unlike traditional joint replenishment problems, our model explicitly considers truck capacities and per-shipment fixed costs (rather than a simple shared fixed cost for each order). Similar to Ben-Khedher and Yano (1994), Levi et al. (2008), and Kang and Kim (2010), the number of trucks used to deliver a consolidated shipment is a decision variable (this generalization is sometimes referred to as the case with soft capacities, see, e.g., Levi et al., 2008). Ben-Khedher and Yano (1994), Levi et al. (2008), and Kang and Kim (2010) focus on a joint replenishment problem in a finite planning horizon such that each planning horizon has finite number of periods and each period constitutes the base cycle length. In our model, since no such base cycle length is defined, our solution methods differ from the solution methods proposed for joint replenishment problems. In particular, we model the problem of interest as a set partitioning problem and propose a column generation based solution method as well as two heuristic solution approaches. Sindhuchao et al. (2005) model a set partitioning problem and develop column generation based solution method for an inventory routing problem with limited truck capacity. While they adopt a direct grouping strategy as we do, they assume that each subset of consolidated items is shipped by a single truck. On the other hand, our model defines the number of trucks used for each dispatch of a consolidated group of items as a decision variable. To the best of our knowledge, this model has not been studied in the supply chain literature with the explicit consideration of truckload cost structures. This work therefore contributes to the literature by modeling this problem, studying and revealing its underlying structural properties, and providing efficient solution methods.

It is well known that set partitioning problems are NP-complete (Garey and Johnson, 1979). Hence, it is not an uncommon practice in the literature to use column generation methods to solve set partitioning problems. Branch-and-price methods use column generation within a branch-and-bound scheme for solving integer (or mixed integer) programs with a large number of columns. One may refer to Barnhart et al. (1998) and Lübbecke and Desrosiers (2005) for examples of the use of column generation methods for the exact solution of integer programs, while Wilhelm (2001) provides a technical review of column generation in integer programming. Branch-and-price has also been employed for solving multi-item or multi-period inventory control problems with integer decision variables (see, e.g., Díaz and Fernández, 2002, Shen et al., 2003, Freling et al., 2003, Lulli and Sen, 2004, Degraeve and Jans, 2007). In applying the branch-and-price method to our model, we observe that the pricing problem, which is used to generate columns, is NP-hard; therefore, we propose two heuristic methods for generating attractive columns for the general case and use one of them (the one which can generate better columns, on average, with similar computational times, based on our computational experience) as a heuristic column generator within the branch-and-price method. Using a heuristic for generating columns tends to reduce the average time required to generate an attractive column. However, when such a heuristic is unable to find an attractive column, exact solution of the underlying set partitioning problem requires solving the pricing problem exactly. As a heuristic method, however, we might choose not to solve the pricing problem exactly. Although this approach can lead to invalid bounds for the original problem in the branch-and-bound tree, it often enables finding quick feasible solutions, as we later show in our numerical study section. Therefore, we consider the application of the branch-and-price method as both an exact method and as a heuristic method, where the latter corresponds to cases in which we do not solve the pricing problems to optimality at each node in the branch-and-bound tree. In addition, for a practical case, we show that the pricing problem can be solved in polynomial time. It is worth noting that this special case generalizes the EOQ model with market choice flexibility defined and analyzed in Geunes et al. (2004) by modeling explicit truckload transportation costs.

This work serves as an important decision tool for the following practical supply chain scenarios.

1. *Single-Retailer, Multi-item Systems*: In this setting, a single retailer who sells multiple items needs to control each item's inventory. Each item obeys the assumptions of the EOQ model, and the retailer is responsible for the transportation cost of any order. The transportation cost is determined by the number of trucks used to ship inbound orders to the retailer. In this case, the retailer may achieve substantial savings in transportation costs by consolidating the orders for different items, which requires common replenishment cycle lengths for consolidated items. On the other hand, a common replenishment cycle length may increase inventory-related costs, including holding and order setup costs, for consolidated items. The problem is to determine which subsets of items should be consolidated, as well as the common cycle length specific to each set of consolidated items.

2. *Single-Distributor, Multi-Retailer Systems*: In this setting, multiple retailers order a product from a common distributor. Assuming Vendor-Managed-Inventory, the distributor controls the inventory at the retailer locations and pays for the transportation cost of any shipment. The distributor can increase truck utilization by consolidating retailer shipments. Similar to the previous scenario, a tradeoff exists between reduced truck costs and increased inventory-related costs. The distributor's problem is to determine which retailers should be consolidated, as well as the common delivery cycle length for each subset of consolidated retailers.

3. *Multi-Supplier, Multi-Retailer Systems*: In this setting, a set of retailers is owned by a single firm, as in a retail chain. Each retailer requires shipments from a set of suppliers. The chain store's

problem is to determine which suppliers' shipments to which retailers should be consolidated, as well as a common delivery cycle for each subset of consolidated product and retailer shipments.

Scenario 3 above corresponds to a problem the authors observed in a distribution channel that motivated this study. In particular, an air-conditioning company in Florida operates as follows. The company has a set of retailers throughout the region, and these retailers place orders for products from multiple suppliers via the company's distribution department. Currently, the distribution department simply passes orders to suppliers who then ship to individual retailers, i.e., orders are decentralized and are not coordinated. However, the company observed high transportation costs due to under-utilized trucks and wished to consider consolidation approaches to increase truck utilization and reduce transportation costs. Our modeling approach will assume that in Scenario 1, the different items come from a common supplier or distributor. In Scenario 2, our modeling approach would assume that the multiple retailers are co-located in an area that is far from the distributor location, so that any local drop-off (routing) costs are extremely small in comparison to the long-haul truckload shipping cost from the distributor's location to the retailers' local area. As a result, we can use a single model to analyze either of the first two scenarios, as each requires managing a number of consolidated shipments that are effectively between two locations (or two regions). The third scenario can be reduced to the first (or second) scenario by decomposing the problem by each origin-destination pair. That is, we consider the problem of consolidating shipments of multiple products from a given supplier to each retail area. Thus, for ease of exposition in defining and formulating our model, it is sufficient for us to use Scenario 1 as a basis for this description.

The rest of this paper is organized as follows. In Section 2, we formulate the multi-item EOQ model with shipment consolidation and explicit truckload transportation costs as a set partitioning problem. Furthermore, for two subproblems (involving a single-item and a given subset of consolidated items), we discuss how to determine the shipment policy. Section 2.1 explains the details of the branch-and-price method applied to the set partitioning problem and discusses two heuristic methods for the model of interest in this paper. A set of numerical studies is conducted in Section 3 to analyze the efficiency of the heuristic column generation techniques proposed for the branch-and-price method, the efficiency of the heuristic approaches to the set partitioning problem, and the costs and benefits of the proposed modeling approach. Concluding remarks, a summary of the contributions, and a set of future research directions are given in Section 4.

2 Research Approach

Consider a set S containing n items, i.e., $|S| = n$, where $|S|$ denotes the cardinality of S. We assume that a single supply chain agent controls the distribution of these items. Each item may be used to represent a collection of different product types replenished from individual suppliers to the agent. We assume that each item obeys the basic Economic Order Quantity (EOQ) model assumptions. That is, (i) the demand rate for each item is assumed to be known and constant, (ii) there is a constant lead time associated with an order of any item, (iii) shortages are not allowed, and (iv) the planning horizon is infinite. Assumption (iv) indicates that the chosen replenishment policy will be applied over the foreseeable future, which can be assumed to be infinity. Under the basic EOQ model assumptions, the relevant costs associated with any given item are defined as follows. A fixed cost is incurred when placing an order for an item. Let a_i denote the fixed order cost associated with item i, $i = 1, 2, \ldots, n$. An inventory holding cost is incurred for each item; let h_i denote the per unit volume per unit time inventory holding cost for item i, $i = 1, 2, \ldots, n$. Furthermore, let c_i denote the per unit volume purchase cost of item i.

Now, suppose that the agent controls each item independently, and s/he wishes to determine the optimal order volume for an item, which also specifies the time between orders for the item. That is, let v_i and t_i denote the order volume and the replenishment cycle length of item i, respectively. Then $v_i = \lambda_i t_i$, where λ_i denotes the demand rate (in volume per unit time) for item i. It is well known that the total cost per unit time associated with item i amounts to

$$f_i(v_i) = \frac{h_i v_i}{2} + \frac{a_i \lambda_i}{v_i}. \tag{1}$$

The first term in Equation (1) is the average holding cost per unit time and and the second term is the average order cost per unit time associated with item i.* It can be easily shown that $f_i(v_i)$ is convex in v_i and, hence, the agent controlling item i will achieve the minimum cost by replenishing $v_i^{eoq} = \sqrt{2a_i\lambda_i/h_i}$ units with any order of item i.

Equation (1) does not explicitly account for the structure of transportation costs encountered in many applications.† Similar to studies by Aucamp (1982), Lee (1986), Hwang et al. (1990), Çetinkaya and Lee (2002), Lee et al. (2003), Toptal et al. (2003), Zhao et al. (2004), Toptal and Çetinkaya (2006), Mendoza and Ventura (2008), Toptal (2009), and Zhang et al. (2009), we next consider a generalization of the basic EOQ model that explicitly considers transportation costs using a truck-load

*The purchase cost for item i is not considered in Equation (1) as purchase cost per unit time, $c_i\lambda_i$, is constant for any item i.

†Except for cases in which item i is the only item, v_i^{eoq} does not exceed truckload capacity, a_i accounts for the fixed truckload transportation cost, and c_i for the variable transportation cost.

(TL) transportation cost approach.

Let R denote the cost of a truck shipment and let P denote the per truck capacity. Then, the total transportation cost associated with an order of item i is equal to $\lceil v_i/P \rceil R$. The total cost per unit time, including explicit transportation costs for item i, reads as

$$g_i(v_i) = \frac{h_i v_i}{2} + \frac{a_i \lambda_i}{v_i} + \left\lceil \frac{v_i}{P} \right\rceil \frac{\lambda_i R}{v_i}. \tag{2}$$

Note that the only difference between Equations (1) and (2) is the transportation cost per unit time, accounted for in the last term of Equation (2). Modeling transportation costs under the TL approach introduces discontinuities and results in a non-convex cost function. Therefore, one cannot directly use first-order optimality conditions to determine the optimal shipping volume for item i. A detailed analysis of $g_i(v_i)$ shows that a procedure can be developed to find the value of v_i, say $v^{(i)}$, that minimizes $g_i(v_i)$. In particular, note that $g_i(v_i)$ has a piecewise continuous structure, where each piece is an EOQ-type of cost function. Let $\tilde{v}_i^{(k)} = \sqrt{2(a_i + kR)\lambda_i/h_i}$, for some nonnegative integer k. Furthermore, let ℓ be the unique integer such that $\ell P < v_i^{eoq} \leq (\ell+1)P$. The following properties of $g_i(v_i)$ are given without proof. One may refer to Lee (1986) and Toptal et al. (2003) for a deeper discussion on minimizing $g_i(v_i)$.

Property 1 *$g_i(v_i)$ satisfies the following properties:*

- *$g_i(v_i)$ is decreasing over $(k-1)P < v_i \leq kP$, $\forall k \leq \ell$.*
- *$g_i(kP) \leq g_i(v_i)$ for $v_i \geq kP$ if $k \geq \ell + 1$.*
- *If $\tilde{v}_i^{(\ell+1)} \geq (\ell+1)P$, then $g_i(v_i)$ is decreasing over $\ell P < v_i^{eoq} \leq (\ell+1)P$. If $\tilde{v}_i^{(\ell+1)} < (\ell+1)P$, then $g_i(v_i)$ is decreasing over $\ell P < v_i^{eoq} \leq \tilde{v}_i^{(\ell+1)}$ and increasing over $\tilde{v}_i^{(\ell+1)} < v_i^{eoq} \leq (\ell+1)P$.*

Based on Property 1, the minimizer of $g_i(v_i)$ is defined as follows

$$v^{(i)} = \arg\min\{g_i(\min\{\tilde{v}_i^{(\ell+1)}, (\ell+1)P\}), g_i(\ell P)\}. \tag{3}$$

Then we have $t^{(i)} = v_i^{(i)}/\lambda_i$.

When the supply chain agent controls each item separately, s/he should minimize $g_i(v_i)$ for each item i, $i = 1, 2, \ldots, n$. However, as noted previously, independent control of different items is suboptimal and tends to maximize the number of trucks required for delivery. The model we define next aims at finding the best partition of the set of items, such that each subset of the partition contains items that will be consolidated on common shipments. In particular, let J denote the set of all possible subsets of S, indexed by j, and let S_j denote a particular subset of S, for each $j \in J$. Suppose that all item orders for items in S_j are consolidated on common shipments. In this case, the agent's decision variable for the items in S_j corresponds to the length of a single replenishment cycle, which will be common for all items in S_j. Let T_j denote the common replenishment cycle length when all of the item orders in S_j are consolidated. If we define V_j as the aggregate replenishment volume of the items in S_j, then $V_j = T_j \sum_{i \in S_j} \lambda_i$. Furthermore, let $\Lambda_j = \sum_{i \in S_j} \lambda_i$, $A_j = \sum_{i \in S_j} a_i$, and $H_j = (\sum_{i \in S_j} h_i \lambda_i)/(\sum_{i \in S_j} \lambda_i)$. Note that Λ_j defines the total demand volume per unit time of the items in S_j, A_j defines the total fixed order cost for consolidating these items, and H_j defines the weighted average holding cost per unit volume per unit time for the consolidated items in S_j. Then the total cost per unit time associated with the items in S_j reads

$$G_j(V_j) = \frac{H_j V_j}{2} + \frac{A_j \Lambda_j}{V_j} + \left\lceil \frac{V_j}{P} \right\rceil \frac{R \Lambda_j}{V_j}. \tag{4}$$

The first term in Equation (4) is the total holding cost per unit time, the second term is the total order cost per unit time, and the last term is the total transportation cost per unit for the items in S_j. We note that $G_j(V_j)$, defined in Equation (4), has the same structure as $g_i(v_i)$, defined in Equation (2). Therefore, one can easily show that Property 1 also holds for $G_j(V_j)$. In particular, let $V_j^{eoq} = \sqrt{2A_j\Lambda_j/H_j}$, and define Υ_j to be the unique integer such that $\Upsilon_j P < V_j^{eoq} \leq (\Upsilon_j + 1)P$. Furthermore, let $\tilde{V}_j^{(k)} = \sqrt{2(A_j + kR)\Lambda_j/H_j}$ for some nonnegative integer k. Then V_j^*, i.e., the minimizer of $G_j(V_j)$ for $V_j \geq 0$, can be written as

$$V_j^* = \arg\min\{G_j(\min\{\tilde{V}_j^{(\Upsilon_j+1)}, (\Upsilon_j + 1)P\}), G_j(\Upsilon_j P)\}. \tag{5}$$

Therefore, we have $T_j^* = V_j^*/\Lambda_j$.

Once a partition of the items is determined, the supply chain agent needs to minimize $G_j(V_j)$ for each subset S_j included in the partition.[‡] However, given n items, there are $2^n - 1$ possible subsets of items, and the agent's problem is to choose a partition, i.e., a set of subsets such that each item is contained in exactly one subset. Furthermore, we seek the partition that will minimize the agent's total cost per unit time. Then $G_j(V_j^*)$ corresponds to the cost per unit time when the supply chain agent chooses subset S_j in his/her partition.

The agent's set partitioning problem can be formulated as follows. Let

$$x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is in subset } S_j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the x_{ij} values are defined by the the subsets S_j, i.e., $x_{ij} = 1 \; \forall i \in S_j$. Furthermore, let

$$y_j = \begin{cases} 1 & \text{if subset } S_j \text{ is in the selected partition,} \\ 0 & \text{otherwise.} \end{cases}$$

[‡]As total purchase cost per unit time will be the same constant for any partition, purchase costs can be ignored in formulation of the set partitioning problem. Note that $G_j(V_j)$ does not include purchase costs per unit time, therefore, the set partitioning problem also does not consider purchase costs.

The y_j variables determine the selection of item subsets, i.e., $y_j = 1$ implies that subset S_j is selected. Then, the agent's set partitioning problem is:

$$
\begin{aligned}
(\mathbf{P}): \quad \min \quad & \sum_{j \in J} G_j(V_j^*) y_j \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} y_j = 1, \;\; i \in S, \\
& y_j \in \{0, 1\}, \;\; j \in J.
\end{aligned}
$$

The goal is to choose a partition that minimizes the total cost associated with item replenishments and shipments. The first set of constraints assures that each item is included in one consolidated shipment, i.e., only one of the selected subsets can contain each item. A summary of the notation used is given in Appendix 5.1. Additional notation will be defined as needed. In Section 2.1, we focus on methods to solve problem **P**.

2.1 Analysis of the Problem

As noted previously, problem **P**, defined in Section 2, is a set partitioning problem. We note that set partitioning problems have been shown to be NP-complete (Garey and Johnson, 1979). More specifically, given n products, problem **P** has $2^n - 1$ decision variables. Hence, it is not an uncommon practice in the literature to use column generation methods to solve set partitioning problems, as the set partitioning formulation often has a small integrality gap. In this section, we next discuss a column generation based method called branch-and-price to solve problem **P**. We then discuss two heuristic methods for solving this problem.

2.1.1 Branch-and-Price Algorithm

Column generation methods are often used when a problem of interest has a number of variables too large to enumerate explicitly. Branch-and-price methods use column generation within a branch-and-bound scheme for solving integer programs with a large number of columns. Barnhart et al. (1998) and Lübbecke and Desrosiers (2005) provide detailed discussions on classes of problems suitable for column generation. In a branch-and-price scheme, the linear relaxation problem (LRP) at a node of the branch-and-bound search tree is optimized using column generation. First, a restricted LRP is considered, where only a subset of the columns is considered. This restricted LRP is also called the restricted master problem (RMP). Then, a pricing problem is used to potentially generate new columns with attractive reduced cost values. If no new columns can be generated, this implies that the solution to the RMP is an optimal solution for the LRP. When this solution is fractional, the branching process is applied and the relaxed problems at new nodes are again solved using the column generation method. Next, we discuss a branch-and-price method for problem **P**.

Recall that problem **P** is a set partitioning problem. The LP relaxation of problem **P** is

$$
\begin{aligned}
(\textbf{LRP}): \quad \min \quad & \sum_{j \in J} g_j y_j \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} y_j = 1, \; i \in S, \\
& 0 \leq y_j \leq 1, \; j \in J,
\end{aligned}
$$

where we use $g_j = G(V_j^*)$ for notational simplicity. Now, suppose that only a subset, J' ($J' \subset J$), of the decision variables, i.e., columns (where each column is a vector representation of a subset of products) is considered. Then the RMP is written as

$$
\begin{aligned}
(\textbf{RMP}): \quad \min \quad & \sum_{j \in J'} g_j y_j \\
\text{s.t.} \quad & \sum_{j \in J'} x_{ij} y_j = 1, \; i \in S, \\
& 0 \leq y_j \leq 1, \; j \in J'.
\end{aligned}
$$

Let π_i be the dual variable associated with the constraint $\sum_{j \in J'} x_{ij} y_j = 1$. (Note that the dual variables associated with the constraints $y_j \leq 1 \; \forall j \in J'$ are ignored as they will be equal to zero in the optimal dual solution. Also, one can equivalently state **RMP** without these constraints.) The solution of **RMP** is optimal for **LRP** if there is no column with a negative reduced cost. To determine whether a column exists with a negative reduced cost, the following pricing problem is used:

$$
(\textbf{PP0}): \quad \min_{j \in \{J \backslash J'\}} \quad g_j - \sum_{i \in S} x_{ij} \pi_i.
$$

Note that $g_j - \sum_{i \in S} x_{ij} \pi_i$ gives the reduced cost for the variable y_j. Therefore, the solution of **PP0** will determine the column with the minimum reduced cost. If the optimal objective value of **PP0** is non-negative, then this implies that the solution of **RMP** is optimal for **LRP** and column generation at the node terminates. Otherwise, a new column is added to **RMP** and a new problem of the form **PP0** is solved using the dual solution of the new **RMP**. This process continues until no column is found with a negative reduced cost. Next, we reformulate the pricing subproblem.

First, we note that the pricing subproblem can be optimized over J instead of $J \backslash J'$ (because the reduced cost of any column in J' is nonnegative). Then the pricing subproblem is used to find the column with the minimum reduced cost over all possible feasible columns. Recall that a column corresponds to a vector representation of a subset of items. That is, the pricing subproblem seeks the subset of items with the minimum reduced cost. Let

$$
x_i = \begin{cases} 1 & \text{if item } i \text{ is in the subset,} \\ 0 & \text{otherwise,} \end{cases}
$$

and let $\mathbf{x}$ denote the n-vector of x_i values. Then the pricing subproblem can be reformulated as follows:

$$(\mathbf{PP1}): \quad \min \quad G(\mathbf{x}) - \sum_{i \in S} x_i \pi_i$$
$$\text{s.t.} \quad x_i \in \{0,1\}, \ i \in S,$$

where $G(\mathbf{x})$ is the total cost per unit time associated with $\mathbf{x}$. Considering Equation (4), **PP1** can be rewritten as

$$(\mathbf{PP2}): \quad \min \quad -\sum_{i\in S}\pi_i x_i + \frac{V\sum_{i\in S} h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \frac{1}{V}\sum_{i\in S} a_i x_i \sum_{i\in S}\lambda_i x_i + \frac{R\Upsilon}{V}\sum_{i\in S}\lambda_i x_i$$
$$\text{s.t.} \quad (\Upsilon - 1)P < V \leq \Upsilon P,$$
$$x_i \in \{0,1\}, \ i \in S,$$
$$\Upsilon \in \{1,2,3,\ldots\}.$$

The decision variable Υ denotes the number of trucks used for each shipment of items in the subset, i.e., the objective function computes the cost of using Υ trucks (this is why the lower bounding constraint for V uses a strict inequality). Note that **PP2** is a Mixed Integer Nonlinear Programming (MINLP) problem and let us define $\mathbf{x}^*$, V^*, and Υ^* to be an optimal solution of **PP2**. We next discuss an important property of **PP2**.

Property 2 *Let* $(\mathbf{x}^0, V^0)$ *be a solution to the following problem:*

$$(\textbf{\textit{R-PP2}}): \quad \min \quad -\sum_{i\in S}\pi_i x_i + \frac{V\sum_{i\in S} h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \frac{1}{V}\sum_{i\in S} a_i x_i \sum_{i\in S}\lambda_i x_i + \frac{R}{P}\sum_{i\in S}\lambda_i x_i$$
$$s.t. \quad V \geq 0,$$
$$x_i \in \{0,1\}, \ i \in S.$$

Define $\Upsilon^0 = \lfloor V^0/P \rfloor$*. Then* $\Upsilon^0 \leq \Upsilon^* \leq \Upsilon^0 + 1$.

Proof: The proof is given in Appendix 5.2.

It follows from Property 2 that if we know the value V_0, then we can solve **PP2** by solving the problem **PP2-k**, defined below, for $k = \Upsilon^0$ and $k = \Upsilon^0 + 1$.

$$(\mathbf{PP2\text{-}k}): \quad \min \quad -\sum_{i\in S}\pi_i x_i + \frac{V\sum_{i\in S} h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \frac{1}{V}\sum_{i\in S} a_i x_i \sum_{i\in S}\lambda_i x_i + \frac{Rk}{V}\sum_{i\in S}\lambda_i x_i$$
$$\text{s.t.} \quad (k-1)P < V \leq kP,$$
$$x_i \in \{0,1\}, \ i \in S.$$

Note that **PP2-k** is a generalization of the Unconstrained Binary Quadratic Optimization Problem, which is known to be NP-hard (Palubeckis, 2004)[§]. This then implies that **PP2** is NP-hard. Therefore, we focus on heuristic methods to solve the pricing subproblem. However, in what follows, we consider

[§]For a given V, when $h_i = h$, $\forall i \in S$, **PP2-k** is the unconstrained binary quadratic optimization problem.

a special case of **PP2**, for which we propose an exact solution method that runs in polynomial time. Prior to the analysis of this special case, we next note another property of **PP2-k**, which we utilize in the solution of the special case. In particular, let $(\mathbf{x}^k, V^k)$ be a solution of **PP2-k**. Note that for any given $\mathbf{x}$, the objective function of **PP2-k** is convex in V and, hence, in the optimal solution of **PP2-k**, we have

$$V^k = \begin{cases} \lim_{V \downarrow (k-1)P} V & \text{if } \tilde{V}^{(k)}(\mathbf{x}^k) \le (k-1)P, \\ \tilde{V}^{(k)}(\mathbf{x}^k) & \text{if } (k-1)P < \tilde{V}^{(k)}(\mathbf{x}^k) \le kP, \\ kP & \text{if } \tilde{V}^{(k)}(\mathbf{x}^k) > kP, \end{cases}$$

where $\tilde{V}^{(k)}(\mathbf{x}^k) = \sum_{i \in S} \lambda_i x_i^k \sqrt{2\left(\sum_{i\in S} a_i x_i^k + kR\right) \Big/ \sum_{i \in S} h_i \lambda_i x_i^k}$.

Property 3 *For $k = \Upsilon^0$ and $k = \Upsilon^0 + 1$, if $\tilde{V}^{(k)}(\mathbf{x}^k) = \lim_{V \downarrow (k-1)P} V$ in any solution of **PP2-k**, then it is not a solution of **PP2**.*

Proof: The proof is given in Appendix 5.3.

Property 3 implies that we do not need to consider the cases when V^k converges to the lower bound of **PP2-k** for $k = \Upsilon^0$ and $k = \Upsilon^0 + 1$ in the solution of **PP2**.

2.1.2 A Special Case: Identical holding costs and a fixed order cost value

We now discuss how to solve the pricing problem when (i) all items in a set S have the same holding cost per unit volume per unit time (or when the average holding cost per item serves as a sufficiently accurate approximation that can be applied to all items) and (ii) a single fixed order cost is charged for any consolidated set instead of for each item. This special case most closely corresponds to the second scenario mentioned in Section 1, i.e., when a distributor controls the ordering decisions for a set of retailers who receive the same product. Under this scenario, while the retailers may have different demand rates, the holding cost is roughly identical for each retailer, as each holds the same product. Furthermore, since the distributor controls ordering decisions, consolidating orders may lead to a single order cost for the distributor. Thus, it is reasonable to assume $h_i = h \ \forall i \in S$ and $\sum_{i \in S} a_i x_i = a$. Under these assumptions, problem **R-PP2**, defined in Property 2, reduces to

$$(\textbf{R-PP2}_{SC}) \quad \min \quad \sum_{i \in S} \widehat{c}_i \lambda_i x_i + \sqrt{2ah \sum_{i \in S} \lambda_i x_i}$$
$$\text{s.t.} \quad x_i \in \{0, 1\}, \ i \in S,$$

where $\widehat{c}_i = R/P - \pi_i/\lambda_i$. **R-PP2**$_{SC}$ can be solved in polynomial time (see Shen et al., 2003, Geunes et al., 2004). In particular, one first sorts items in increasing order of $\widehat{c}_i$ values. Then, if an optimal solution has ℓ items consolidated, these items will be the ones with ℓ smallest $\widehat{c}_i$ values. Therefore, one can determine Υ^0 in polynomial time, and then solve **PP2-k** with $k = \Upsilon^0$ and $k = \Upsilon^0 + 1$.

For the special case of interest, **PP2-k** reduces to

$$(\mathbf{PP2\text{-}k}_{SC}): \quad \begin{array}{ll} \min & -\sum_{i\in S}\pi_i x_i + \frac{hV}{2} + \frac{Rk+a}{V}\sum_{i\in S}\lambda_i x_i \\ \text{s.t.} & (k-1)P < V \le kP, \\ & x_i \in \{0,1\},\ i \in S. \end{array}$$

Note that for any feasible V, one can determine the corresponding $\mathbf{x}$ that minimizes the objective function value of $\mathbf{PP2\text{-}k}_{SC}$ for the given V by assigning $x_i = 1$ when $c_i(V) = ((Rk+a)/V - \pi_i/\lambda_i)\lambda_i < 0$ and $x_i = 0$ otherwise (when $c_i(V) > 0\ \forall i \in S$, we then set $x_j = 1$ for $j = \arg\min\{c_i(V) : i \in S\}$ and $x_i = 0\ \forall i \in S\backslash\{j\}$, as we do not consider the empty set, i.e., $\mathbf{x} = \mathbf{0}$, as a feasible consolidation). Furthermore, we know from Property 3 that V^k is equal to either kP or $\sqrt{2(a+kR)\sum_{i\in S}\lambda_i x_i^k/h}$. We can thus provide a polynomial-time solution method for $\mathbf{PP2\text{-}k}_{SC}$, formally stated as follows.

Property 4 *Algorithm 1, stated below, solves* $\mathbf{PP2\text{-}k}_{SC}$ *for* $k = \Upsilon^0$ *and* $k = \Upsilon^0 + 1$.

Algorithm 1

1. *Let* $V^{(k)} = kP$. *Index items in nondecreasing order of* $c_i(V^{(k)}) = ((Rk+a)/V^{(k)} - \pi_i/\lambda_i)\lambda_i$. *If* $c_1(V^{(k)}) > 0$, *define* $\mathbf{x}^{(k)}$ *by assigning* $x_1^{(k)} = 1$ *and* $x_i^{(k)} = 0\ \forall i \in S\backslash\{1\}$. *Else, define* $\mathbf{x}^{(k)}$ *by assigning* $x_i^{(k)} = 1$ *if* $c_i(V^{(k)}) < 0$, $x_i^{(k)} = 0$ *otherwise. Let* $z^{(k)}$ *equal the objective function value of* $\mathbf{PP2\text{-}k}_{SC}$ *at* $(\mathbf{x}^{(k)}, V^{(k)})$. *Set* $\mathbf{x}^k = \boldsymbol{x}^{(k)}$, $V^k = V^{(k)}$, *and* $z^k = z^{(k)}$. *Go to Step 2.*

2. *Let* $V^{(k)} = \sqrt{2(a+kR)\sum_{i\in S}\lambda_i x_i^{(k)}/h}$. *If* $V^{(k)} \in \big((k-1)P, kP\big]$, *let* $z^{(k)}$ *equal the objective function value of* $\mathbf{PP2\text{-}k}_{SC}$ *at* $(\mathbf{x}^{(k)}, V^{(k)})$ *and if* $z^{(k)} < z^k$, *set* $\mathbf{x}^k = \mathbf{x}^{(k)}$, $V^k = V^{(k)}$, *and* $z^k = z^{(k)}$. *Go to Step 3.*

3. *If* $c_1(V^{(k)}) > 0$, *go to Step 4. Else, let* j *be the largest index such that* $x_j^{(k)} = 1$. *Then redefine* $\mathbf{x}^{(k)}$ *by assigning* $x_j^{(k)} = 0$ *while keeping other components unchanged and go to Step 2.*

4. *Return* $\mathbf{x}^k$ *and* V^k.

Proof: The proof is given in Appendix 5.4.

Note that the sorting in Step 1 of Algorithm 1 is the dominant process in the algorithm; hence, Algorithm 1 can solve $\mathbf{PP2\text{-}k}_{SC}$ in $\mathcal{O}(n\log n)$ time for $k = \Upsilon^0$ and $k = \Upsilon^0 + 1$. Furthermore, Υ^0 can be determined in $\mathcal{O}(n\log n)$ time by solving $\mathbf{R\text{-}PP2}_{SC}$ (Geunes et al., 2004). Therefore, the pricing problem for this special case can be solved in $\mathcal{O}(n\log n)$ time. In what follows, we discuss a sorting based heuristic method, similar to Algorithm 1, along with another iterative heuristic method for the pricing problem in the general case.

2.1.3 Heuristic Approaches to the General Pricing Problem

As mentioned above, the pricing problem is NP-hard in the general case. Nevertheless, one does not necessarily need to solve the pricing problem optimally to generate a column with negative reduced cost. Heuristic methods have been commonly employed as column generators for complex pricing problems (see, e.g., Archetti et al., 2011, Min et al., 2011, Salani and Vacca, 2011). Therefore, in what follows, we discuss two heuristic methods for column generation.

The first heuristic is based on a sorting scheme similar to that in Algorithm 1. Suppose that the optimal dual values are given at a specific node, i.e., $\pi_i \ \forall i \in I$ are known. The heuristic method first starts with a column representing a solution in which all of the items are consolidated together. For this consolidation, the optimal shipment policy is determined using Equation (5), i.e., the order volume for each item and the common replenishment cycle length are determined. Therefore, one determines the negative reduced cost associated with this column, i.e., objective function value of **PP2** when n items are consolidated. The heuristic method then moves to another column, which represents a consolidation with $n-1$ items. In moving from the n-item consolidation to an $(n-1)$-item consolidation, one item is excluded based on a heuristic sorting approach. In particular, each item i in the n-item consolidation is assigned a weight, w_i, which is intended to measure item i's contribution to the reduced cost in the n-item consolidation. Then, the consolidation with $n-1$ items is defined by excluding the item with the maximum weight from the n-item consolidation. The item with the maximum weight is excluded because we would ideally like to find a consolidation with minimum reduced cost. Then the reduced cost of the $(n-1)$-item consolidation is determined, weights for each of the $n-1$ items are calculated, and an $(n-2)$-item consolidation is generated by excluding the item with the maximum weight from the $(n-1)$-item consolidation.

The weight of item i in a k-item consolidation is defined as $w_i = -\pi_i + h_i\lambda_i T/2 + a_i/T + R\Upsilon/(kT)$, where T and Υ are given by Equation (5). One can note that $\sum_{i=1}^{n} w_i x_i$ gives the objective function value of **PP2** for the given k-item consolidation. The heuristic method, which we refer to as Sorting-based-exclusion heuristic method (**SE-H**), calculates the reduced cost for each k-item consolidation for $k = n, n-1, \ldots, 1$ and returns the consolidation with the minimum reduced cost. Appendix 5.5 gives the formal statement of **SE-H**.

The second heuristic method for solving the pricing problem proceeds in a similar way to **SE-H**. It starts with an n-item consolidation and excludes one item to generate an $(n-1)$-item consolidation. However, in an intermediate iteration, instead of using weights for moving from a k-item consolidation to a $(k-1)$-item consolidation, it checks all possibilities for excluding an item. In particular, a total of k different $(k-1)$-item combinations are generated from the given k-item consolidation. Each of these $(k-1)$-item combinations corresponds to the exclusion of one of the k items from the given k-item consolidation. Then, the $(k-1)$-item combination with the minimum reduced cost is selected and used to generate the $(k-2)$-item consolidation. Similar to **SE-H**, the heuristic method, which we refer to as Best-exclusion heuristic method (**BE-H**), finds the reduced cost for each k-item consolidation for $k = n, n-1, \ldots, 1$ and returns the consolidation with the minimum reduced cost. Appendix 5.6 gives the formal statement of **BE-H**.

Both of these heuristics run in $\mathcal{O}(n^2)$ time. In our numerical studies, we compare the solutions achieved by the heuristics with the optimal solution for numerous small-size problem instances achieved through total enumeration. For larger size problem instances, we compare the heuristic methods with BARON, a commercial solver for MINLP problems. Our numerical analyses indicate that the heuristic methods are quite efficient in terms of both solution time and solution quality. Therefore, we expect that when these are embedded within the branch-and-price scheme, they will be successful as column generators. Next, we discuss two heuristic methods for solving the set partitioning problem, **P**, defined in Section 2.

2.1.4 Heuristic Approaches to the Set Partitioning Problem

In this section, as an alternative to the branch-and-price method, we propose two heuristic methods for solving problem **P**. The first heuristic method iteratively constructs a partition by forming subsets that will be included in the partition. The second method, on the other hand, starts with a set containing all of the items and forms the subsets from this that will be included in the partition.

The first heuristic method, which we refer to as the *Partitioning via Integration heuristic* (**PI-H**), works as follows. In an intermediate iteration, suppose that we have an infeasible partition with a set of subsets such that some of the items are not included in the partition. We randomly select one item from the set of excluded items, denoted by E. Two options are considered for this item: (i) it can be integrated into one of the subsets of the current partition or (ii) it can be integrated into the current partition as a new subset by itself. We first consider option (i) and determine the subset into which the item will be integrated, so that the increase in the total cost due to integration is minimized. As a result of option (i), a new partition (with the same number of subsets) is formed and its total cost is known. Then, we consider option (ii), which forms a new partition that includes one additional subset different from those in the current partition and its total cost is equal to the cost of the current partition plus the cost of ordering the selected item individually. Finally, the option which results in a partition with lower total cost is chosen. Starting with the case when all of the items are excluded, this process is repeated until no excluded item remains, i.e., a feasible partition is formed and $E = \emptyset$. Appendix 5.7 states **PI-H**, which runs in $\mathcal{O}(n^2)$ time.

The second heuristic method, which we refer to as the *Partitioning via Exclusion heuristic* (**PE-H**), works as follows. In an intermediate iteration, suppose that we have a set of subsets such that some of the items are not included within these subsets. Let E denote the set of items excluded. We execute **BE-H** with E, such that $\pi_i = 0 \ \forall i \in S$ (note that one needs the values of $\pi_i \ \forall i \in S$ to execute **BE-H**, and when $\pi_i = 0 \ \forall i \in S$, **BE-H** seeks a consolidation with low costs instead of low reduced costs). This returns a subset from the set of excluded items (note that one may apply any heuristic to form a subset from a given set of items; however, we use **BE-H** instead of **SE-H**, as our numerical experiments indicate that **BE-H** outperforms **SE-H** in terms of solution quality). Then, the current set of subsets is expanded by including the subset generated by **BE-H** and the set of excluded items is reduced. Starting with the case when all of the items are excluded, this process is repeated until no excluded item remains, i.e., a feasible partition is formed and $E = \emptyset$. Appendix 5.8 states **PE-H**, which runs in $\mathcal{O}(n^3)$ time.

3 Findings and Applications

In this section, we first focus on demonstrating the efficiency of the heuristic methods we have discussed for the pricing problem. Following this, we discuss our results for the branch-and-price and heuristic methods for solving the set partitioning problem. Then, we demonstrate how shipment consolidation leads to reduced truck density on the distribution network and provide sensitivity analysis on the benefits of shipment consolidation.

3.1 Efficiency of the Pricing Heuristic Methods

To analyze the efficiency of the pricing heuristic methods, we first compare **SE-H** and **BE-H** with total enumeration for small size problems. Then, we compare **SE-H** and **BE-H** with BARON for larger size problems. The heuristic methods are coded in MATLAB, and GAMS is utilized to solve the pricing problems via BARON.

To compare the pricing heuristic methods with total enumeration, for each $n = \{5, 10, 15, 20\}$, we solve a randomly generated problem instance from each of 32 combinations of $\lambda \sim \{U[1000, 1500], U[1500, 2000]\}$, $\boldsymbol{a} \sim \{U[250, 500], U[500, 750]\}$, $\boldsymbol{h} \sim \{U[2, 4], U[4, 6]\}$, $P = \{750, 1000\}$, and $R = \{500, 750\}$, where λ, $\boldsymbol{a}$, and $\boldsymbol{h}$ denote n-vectors of λ_i, a_i, and h_i values, respectively. Furthermore, $U[l, u]$ denotes a uniform distribution with lower bound l and upper bound u. We note that when solving a pricing problem within the branch-and-price method, we have the precise π_i values. However, our aim here is to analyze the efficiency of the pricing heuristic methods for any given problem parameters. Therefore, in determining the dual values required in the pricing problem, for each problem instance we let $\pi_i \sim U[0, z_i]$, where z_i denotes the value of Equation (2) with a_i, d_i, and h_i. The rationale behind this selection of π_i values is as follows. The dual problem of **LRP**, stated in Section 2.1, specifies an upper bound on each dual variable (i.e., we have the constraints $\pi_i \leq u_i \ \forall i \in I$ in the dual of **LRP**) such that this upper bound is the minimum cost of replenishing this item independently of the others. Thus, we assume that π values are bounded by the z_i values. Table 1 documents the average values, over all 32 problem instances solved for each n, for the reduced cost of the column found by total enumeration (opt. value), and the reduced costs of the best columns (best value) found by the pricing heuristic methods, along with the computational times for total enumeration and the pricing heuristics in seconds, and the optimality gap (gap) of the pricing heuristic methods.

Table 1: Comparison of Total Enumeration and The Pricing Heuristic Methods

	Enumeration		**SE-H**			**BE-H**		
n	Opt. Value	Time	Best Value	Time	Gap	Best Value	Time	Gap
5	657.66	0.001	675.33	0.000	2.69%	657.66	0.002	0.00%
10	221.04	0.031	223.57	0.000	1.14%	221.04	0.001	0.00%
15	65.63	1.720	76.53	0.001	16.60%	67.08	0.003	2.20%
20	-57.32	2288.249	-47.86	0.002	16.50%	-55.63	0.008	2.96%
avg	221.75	572.500	231.89	0.001	9.23%	222.54	0.004	1.29%

As is clear from Table 1, both pricing heuristic methods are far more efficient than total enumeration in computational time. Moreover, **BE-H** finds the optimal solution for all of the problem instances solved for $n = 5$ and $n = 10$, resulting in 0% optimality gap, and the optimality gap of **BE-H** is less than 3% for $n = 15$ and $n = 20$. While **BE-H** has a 1.29% optimality gap on average, **SE-H**, which is slightly faster than **BE-H**, has a 9.23% optimality gap on average.

For larger size problem instances, we compare **SE-H** and **BE-H** with GAMS/BARON for $n = \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$. For each n, we solve a randomly generated problem instance from each of 32 combinations of $\lambda \sim \{U[1000, 1500], U[1500, 2000]\}$, $\boldsymbol{a} \sim \{U[250, 500], U[500, 750]\}$, $\boldsymbol{h} \sim \{U[2, 4], U[4, 6]\}$, $P = \{750, 1000\}$, and $R = \{500, 750\}$. The dual prices for any problem instance are generated as explained previously. The time limit used for BARON was 1000 seconds. Table 2 summarizes the average values, over all 32 problem instances solved for each n, for the reduced costs of the best columns (best value) found by BARON and the pricing heuristic methods along with the computational times of each method.

It can be observed from Table 2 that the pricing heuristic methods are computationally more efficient than BARON. In particular, BARON always terminated due to the time limit imposed in GAMS. While BARON has the best average result in terms of the solution quality, **BE-H** is within less than 2% of

Table 2: Comparison of BARON and The Pricing Heuristic Methods

	BARON		**SE-H**		**BE-H**	
n	Best Value	Time	Best Value	Time	Best Value	Time
10	225.47	1004.309	230.07	0.000	225.47	0.006
15	61.92	1003.044	63.64	0.001	61.92	0.004
20	-47.06	1002.297	-43.46	0.001	-47.06	0.009
25	-124.89	1000.616	-116.01	0.001	-124.89	0.015
30	-183.06	1001.289	-173.19	0.002	-181.27	0.017
35	-222.04	1002.160	-217.56	0.002	-217.89	0.019
40	-245.55	1000.861	-238.38	0.002	-247.14	0.029
45	-302.92	1001.368	-295.15	0.003	-295.15	0.032
50	-450.64	1000.852	-444.96	0.002	-444.96	0.042
avg	-143.20	1001.866	-137.22	0.002	-141.22	0.019

BARON on average. Furthermore, **BE-H** was able to find a lower average reduced cost for the case of $n = 40$. Both BARON and **BE-H** outperformed **SE-H** in solution quality and **BE-H** is very efficient computationally. Therefore, in our analysis of the branch-and-price method, we use **BE-H** as a heuristic column generator.

3.2 Efficiency of the Set Partitioning Heuristic Methods

To analyze the efficiency of the heuristic set partitioning methods, we first compare **PI-H** and **PE-H** with CPLEX for small size problems. In particular, for small size problems, all of the possible subsets of the items and their costs can be generated; thus, the set partitioning problem **P** corresponds to a pure integer programming problem, which can be solved via CPLEX. On the other hand, for larger problem sizes, the data generation time is very long; hence, we compare **PI-H** and **PE-H** with the branch-and-price method (**B&P**).

To compare the set partitioning heuristic methods with GAMS/CPLEX, for each $n = \{5, 10, 15, 20\}$, we solve a randomly generated problem instance from each of 32 combinations of $\lambda \sim \{U[1000, 1500], U[1500, 2000]\}$, $\boldsymbol{a} \sim \{U[250, 500], U[500, 750]\}$, $\boldsymbol{h} \sim \{U[2, 4], U[4, 6]\}$, $P = \{750, 1000\}$, and $R = \{500, 750\}$. The time limit for CPLEX was set to 1000 seconds and the relative gap was defined to be 0.001 in GAMS (the default relative gap is 0.1; however, in this case, the heuristic methods were much more effective than CPLEX in terms of solution quality and solution time as CPLEX terminated after analyzing a limited number of integer solutions due to the relative gap). Table 3 documents the average values, over all 32 problem instances solved for each n, for the total cost of the best partition (best value) found by CPLEX and the set partitioning heuristic methods, along with the data generation time (DGT) required for CPLEX, and the computational times of CPLEX and the set partitioning heuristic methods in seconds.

Table 3: Comparison of CPLEX and The Set Partitioning Heuristic Methods

	CPLEX			**PI-H**		**PE-H**	
n	Best Value	DGT	Time	Best Value	Time	Best Value	Time
5	17482.99	0.002	0.206	17512.60	0.000	17498.40	0.002
10	35015.21	0.031	0.248	35116.81	0.002	35095.24	0.003
15	52326.27	1.611	3.729	52475.54	0.003	52449.88	0.003
20	69440.42	2140.210	660.248	69482.39	0.003	69473.39	0.006
avg	43566.23	535.463	166.108	43646.83	0.002	43629.23	0.004

It follows from Table 3 that both of the set partitioning heuristic methods outperform CPLEX in computational time. Specifically, as n increases, the relative time efficiency of the heuristic methods drastically increases. Furthermore, the data generation time increases exponentially (it takes more than a day to generate data for CPLEX when $n = 25$). While CPLEX is slightly better than the heuristic methods in terms of solution quality, both heuristic methods were able to find solutions that were within less than 1% of CPLEX; hence, the heuristic methods are quite efficient. When **PI-H** and **PE-H** are compared, it is observed that **PE-H** is slightly better in terms of solution quality.

For larger problem sizes, we compare the set partitioning heuristic methods with **B&P**. The implementation details of **B&P** are explained in Appendix 5.9. In particular, since a heuristic method (**BE-H**) is used for solving the pricing problems, **B&P** is also a heuristic method. For each $n = \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$, we randomly generate a problem instance for each of the 32 combinations of $\lambda \sim \{U[1000, 1500], U[1500, 2000]\}$, $\boldsymbol{a} \sim \{U[250, 500], U[500, 750]\}$, $\boldsymbol{h} \sim \{U[2, 4], U[4, 6]\}$, $P = \{750, 1000\}$, and $R = \{500, 750\}$. Table 4 summarizes the average values, over all 32 problem instances solved for each n, for the total cost of the best partition (best value) found by **B&P** and the set partitioning heuristic methods, and the computational times of **B&P** and the set partitioning heuristic methods in seconds. Furthermore, the average number of nodes analyzed (# of nodes), average number of columns generated in nodes (avg # of columns), and average solution time required for nodes (avg node time) are documented for the branch-and-price method.

It follows from Table 4 that both of the set partitioning heuristics outperform **B&P** in computational time. We also observe that **B&P** is able to find better solutions. However, the average improvement due to **B&P** over both of the heuristic set partitioning methods is less than 2.6% for each n value and the average improvement due to **B&P** is around 1.3%. Furthermore, the improvement in the objective function due to **B&P** decreases as the problem size increases. It is worth noting that the number of nodes analyzed, average number of columns generated in solving the LRP at the nodes, the average solution time of a node, and the total time required for **B&P** increase as the number of items increases. We can conclude that the set partitioning heuristics are very efficient, as they outperform **B&P** with

Table 4: Comparison of The Branch-and-Price and The Set Partitioning Heuristic Methods

	Branch-and-Price					**PI-H**		**PE-H**	
n	# of Nodes	Avg # of Columns	Avg Node Time	Best Value	Time	Best Value	Time	Best Value	Time
10	97.06	13.10	0.151	34018.73	10.709	34920.14	0.002	34894.19	0.004
15	166.23	19.19	0.293	51171.89	36.524	52454.03	0.006	52438.21	0.004
20	227.06	26.67	0.531	68577.53	74.801	69823.75	0.006	69799.89	0.007
25	273.21	26.35	0.671	85768.02	114.597	87493.52	0.007	87475.12	0.007
30	583.78	35.78	1.338	102976.45	457.269	104631.05	0.010	104609.11	0.009
35	606.21	40.34	2.055	120547.65	671.874	121974.94	0.009	121953.78	0.013
40	611.44	47.38	3.034	138530.71	1401.903	139820.07	0.011	139798.43	0.017
45	672.39	46.23	3.322	155640.78	1653.121	157310.93	0.010	157306.02	0.023
50	969.44	49.38	3.489	173078.32	2361.469	174303.78	0.011	174299.94	0.025
avg	467.42	33.82	1.654	103367.79	753.585	104748.02	0.008	104730.52	0.012

respect to computational time, and their solution qualities are very close to **B&P**. When **PI-H** and **PE-H** are compared, it is observed that **PE-H** is slightly better in terms of solution quality. Therefore, in the following analyses, we use **PE-H** as our solution method for the multi-item EOQ model with shipment consolidation and explicit truckload transportation costs.

3.3 Cost and Environmental Benefits of Shipment Consolidation

This section discusses how the shipment consolidation approach we have introduced in this paper reduces the total costs per unit time as well as the number of trucks used for shipments compared to the scenario when the items are independently shipped, which we refer to as *independent shipment*. Furthermore, we quantify the decrease in truck density on the transportation network and increase in truck capacity utilization due to shipment consolidation. We document how the changes in (i) the number of items, (ii) demand levels, (iii) fixed order costs, (iv) holding costs per unit volume per unit time, (v) per truck capacity, and (vi) per truck cost affect the total shipment costs per unit time (total cost), total number of truck trips for a single shipment of each item (truck number), total truckload transportation cost per unit time (truck cost), truck utilization, and the number of trucks used for shipment per unit time (truck density). All of the associated tables are given in Appendix 5.10.

To analyze the effects of any of the aforementioned parameters, we consider different combinations of $n = \{150, 75, 100\}$, $\lambda \sim \{U[1000, 1500], U[1500, 2000], U[2000, 2500]\}$, $\boldsymbol{a} \sim \{U[250, 500], U[500, 750], U[750, 1000]\}$, $\boldsymbol{h} \sim \{U[2, 4], U[4, 6], U[6, 8]\}$, $P = \{750, 1000, 1250\}$, and $R = \{500, 750, 1000\}$, changing the parameter of interest as indicated in Tables 5-10. Ten randomly generated problem instances are solved for each of 243 combinations for each specific value of the parameter (i.e., 2340 problem instances are solved for each specific value of the parameter of interest).

(i) Effects of n: Table 5 summarizes the results for each n value. We have the following observations based on Table 5.

- As n increases, as expected, total cost, number of trucks, truck cost, and truck density increase under any shipment policy. It is observed that truck utilization follows a stable pattern for both independent shipment and shipment consolidation cases.
- The percent increase in truck utilization and percent decrease in truck density due to shipment consolidation follow a stable pattern as n increases. On the other hand, the percent reduction in total cost, number of trucks, and truck cost due to shipment consolidation slightly increase as n increases.

The observations stated above suggest that the cost savings due to shipment consolidation increase as the number of items considered increases, while improvements in truck utilization and truck density due to shipment consolidation remain the same.

(i) Effects of λ: Table 6 summarizes the results for each λ value. We have the following observations based on Table 6.

- As λ increases, as expected, total cost, number of trucks, truck cost, and truck density increase under any shipment policy. While the truck utilization under independent shipment increases with λ, truck utilization under shipment consolidation follows a stable pattern.
- The percent reduction in total cost, number of trucks, truck cost, truck density and the percent increase in truck utilization due to shipment consolidation diminish as λ increases.

The observations stated above suggest that the benefits of shipment consolidation are greater when the items have lower demand rates. This is expected as full truck loads are justifiable when the demand rate is high for an item, hence, independent shipment results in more full truck loads for higher demand rates than it would for lower demand rates. Nevertheless, shipment consolidation is observed to be superior to independent shipment in all demand cases.

(i) Effects of $\boldsymbol{h}$: Table 7 summarizes the results for each $\boldsymbol{h}$ value. We have the following observations based on Table 7.

- As $\boldsymbol{h}$ increases, total cost increases under any shipment policy as expected. Under independent shipments, while the total number of trucks decreases, truck cost and truck density increase with $\boldsymbol{h}$. Furthermore, truck utilization decreases. These imply that, under independent shipments, higher holding costs result in smaller but more frequent shipments by under-utilized trucks compared to the case of lower holding costs.

- Under shipment consolidation, similar to independent shipments, the total number of trucks decreases with $\boldsymbol{h}$; however, truck cost, truck utilization, and truck density follow a stable pattern.
- The percent reduction in total cost, number of trucks, truck cost, truck density and the percent increase in truck utilization due to shipment consolidation increase as $\boldsymbol{h}$ increases.

The observations stated above suggest that the benefits of shipment consolidation are greater when the items are subject to high holding costs. Furthermore, while truck utilization, truck density, and truck costs under shipment consolidation are not affected by changes in holding costs, these values are negatively affected under independent shipments as holding costs increase.

(i) Effects of $\boldsymbol{a}$: Table 8 summarizes the results for each $\boldsymbol{a}$ value. We have the following observations based on Table 8.

- As $\boldsymbol{a}$ increases, total cost and the total number of trucks increase under any shipment policy. This is expected as higher fixed order costs result in larger orders, hence, more trucks are used for shipments. Under independent shipments, while the total number of trucks increases, truck cost and truck density decrease with $\boldsymbol{a}$. Furthermore, truck utilization increases. These imply that, under independent shipments, higher fixed order costs result in larger but less frequent shipments by better utilized trucks compared to the case of lower fixed order costs.
- Under shipment consolidation, similar to independent shipments, the total number of trucks increases with $\boldsymbol{a}$; however, truck cost and truck density follow a stable pattern and truck utilization decreases very slightly.
- The percent reduction in total cost, number of trucks, truck cost, truck density and the percent increase in truck utilization due to shipment consolidation decrease as $\boldsymbol{a}$ increases.

The observations stated above suggest that the benefits of shipment consolidation are greater when the items are subject to low fixed order costs. Furthermore, while truck utilization, truck density, and truck costs under shipment consolidation are not affected by changes in fixed order costs, these values are negatively affected under independent shipments as fixed order costs decrease.

(i) Effects of P: Table 9 summarizes the results for each P value. We have the following observations based on Table 9.

- As P increases, as expected, total cost, total number of trucks, truck cost, and truck density decrease under any shipment policy. Nevertheless, the rate of decrease in these values is higher under shipment consolidation. Truck utilization under independent shipments decreases as P increases whereas it follows a stable pattern under shipment consolidation.

- The percent reduction in total cost, number of trucks, truck cost, truck density and the percent increase in truck utilization due to shipment consolidation increase as P increases.

The observations stated above suggest that the benefits of shipment consolidation are greater when trucks used for shipments have high capacities.

(i) Effects of R: Table 10 summarizes the results for each R value. We have the following observations based on Table 10.

- As R increases, as expected, total cost and truck cost increase under any shipment policy. Under independent shipments, while the total number of trucks follows a stable pattern, truck utilization increases, hence, truck density decreases as R increases.

- Similar to independent shipments, the total number of trucks under shipment consolidation follows a stable pattern as R increases. While truck utilization slightly increases, truck density follows a stable pattern under shipment consolidation.

- The percent reduction in total cost, truck cost, truck density and the percent increase in truck utilization due to shipment consolidation decrease as R increases. On the other hand, the reduction in the total number of trucks due to shipment consolidation is stable for different R values.

The observations stated above suggest that the benefits of shipment consolidation are greater when the trucks used for shipments have lower costs.

4 Conclusions, Recommendations, and Suggested Research

This study analyzed shipment consolidation policies with explicit truckload transportation costs in a multi-item inventory system. To the best of our knowledge, shipment consolidation problems with explicit truckload transportation costs in multi-item inventory systems have not been analyzed in the literature. To analyze this problem, we formulated a set partitioning problem and proposed a branch-and-price method for solving the set partitioning problem. The pricing problem associated with the branch-and-price method was shown to be NP-hard. Therefore, we provided two heuristic methods to solve the pricing problem. For a practical special case of the pricing problem, we showed that the pricing problem can be solved to optimality in polynomial time. This special case extends the EOQ model with market choice flexibility defined and analyzed in Geunes et al. (2004) by modeling explicit truckload transportation costs. Furthermore, as alternatives to the branch-and-price method, two heuristic methods are discussed for the set partitioning problem of interest in this study.

An extensive set of numerical studies was documented to analyze the efficiency of the heuristic methods proposed for the pricing subproblem and the set partitioning problem. The first set of numerical studies indicated that the pricing heuristic methods are quite efficient compared to BARON. The second set of numerical studies demonstrated the efficiency of the set partitioning heuristic methods. The last set of numerical studies focused on sensitivity analyses of each problem parameter with respect to the benefits of shipment consolidation. We concluded that in case of lower demand, higher holding costs, lower fixed order costs, higher per truck capacity, or lower per truck costs scenarios, the benefits of shipment consolidation are more substantial. Particularly, in such scenarios, the decrease in truck density, total costs, and transportation costs, and the increase in truck utilization due to shipment consolidation are greater.

The consolidation policy analyzed in this study leads to a decreased number of trucks used to ship same total amount of commodities, i.e., reduced truck density and increased truck capacity utilization. Considering the need for low CO_2 emissions in transportation, this study ideally is able to propose policies for greener transportation in supply chains. Furthermore, these policies lead to less truck congestion on the distribution network. Future research directions include analyzing shipment consolidation policies for multi-item inventory systems with stochastic demands. Furthermore, introducing different truck types, with distinct per truck costs and per truck capacities, remain as a future research direction. Another future research direction would explicitly account for truck routes in forming consolidated sets of items.

Acknowledgements

This work was partially supported by a grant from the Center for Multimodal Solutions for Congestion Mitigation (Project #2010-018) a University Transportation Center supported by the USDOT.

References

Archetti, C., Bouchard, M., Desaulniers, G., 2011. Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. Transportation Science 45 (3), 285–298.

Aucamp, D., 1982. Nonlinear freight costs in the EOQ problem. European Journal of Operations Research 9, 61–62.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., P.Savelsbergh, M. W., Vance, P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46 (3), 316–329.

Ben-Khedher, N., Yano, C. A., 1994. The multi-item joint replenishment problem with transportation and container effects. Transportation Science 28, 37–54.

Bookbinder, J. H., Higgison, J. K., 2002. Probabilistic modeling of freight consolidation by private carriage. Transportation Research Part E 38 (5), 305–318.

Çetinkaya, S., 2005. Coordination of inventory and shipment consolidation decisions: a review of premises, models, and justification. In: Geunes, J., Akçali, E., Pardalos, P. M., Romeijn, E. H. (Eds.), Applications of Supply Chain Management and e-commerce research in industry. Kluwer, New York, Ch. 1.

Çetinkaya, S., Bookbinder, J. H., 2003. Stochastic models for the dispatch of consolidated shipments. Transportation Research Part B 37 (8), 747–768.

Çetinkaya, S., Lee, C.-Y., 2000. Stock replenishment and shipment scheduling for Vendor-Managed inventory systems. Management Science 46 (2), 217–232.

Çetinkaya, S., Lee, C.-Y., 2002. Outbound transportation: modeling inventory and cargo capacity. Naval Research Logistics 49, 531–556.

Çetinkaya, S., Mutlu, F., Lee, C.-Y., 2006. A comparison of outbound dispatch policies for integrated inventory and transportation decisions. European Journal of Operational Research 171 (3), 1094–1112.

Chen, F. Y., Wang, T., Xu, T. Z., 2005. Integrated inventory replenishment and temporal shipment consolidation: A comparison of quantity-based and time-based models. Annals of Operations Research 135 (1), 197–210.

Ching, W.-K., Tai, A. H., 2005. A quantity-time-based dispatching policy for a VMI system. In: Computational Science and Its Applications-ICCSA 2005: International Conference.

Degraeve, Z., Jans, R., 2007. A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. Operations Research 55 (5), 909–920.

Díaz, J. A., Fernández, E., 2002. A branch-and-price algorithm for the single source capacitated plant location problem. The Journal of the Operational Research Society, 53 (7), 728–740.

Freling, R., Romeijn, H. E., Morales, D. R., Wagelmans, A. P. M., 2003. A branch-and-price algorithm for the multiperiod single-sourcing problem. Operations Research 51 (6), 922–939.

Garey, M. R., Johnson, D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY.

Geunes, J., Shen, Z.-J., Romeijn, H. E., 2004. Economic ordering decisions with market choice flexibility. Naval Research Logistics 51 (1), 117–136.

Goyal, S., 1974. Determination of optimum packaging frequency of items jointly replenished. Management Science 21 (4), 436–443.

Goyal, S., 1975. Analysis of joint replenishment inventory systems with resource restriction. Operational Research Quarterly 26 (1), 197–203.

Higgison, J. K., 1995. Recurrent decision approaches to shipment-release timing in freight consolidation. International Journal of Physical Distribution & Logistics Management 25 (5), 3–23.

Higgison, J. K., Bookbinder, J. H., 1994. Policy recommendations for a shipment consolidation program. Journal of Business Logistics 15 (1), 87–112.

Higgison, J. K., Bookbinder, J. H., 1995. Markovian decision processes in shipment consolidation. Transportation Science 29 (3), 242–255.

Hwang, H., Moon, D. H., Shinn, S. W., 1990. An EOQ model with quantity discounts for both purchasing and freight cost. Computers and Operations Research 17 (1), 73–78.

Jackson, P., Maxwell, W., Muckstadt, J., 1985. The joint replenishment problem with a powers-of-two restriction. IIE Transactions 17 (1), 25–32.

Kang, J.-H., Kim, Y.-D., 2010. Coordination of inventory and transportation managements in a two-level supply chain. International Journal of Production Economics 123, 137–145.

Khouja, M., Goyal, S., 2008. A review of the joint replenishment problem literature: 19892005. European Journal of Operational Research 186, 1–16.

Lee, C.-Y., 1986. The economic order quantity for freight discount costs. IIE Transactions 18 (3), 318–320.

Lee, C.-Y., Çetinkaya, S., Jaruphongsa, W., 2003. A dynamic model for inventory lot sizing and outbound shipment scheduling at a third- party warehouse. Operations Research 51 (5), 735–747.

Levi, R., Roundy, R., Shmoys, D., Sviridenko, M., 2008. A constant approximation algorithm for the one-warehouse multiretailer problem. Management Science 54, 763–776.

Lübbecke, M. E., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53 (6), 1007–1023.

Lulli, G., Sen, S., 2004. A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. Management Science 50 (6), 786–796.

Mendoza, A., Ventura, J. A., 2008. Incorporating quantity discounts to the EOQ model with transportation costs. International Journal of Production Economics 113 (2), 754–765.

Min, Y.-H., Park, M.-J., Hong, S.-P., Hong, S.-H., 2011. An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. Transportation Research Part B 45 (2), 409–429.

Moon, I., Cha, B., Lee, C., 2010. The joint replenishment and freight consolidation of a warehouse in a supply chain. International Journal of Production Economics 133 (1), 344–350.

Mutlu, F., Çetinkaya, S., Bookbinder, J. H., 2010. An analytical model for computing the optimal time-and-quantity-based policy for consolidated shipments. IIE Transactions 42 (5), 367–377.

Palubeckis, G., 2004. Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. Annals of Operations Research 131 (1-4), 259–282.

Salani, M., Vacca, I., 2011. Branch and price for the vehicle routing problem with discrete split deliveries and time windows. European Journal of Operational Research 213 (3), 470–477.

Shen, Z.-J. M., Coullard, C., Daskin, M. S., 2003. A joint location-inventory model. Transportation Science 37 (1), 40–55.

Silver, E. A., 1975. Modifying the economic order quantity (EOQ) to handle coordinated replenishment of two or more items. Production & Inventory Management 16, 26–38.

Silver, E. A., 1976. A simple method of determining order quantities in jointly replenishments under deterministic demand. Management Science 22 (12), 1351–1361.

Silver, E. A., Pyke, D. F., Peterson, R., 1998. Inventory Management and Production Planning and Scheduling, 3rd Edition. John Wiley & Sons, New York.

Sindhuchao, S., Romeijn, H. E., Akcali, E., Boondiskulchok, R., 2005. An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. Journal of Global Optimization 32, 93–118.

Toptal, A., 2009. Replenishment decisions under an all-units discount schedule and stepwise freight costs. European Journal of Operations Research 198, 504–510.

Toptal, A., Çetinkaya, S., 2006. Contractual agreements for coordination and vendor-managed delivery under explicit transportation considerations. Naval Research Logistics 53 (5), 397–417.

Toptal, A., Çetinkaya, S., Lee, C.-Y., 2003. The buyer-vendor coordination problem: Modeling inbound and outbound cargo capacity and costs. IIE Transactions 35 (11), 987 – 1002.

Viswanathan, S., 1996. A new optimal algorithm for the joint replenishment problem. The Journal of the Operational Research Society 47 (7), 936–944.

Viswanathan, S., 2002. On optimal algorithms for the joint replenishment problem. The Journal of the Operational Research Society 53 (11), 1286–1290.

Wildeman, R., Frenk, J., Dekker, R., 1997. An efficient optimal solution method for the joint replenishment problem. European Journal of Operational Research 99 (2), 433–444.

Wilhelm, W. E., 2001. A technical review of column generation in integer programming. Optimization and Engineering 2 (2), 159–200.

Zhang, J.-L., Lee, C.-Y., Chen, J., 2009. Inventory control problem with freight cost and stochastic demand. Operations Research Letters 37 (6), 443–446.

Zhao, Q.-H., Wangb, S.-Y., Laic, K.-K., Xia, G.-P., 2004. Model and algorithm of an inventory problem with the consideration of transportation cost. Computers and Industrial Engineering 46 (2), 389–397.

5 Appendix

5.1 Summary of the Notation Used

i : index for items, $i = 1, 2, \ldots, n$,
S : set of items, $|S| = n$,
j : index for the subsets of S, $j = 1, 2, \ldots, 2^n - 1$,
S_j: j^{th} subset of S,
c_i : per unit volume procurement cost of item i,
h_i : per unit volume per unit time holding cost of item i,
a_i : fixed order cost of item i,
λ_i: demand rate in volumes for item i,
P : per truck capacity,
R : per truck cost,
t_i : replenishment cycle length of item i,
v_i : replenishment volume of item i,
$f_i(v_i)$: total cost per unit time excluding shipment costs for item i,
v_i^{eoq}: minimizer of $f_i(v_i)$,
$g_i(v_i)$: total cost per unit time including shipment costs for item i,
$v^{(i)}$: any minimizer of $g_i(v_i)$,
T_j : common replenishment cycle length for items in S_j,
V_j : total replenishment volume of the items in S_j, $V_j = T_j \sum_{i \in S_j} \lambda_i$,
$G_j(V_j)$: total cost per unit time for items in S_j including shipment costs,
V_j^* : any minimizer of $G_j(V_j)$,
x_{ij} : 1 if $i \in S_j$, 0 otherwise,
y_j : 1 if S_j is in the partition, 0 otherwise.

5.2 Proof of Property 2

We first relax the integrality of Υ in **PP2**, i.e., consider the following relaxation of **PP2**.

$$\begin{array}{ll} \min & -\sum_{i\in S}\pi_i x_i + \dfrac{V\sum_{i\in S} h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \dfrac{1}{V}\sum_{i\in S} a_i x_i \sum_{i\in S}\lambda_i x_i + \dfrac{R\Upsilon}{V}\sum_{i\in S}\lambda_i x_i \\ \text{s.t.} & (\Upsilon - 1)P < V \leq \Upsilon P, \\ & x_i \in \{0,1\},\ i \in S. \end{array}$$

For any given $\mathbf{x}$ and V, $\Upsilon = V/P$ in the solution of above relaxed problem; hence, it reduces to **R-PP2**. Let $(\mathbf{x}^0, V^0)$ be a solution of **R-PP2** and let $\Upsilon^0 = \lfloor V^0/P \rfloor$. Note that $(\mathbf{x}^0, V^0, \Upsilon^0)$ also solves

$$\begin{array}{ll} \min & -\sum_{i\in S}\pi_i x_i + \dfrac{V\sum_{i\in S} h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \dfrac{1}{V}\sum_{i\in S} a_i x_i \sum_{i\in S}\lambda_i x_i + \dfrac{R\Upsilon}{V}\sum_{i\in S}\lambda_i x_i \\ \text{s.t.} & (\Upsilon - 1)P < V \leq \Upsilon P, \\ & \Upsilon \leq \Upsilon^0 + 1, \\ & x_i \in \{0,1\},\ i \in S. \end{array}$$

This then implies that $\Upsilon^* \leq \Upsilon^0 + 1$. Similarly, $(\mathbf{x}^0, V^0, \Upsilon^0)$ also solves

$$
\begin{array}{ll}
\min & -\sum_{i\in S}\pi_i x_i + \dfrac{V\sum_{i\in S}h_i\lambda_i x_i}{2\sum_{i\in S}\lambda_i x_i} + \dfrac{1}{V}\sum_{i\in S}a_i x_i\sum_{i\in S}\lambda_i x_i + \dfrac{R\Upsilon}{V}\sum_{i\in S}\lambda_i x_i \\
\text{s.t.} & (\Upsilon-1)P < V \le \Upsilon P, \\
 & \Upsilon \ge \Upsilon^0, \\
 & x_i \in \{0,1\},\ i\in S.
\end{array}
$$

This then implies that $\Upsilon^* \ge \Upsilon^0$. Therefore, we have $\Upsilon^0 \le \Upsilon^* \le \Upsilon^0 + 1$. ■

5.3 Proof of Property 3

First consider the case when $k = \Upsilon^0 + 1$. Let $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$ be a solution of **PP2-**$\Upsilon^0 + 1$ such that $\tilde{V}^{\Upsilon^0+1} = \lim\limits_{V\downarrow\Upsilon^0 P} V$. In this case, the objective function value of **PP2-**$\Upsilon^0 + 1$ excluding truckload transportation costs at $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$ is approximately equal to the objective function value of **PP2-**Υ^0 excluding truckload transportation costs at $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$. Since, $\frac{R\Upsilon^0}{\Upsilon^0 P}\sum_{i\in S}\lambda_i x_i \le \frac{R(\Upsilon^0+1)}{\lim_{V\downarrow\Upsilon^0 P} V}\sum_{i\in S}\lambda_i x_i$, then the objective function value of **PP2-**Υ^0 at $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$ is smaller than the objective function value of **PP2-**$\Upsilon^0 + 1$ at $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$. Therefore, $(\mathbf{x}^{\Upsilon^0+1}, V^{\Upsilon^0+1})$ cannot solve **PP2** unless $\tilde{V}^{\Upsilon^0+1} \ne \lim\limits_{V\downarrow\Upsilon^0 P} V$. Now consider the case when $k = \Upsilon^0$. Let $(\mathbf{x}^{\Upsilon^0}, V^{\Upsilon^0})$ be a solution of **PP2-**Υ^0 such that $\tilde{V}^{\Upsilon^0} = \lim\limits_{V\downarrow(\Upsilon^0-1)P} V$. Similarly, it can be argued that the objective function value of **PP2-**$\Upsilon^0 - 1$ at $(\mathbf{x}^{\Upsilon^0}, V^{\Upsilon^0})$ is smaller than the objective function value of **PP2-**Υ^0 at $(\mathbf{x}^{\Upsilon^0}, V^{\Upsilon^0})$. Therefore, $(\mathbf{x}^{\Upsilon^0}, V^{\Upsilon^0})$ cannot solve **PP2** unless $\tilde{V}^{\Upsilon^0} \ne \lim\limits_{V\downarrow(\Upsilon^0-1)P} V$. ■

5.4 Proof of Property 4

The proof of Property 4 follows from the following observations.

1. $c_i(V) = ((Rk+a)/V - \pi_i/\lambda_i)\lambda_i$ is decreasing in V, hence, $c_i(V)$ values are minimized when $V = kP$. This then implies that the maximum number of items consolidated will be achieved when $V = kP$.

2. Let products be sorted in nondecreasing order of $c_i(kP)$ values. The order of the items based on $c_i(V)$ values, $V \in \big((k-1)P, kP\big]$, will be the same as V changes. That is, the order of items is the same for any $V \in \big((k-1)P, kP\big]$.

3. It is clear that for any given $V \in \big((k-1)P, kP\big]$, the corresponding $\mathbf{x}$ that minimizes the objective function value of **PP2-k**$_{SC}$ for the given V is achieved by assigning $x_i = 1$ when $c_i(V) = ((Rk + a)/V - \pi_i/\lambda_i)\lambda_i < 0$ and $x_i = 0$ otherwise. When $c_i(V) > 0\ \forall i \in S$, then $x_j = 1$ such that $j =\text{argmin}\{c_i(V) : i \in S\}$ and $x_i = 0\ \forall i \in S\backslash\{j\}$.

4. It follows from Observations 1–3 that when items $1, 2, \ldots, \ell$ are consolidated and $V = kP$, products $1, 2, \ldots, m$, with $m \leq \ell$ can be consolidated for $V \leq kP$.
5. Property 3 implies that V^k will be equal to either kP or $\sqrt{2(a + kR)\sum_{i\in S} \lambda_i x_i^k / h}$.

Then the correctness of Algorithm 1 follows from Observations 4–5. ∎

5.5 Pseudo-code for SE-H

Algorithm 2 *Sorting-based-exclusion heuristic method (**SE-H**) for the general pricing problem:*

*Define $z(\mathbf{x})$ as the optimum objective function value of **PP2** given $\mathbf{x}$.*
Let $\mathbf{x}$ be defined such that $x_i = 1,\ \forall i \in S$. Define $z^* = M$, where M is a large positive number.
For $k = n : 1 : -1$
    Calculate $z(\mathbf{x})$ and the associated T
        If $z(\mathbf{x}) < z^*$, set $z^* = z(\mathbf{x})$ and $\mathbf{x}^* = \mathbf{x}$.
    $i^* = 0$ and $w = -M$
    For $i = 1 : n$
        If $x_i = 1$, set $w_i = -\pi_i + \frac{h_i \lambda_i T}{2} + \frac{a_i}{T} + \frac{R\Upsilon}{kT}$
            If $w_i > w$, set $w = w_i$ and $i^* = i$
    End
    $x_{i^*} = 0$
End
Return $\mathbf{x}^*$.

5.6 Pseudo-code for BE-H

Algorithm 3 *Best-exclusion heuristic method (**BE-H**) for the general pricing problem:*

*Define $z(\mathbf{x})$ as the optimum objective function value of **PP2** given $\mathbf{x}$.*
Let $\mathbf{x}$ be defined such that $x_i = 1,\ \forall i \in S$. Define $z^{**} = M$, where M is a large positive number.
For $k = n : 1 : -1$
    Calculate $z(\mathbf{x})$
        If $z(\mathbf{x}) < z^{**}$, set $z^{**} = z(\mathbf{x})$ and $\mathbf{x}^* = \mathbf{x}$.
    Set $z^* = M$ and $j^* = 0$
    For $i = 1 : n$
        If $x_i = 1$, set $x_i = 0$ and calculate $z(\mathbf{x})$
            If $z(\mathbf{x}) < z^*$, set $i^* = i$

9: End

10: $x_{i^*} = 0$

11: End

12: Return $\mathbf{x}^*$.

5.7 Pseudo-code for PI-H

Recall that each subset can be represented by a binary n-vector and each item is itself a subset, i.e., item i is also represented by a binary n-vector such that its i^{th} entry is 1 and all other entries are 0. We use $\mathbf{e}^i$ to denote the binary n-vector defining item i. A partition can be represented by a set of binary n-vectors and we use $\mathbf{x}^j$ to denote the j^{th} component included in the partition.

Algorithm 4 *Partitioning via Integration heuristic method (**PI-H**) for problem **P**:*

0: Set $E = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \ldots, \boldsymbol{e}^n\}$ *and Partition* $= \emptyset$.

1: Randomly select $\boldsymbol{e}^i$ *from* E *and set* $E =: E \backslash \{\boldsymbol{e}^i\}$.

2: Set $g^* = M$ *and* $j^* = 0$.

3: For $j = 1 : |Partition|$

4: $\boldsymbol{x} = \boldsymbol{x}^j + \boldsymbol{e}^i$ *and calculate* $G(\boldsymbol{x})$

5: If $G(\boldsymbol{x}) - G(\boldsymbol{x}^j) < g^*$, $g^* = G(\boldsymbol{x}) - G(\boldsymbol{x}^j)$ *and* $j^* = j$.

6: End

7: If $g^* \leq G(\boldsymbol{e}^k)$, *update Partition by setting* $\boldsymbol{x}^{j^*} =: \boldsymbol{x}^{j^*} + \boldsymbol{e}^i$

8: Else $Partition =: Partition \cup \{\boldsymbol{e}^i\}$

9: If $E = \emptyset$, *stop and return Partition; else, go to Step 1.*

5.8 Pseudo-code for PE-H

Algorithm 5 *Partitioning via Best-exclusion heuristic method (**PE-H**) for problem **P**:*

0: Let $\pi_i = 0\ \forall i \in S$. *Set* $E = \{\boldsymbol{e}^1, \boldsymbol{e}^2, \ldots, \boldsymbol{e}^n\}$ *and Partition* $= \emptyset$.

*1: Execute **BE-H** with* E *and let* $\boldsymbol{x}$ *be the output.* $Partition =: Partition \cup \{\boldsymbol{x}\}$.

2: For $i = 1 : n$

3: If $x_i = 1$, $E =: E \backslash \{\boldsymbol{e}^i\}$

4: End

5: If $E = \emptyset$, *stop and return Partition; else, go to Step 1.*

5.9 Implementation Details of the Branch-and-Price Method

Here, we explain the implementation details of the branch-and-price method discussed in Section 2.1. CPLEX via GAMS is used in solution of any linear programming problems.

Branching rule: We branch on the binary decision variables, that is, y_j values. In particular, consider a feasible non-integer solution of a node. We choose the variable which is the closest to 0 or 1 to branch on. We define the first child by including the additional $y_j = 1$ constraint and the second child by including $y_j = 0$ constraint.

Node priority: We use a depth-first approach and solve the first child of a parent node. If the parent node is fathomed due to integrality, infeasibility, or bounded by the best integer solution, we move to its sibling. The reason behind applying a depth-first approach with priority given to the first child is due to the fact that as the branch-and-bound tree gets deeper, the LRP of the node can be reduced to a smaller LRP that we explain next.

Solving a node: Suppose that we need to solve a node. Each node is defined by problem **LRP** and the additional $y_j = 1 \ \forall j \in J^1$ and $y_j = 0 \ \forall j \in J^0$ type of constraints, where J^1 and J^0 define the sets of the subsets to be included and excluded, respectively, from the LRP solution of the node under consideration. First, we check whether the node is infeasible due to $y_j = 1$ type of constraints. If there exists at least one i such that $\sum_{j \in J^1} x_{ij} y_j > 1$, then the node is stated as infeasible since it violates the assignment constraints, and, therefore, it is fathomed. Otherwise, we reduce the LRP to a smaller LP, which we refer to as the *reduced LRP*. This reduction is executed as in the following simple example.

Suppose we have five items $S = \{1, 2, 3, 4, 5\}$. The current node has to include $S_1 = \{1\}$ and $S_2 = \{2\}$, that is, columns $[1, 0, 0, 0, 0]^t$ and $[0, 1, 0, 0, 0]^t$ has to be included, i.e., $y_1 = 1$ and $y_2 = 1$. Furthermore, let us assume that the current node has to exclude $S_3 = \{1, 2, 3\}$ and $S_4 = \{4, 5\}$, that is, columns $[1, 1, 1, 0, 0]^t$ and $[0, 0, 0, 1, 1]^t$ has to be excluded, i.e., $y_3 = 0$ and $y_4 = 0$. Since $S_1 \cup S_2 = \emptyset$, we cannot tell that the node is infeasible at this point. Now since items 1 and 2 are already covered, we eliminate them and we have $S^0 = \{3, 4, 5\}$. That is, we now have a smaller set of items that we need to solve the LRP for. Note that the reduced LRP will have no $y_j = 1$ type of constraints. Now suppose that we solve the reduced LRP with no $y_j = 1$ and no $y_j = 0$ type of constraints. It may be the case that $S_4 = \{4, 5\}$ and $S_5 = \{3\}$ define the solution of the reduced LRP. Together with the initially included subsets, this implies that S_1, S_2, S_4, S_5 define the solution of the original node. However, this solution is not feasible to the original node as we should have $y_4 = 0$. Here, we need to define $y_j = 0$ type of constraints for the reduced LRP. In particular, if S_j is to be excluded from the original LRP, it will also be excluded from the reduced LRP when it does not have any of the items that we will eliminate from the original set of items. For the example above, the reduced LRP will only consider items 3,4 and 5, and it has to exclude S_4.

After forming the reduced LRP, we apply column generation to solve it. To initiate the column generation for solving the reduced LRP of the node, a feasible starting solution and a modified column generator, which will not generate columns that has to be excluded from the solution of the reduced LRP, are needed.

Generating a feasible starting solution: We first see whether the column including all of the items, which is a feasible solution by itself, can be used as a starting solution. If the reduced LRP does not require the full column to be excluded, we choose it as the feasible starting solution. Else, we use a modified version **BE-H** to generate a starting feasible solution. We start with the full column as if it can be included, generate dual variables, then apply **BE-H** to generate a new column. We modify **BE-H** such that it never generates a column that should be excluded from the reduced LRP by assigning high reduced costs to the columns that has to be excluded from the reduced LRP. If the new column constitutes a feasible solution by itself, we take it as our starting solution. Else, it is included in the set of columns that should not be generated and we apply **BE-H** to generate one more new column. Then we check whether the newly generated columns constitute a feasible solution. This process is repeated until a feasible solution is constituted or **BE-H** generates a column that has already been generated. In the former case, a feasible starting solution is known, while we use the later case to fathom the node due to infeasibility.

Fathoming: A node is fathomed if its solution is integer, or the objective function value is greater than the best integer solution, or it is stated to be infeasible due to either $y_j = 1$ type of constraints or a feasible starting solution cannot be generated for the reduced LRP due to $y_j = 0$ type of constraints.

5.10 Tables

The tables used for the discussion of the cost and environmental benefits of shipment consolidation compared to independent ordering are stated below.

Table 5: Comparison of Independent Shipment with Shipment Consolidation for Different n Values

	Independent Ordering					Shipment Consolidation				
n	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
10	48898.96	10.19	15147.12	89.14%	20.41	45674.97	7.20	13712.06	99.95%	18.29
20	97715.49	20.37	30274.52	89.16%	40.79	91193.41	14.27	27411.51	99.95%	36.55
30	146673.89	30.55	45440.72	89.15%	61.21	136852.59	21.39	41138.00	99.97%	54.85
40	195506.64	40.75	60605.14	89.12%	81.64	182378.87	28.48	54851.69	99.96%	73.13
50	244374.90	50.91	75723.68	89.14%	102.02	227975.69	35.58	68546.51	99.95%	91.40
60	293188.16	61.11	90867.85	89.13%	122.41	273495.09	42.68	82251.17	99.96%	109.67
70	342164.38	71.31	106009.81	89.13%	142.82	319146.96	49.80	95951.09	99.96%	127.95
80	391000.55	81.46	121154.61	89.13%	163.21	364684.91	56.89	109660.79	99.97%	146.21
90	439808.20	91.63	136299.44	89.13%	183.61	410219.10	64.02	123366.03	99.95%	164.49
100	488740.58	101.82	151438.95	89.12%	204.02	455808.32	71.11	137057.66	99.95%	182.76
avg	268807.18	56.01	83296.18	89.14%	112.21	250742.99	39.14	75394.65	99.96%	100.53

Table 6: Comparison of Independent Shipment with Shipment Consolidation for Different λ Values

	Independent Ordering					Shipment Consolidation				
λ	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
$U[1000, 1500]$	306484.16	75.12	88763.04	82.63%	120.14	277078.89	45.38	73471.77	99.96%	97.96
$U[1100, 1600]$	319251.94	75.23	93499.06	84.55%	126.44	290839.46	47.18	79297.68	99.97%	105.73
$U[1200, 1700]$	331824.08	75.39	98340.26	86.25%	132.84	304403.91	48.90	85187.60	99.96%	113.57
$U[1300, 1800]$	344089.41	75.61	103216.51	87.78%	139.31	317687.24	50.57	91081.08	99.97%	121.44
$U[1400, 1900]$	356169.82	75.87	108135.59	89.09%	145.82	330740.74	52.18	96918.93	99.98%	129.23
$U[1500, 2000]$	368126.41	76.13	113162.69	90.26%	152.47	343658.18	53.74	102817.13	99.98%	137.09
$U[1600, 2100]$	379783.32	76.43	118224.80	91.31%	159.17	356262.80	55.28	108709.53	99.97%	144.95
$U[1700, 2200]$	391175.26	76.74	123238.89	92.26%	165.81	368582.83	56.71	114518.64	99.96%	152.70
$U[1800, 2300]$	402741.89	77.09	128437.05	93.10%	172.68	381018.44	58.21	120448.72	99.94%	160.61
$U[1900, 2400]$	414076.77	77.47	133655.17	93.84%	179.58	393170.42	59.56	126339.12	99.93%	168.46
$U[2000, 2500]$	425184.65	77.91	138859.20	94.49%	186.47	405073.83	60.93	132171.57	99.93%	176.24
avg	367173.43	76.27	113412.02	89.60%	152.79	342592.43	53.51	102814.71	99.96%	137.09

Table 7: Comparison of Independent Shipment with Shipment Consolidation for Different $\boldsymbol{h}$ Values

	Independent Ordering					Shipment Consolidation				
$\boldsymbol{h}$	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
$U[2,4]$	299282.49	79.11	104829.03	97.08%	140.29	290299.83	66.13	102862.89	99.99%	137.15
$U[2.5,4.5]$	317687.71	76.97	106191.68	95.42%	142.36	305541.99	61.04	102841.21	99.97%	137.11
$U[3,5]$	335352.05	75.78	107921.21	93.52%	144.93	319653.80	57.02	102808.27	99.97%	137.05
$U[3.5,5.5]$	352502.00	75.21	110011.26	91.49%	148.02	332922.29	53.75	102815.63	99.94%	137.10
$U[4,6]$	369221.93	75.02	112382.99	89.34%	151.46	345507.50	50.91	102802.01	99.96%	137.08
$U[4.5,6.5]$	385424.53	75.00	114983.17	87.19%	155.18	357418.80	48.53	102802.69	99.97%	137.08
$U[5,7]$	401174.24	75.00	117727.88	85.08%	159.07	368805.60	46.46	102815.52	99.96%	137.08
$U[5.5,7.5]$	416368.04	75.00	120609.63	82.96%	163.18	379530.02	44.58	102778.16	99.96%	137.04
$U[6,8]$	431307.50	75.00	123563.00	80.97%	167.33	390058.78	42.96	102795.15	99.93%	137.06
avg	367591.17	75.79	113135.54	89.23%	152.43	343304.29	52.38	102813.50	99.96%	137.08

Table 8: Comparison of Independent Shipment with Shipment Consolidation for Different $\boldsymbol{a}$ Values

	Independent Ordering					Shipment Consolidation				
$\boldsymbol{a}$	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
U[250,500]	328532.55	75.01	118725.83	85.09%	160.71	290746.59	41.88	102845.47	100.00%	137.13
U[300,550]	336487.27	75.04	117294.83	86.13%	158.54	302854.58	44.59	102831.01	100.00%	137.10
U[350,600]	344246.83	75.12	115941.99	87.09%	156.52	314173.71	47.15	102778.30	100.00%	137.03
U[400,650]	352127.08	75.28	114817.29	88.00%	154.84	325102.97	49.57	102814.76	100.00%	137.09
U[450,700]	359756.20	75.52	113771.24	88.82%	153.28	335409.45	51.89	102808.38	100.00%	137.09
U[500,750]	367398.34	75.83	112828.84	89.61%	151.86	345385.72	54.12	102830.52	100.00%	137.10
U[550,800]	374888.62	76.24	111973.37	90.32%	150.58	354890.13	56.26	102828.81	99.98%	137.10
U[600,850]	382060.17	76.68	111163.46	90.98%	149.39	363846.64	58.35	102801.82	99.97%	137.07
U[650,900]	389539.54	77.14	110473.42	91.59%	148.37	372829.80	60.30	102795.86	99.94%	137.07
U[700,950]	396862.70	77.66	109831.01	92.18%	147.42	381522.49	62.19	102806.04	99.93%	137.09
U[750,1000]	403767.39	78.29	109227.82	92.70%	146.51	389610.06	64.07	102791.91	99.86%	137.05
avg	366878.79	76.16	113277.19	89.32%	152.55	343306.56	53.67	102812.08	99.97%	137.08

Table 9: Comparison of Independent Shipment with Shipment Consolidation for Different P Values

	Independent Ordering					Shipment Consolidation				
P	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
750	379867.41	79.06	132698.20	98.45%	177.43	370180.51	68.12	131279.77	99.97%	175.04
800	374196.05	77.40	125463.25	97.40%	168.03	362053.60	63.86	123039.48	99.98%	164.06
850	369733.80	76.34	119555.08	96.06%	160.40	354773.70	60.15	115815.56	99.98%	154.42
900	366323.31	75.68	114698.79	94.42%	154.20	348342.57	56.78	109340.42	99.94%	145.80
950	363903.32	75.30	110898.47	92.50%	149.38	342696.81	53.84	103604.76	99.95%	138.14
1000	361985.92	75.09	107838.56	90.41%	145.52	337558.83	51.16	98424.04	99.94%	131.23
1050	360541.19	75.02	105460.00	88.15%	142.55	332875.56	48.68	93736.04	99.95%	124.98
1100	359605.34	75.00	103630.06	85.77%	140.30	328697.69	46.42	89480.85	99.94%	119.32
1150	358833.80	75.00	102179.56	83.37%	138.49	324852.73	44.41	85588.31	99.95%	114.12
1200	358231.17	75.00	101083.45	80.94%	137.15	321261.59	42.52	82025.84	99.96%	109.37
1250	357692.46	75.00	100214.83	78.56%	136.06	317874.60	40.82	78739.31	99.97%	104.98
avg	364628.53	75.81	111247.30	89.64%	149.96	340106.20	52.43	101006.76	99.96%	134.68

Table 10: Comparison of Independent Shipment with Shipment Consolidation for Different R Values

	Independent Ordering					Shipment Consolidation				
R	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density	Total Cost	Truck Number	Truck Cost	Truck Utilization	Truck Density
500	328076.19	76.33	80304.78	85.12%	160.61	307564.26	53.44	68530.96	99.86%	137.06
550	336380.38	76.28	87221.74	86.08%	158.58	314688.97	53.29	75412.62	99.95%	137.11
600	343980.50	76.45	93989.21	87.00%	156.65	321085.13	53.19	82193.22	99.96%	136.99
650	351807.40	76.41	100689.71	88.00%	154.91	328113.49	53.28	89158.91	99.98%	137.17
700	359382.28	76.25	107173.51	88.91%	153.11	335065.26	53.35	95938.25	99.98%	137.05
750	367510.01	76.35	113953.56	89.64%	151.94	342338.94	53.37	102902.48	100.00%	137.20
800	374691.31	76.34	120401.62	90.34%	150.50	348768.73	53.27	109638.26	100.00%	137.05
850	382460.53	76.40	127155.49	90.99%	149.59	355953.83	53.47	116681.59	100.00%	137.27
900	389297.82	76.33	133482.68	91.60%	148.31	362182.77	53.28	123326.40	100.00%	137.03
950	397342.00	76.38	140201.58	92.18%	147.58	369585.98	53.37	130373.98	100.00%	137.24
1000	404740.03	76.45	146619.12	92.69%	146.62	376480.94	53.37	137100.98	100.00%	137.10
avg	366878.95	76.36	113744.82	89.32%	152.58	341984.39	53.33	102841.61	99.98%	137.12