Distribution Planning with Consolidation - A Two-Stage Stochastic Programming Approach

by

Aliaa Alnaggar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The distribution planning problem with consolidation center(s) addresses the coordination of distribution activities between a set of suppliers and a set of customers, through the use of intermediate facilities in order to achieve savings in transportation cost. We study the problem from the perspective of a third-party logistics provider (3PL) that is coordinating shipments between suppliers and customers. Given customer demand of products from different suppliers, the goal is to consolidate the shipments in fewer high volume loads, from suppliers to the consolidation center(s) and from the consolidation center(s) to customer. We assume that suppliers have a finite set of transportation options, each with a given capacity and time of arrival at the consolidation center(s). Similarly, customers have a set of transportation options, each with a given capacity and dispatch time from the consolidation center(s). The 3PL wants to determine the optimal transportation options, or shipment schedule, and the allocation of shipments to transportation options from suppliers to consolidation center(s), and from consolidation center(s) to customers, that minimize the total transportation cost and holding cost at the consolidation center.

The literature studies many variations of this problem, which assume deterministic demand. This thesis extends the problem for stochastic demand and formulates it as a two-stage stochastic programming model. We model the case where the choice of transportation options is a *contractual* decision, and a 3PL needs to decide on which options to reserve for a given planning period subject to stochastic customer demand. Therefore, the choices of transportation options are the stage one variables in the two-stage stochastic program. The second stage variables, which are decisions that are made after the uncertainty conditions become known, represent the allocation of orders to reserved transportation options as well as shipping orders through a spot-market carrier, at a greater transportation cost. Because of the high computational demand of the model, the integer L-shaped method is applied to decompose the problem. To increase the efficiency of the algorithm, we experiment with three valid cuts with the goal of generating stronger cuts than the L-cut. We also apply three algorithm enhancement techniques to speed up the convergence of the algorithm. Numerical results show that the performance of our proposed methodology and valid cuts is comparable to that of CPLEX. We suggest promising areas for future work to further improve the computational efficiency of our decomposition algorithm.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Jim Bookbinder and Prof. Fatma Gzara for the great learning experience they have provided me. I wholeheartedly appreciate the support, time and feedback you have given me throughout my program. Prof. Bookbinder, I appreciate the endless encouragement you have given me and that you believed in me and my potential even when I doubted myself. I enjoyed learning from your thorough industrial and practical experience and your deep-rooted academic success. Prof. Gzara, I appreciate how you pushed me throughout my thesis to improve my work and get the most out of it in terms of learning experience and results. I enjoyed learning from your deep theoretical background; your passion and energy for optimization are contagious!

You have both given me an incredible learning experience and I am forever grateful to you. "A mind that is stretched by new experience can never go back to its old dimensions" (Oliver Wendell Holmes, Jr.). Thank you for the memorable mind-stretching intellectual experience that you have guided me through during my MASc program.

I would also like to thank the members of the reading committee, Prof. Stan Dimitrov and Prof. Bissan Ghaddar for their helpful feedback and comments on my thesis.

And finally, I would like to thank my friends in the Management Sciences department, who gave me a home away from home. Cynthia, Mobina, Remziye, Hanan, and Sayeda, you have made my journey an enjoyable one and were always there for me when I needed help. I will always remember and cherish the memories we have created together.

Dedication

To my amazing parents, who taught me to dream big, and who shower me with their love and prayers from miles and miles away.

To my loving and supportive husband, who believes in me and cheers me on every step of the way.

To Sami and Jenna, my two wonderful young children, who inspire me to stay curious and put a smile on my face every single day.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The distribution planning problem with consolidation center(s) addresses the coordination of distribution activities between a set of suppliers and a set of customers, through the use of intermediate facilities or consolidation centers, that consolidate customer demand, in order to achieve savings in transportation cost. We study the problem from the perspective of a third-party logistics provider (3PL) that is coordinating shipments between suppliers and customers. Given customer demand of products from different suppliers, the goal is to consolidate the shipments in fewer high volume loads, from suppliers to the consolidation center(s) and from the consolidation center(s) to customer. We assume that there is a single consolidation center in the network. Additionally, suppliers have a finite set of transportation options, each with a given arrival time at the consolidation center and capacity. Similarly, customers have a set of transportation options, each with a given capacity and dispatch time from the consolidation center. The 3PL wants to determine the optimal transportation options, or shipment schedule, and the allocation of shipments to transportation options from suppliers to the consolidation center, and from consolidation center to customers, that minimize the total transportation cost and holding cost at the consolidation center.

The literature studies many variations of this problem, which assume deterministic demand. Since the solution of the problem with deterministic demand might change drastically if demand changes, and in many real life problems, there could be fluctuations in demand, this thesis extends the problem for stochastic demand. We formulate the model as a two-stage stochastic programming model, where the uncertain customer demand is modeled as a finite number of possible realizations, or scenarios, each with a given probability of occurrence. We model the case where the choice of transportation options is a contractual decision and a 3PL needs to decide on which options to reserve for a given planning period subject to stochastic customer demand. Therefore, the choice of transportation options are the stage one variables in the two-stage stochastic program. The second stage variables, which are decisions that are made after the uncertainty conditions become known, represent the allocation of orders to reserved transportation options as well as shipping orders through a spot-market carrier, at a higher transportation cost.

In order to generate the scenario tree of our problem, we apply some scenario generation methods, then test the model to ensure *in-sample* and *out-sample* stability. Our problem is proven to have *in-sample* and *out-sample* when the number of scenarios equals 50. This high number of scenarios results in a large scale optimization model, with high computational demand. We apply the integer L-shaped method to decompose the problem. To increase the efficiency of the algorithm, we experiment with three valid cuts with the goal of generating stronger cuts than the L-cut. We also apply three algorithm enhancement techniques to speed up the convergence of the algorithm. We test and compare the performance of our proposed algorithm with valid cuts with that of CPLEX and the L-shaped cut. We analyze the effects of the proposed valid cuts and propose some promising future research direction.

The thesis is organized as follows. Chapter 2 reviews the literature of network distribution planning with consolidation. In Chapter 3, we describe and formulate the deterministic case of the problem under study, as well as summarize the results of some numerical experiments for the deterministic case. We formulate the problem as a two-stage stochastic program in Chapter 4. In Chapter 5, we apply the L-shaped method to the stochastic model, explain the algorithm enhancement techniques that we employ in our implementation, and propose three valid cuts. Numerical testing results are given in Chapter 6. Finally, the thesis is concluded in Chapter 7.

Chapter 2

Literature Review

The literature of network distribution planning is rich, with many different types of optimization models that aim to minimize the transportation costs of a distribution network. That is because transportation costs account for a high percentage of the cost of final products; Musa et al. (2010) argues that distribution related costs account for 30% of the price of an item. In this chapter, we first make some remarks on the different types of intermediate facilities and their distinguishing features in Section (2.1). We then review the literature of similar deterministic distribution problems in Section (2.2), and stochastic problems in Section (2.3). We also give a background on scenario generation in stochastic programs in Section (2.4).

2.1 Types of Intermediate Facilities or Consolidation Centers

The literature of distribution planning with consolidation considers many different types of intermediate facilities. Those intermediate facilities are similar in that they all perform some consolidation/deconsolidation activities. The main difference arises from the other functions that these intermediate facilities carry out or their limitations. Guastaroba et al. (2016) review the literature on freight transportation planning with intermediate facilities and explain the difference between the types of intermediate facilities. For instance, in *hub-and-spoke* networks, hubs act like intermediate facilities that perform consolidation, deconsolidation, and sorting activities in order to achieve economies of scale in transportation. However, the main distinguishing factor of the hub location problem is that locating hubs is usually one of the decisions of the network. This makes the decision level of the problem strategic rather than tactical, unlike most similar problems with different types of intermediate facilities.

Cross-docks (CDs), on the other hand, are intermediate facilities that store no inventory. They, therefore, require a significant amount of coordination and information sharing between inbound and outbound shipments in order to minimize the time that the shipments spend at the cross-dock. This type of intermediate facilities is most appropriate for the transportation of perishable products such as fresh or frozen foods and pharmaceutical drugs. Similar to CDs, *in-transit merge centers* carry no inventory. However, they differ from DCs in that shipments to the same destination are not consolidated until all components belonging to that destination arrive at the in-transit merge center.

Finally, *distribution centers (DCs)* allow for the storing of inventory for the period of time between inbound and outbound shipments. They are usually more appropriate when large consolidated shipments are sent from suppliers to DCs, and then DCs later send smaller loads to customers. In this case, economies of scale are achieved mainly when consolidating inbound shipments.

2.2 Distribution Coordination Problem

Song et al. (2008) studied the problem of coordinating distribution between a set of suppliers and a set of customers with a consolidation center operated by a third party logistics provider (3PL) in a global distribution network. Each customer makes an order that is composed of several parts from different suppliers. A supplier is assumed to provide only one unique product. Each supplier sends the product orders of all customers as one consolidated shipment to the consolidation center with one release time. At the consolidation center, the order of each customer from different suppliers is consolidated into a single shipment and delivered to the customer. The consolidation of shipments can be as early as possible or as late as possible depending on the customer requirements and the cost structure. The 3PL needs to decide (a) the pickup time from each supplier, (b) the delivery time for each customer, and (c) the transportation options for each shipment. The problem is modeled as a non-linear optimization model, then a linearization of the model is proposed. An exact algorithm to solve a special case of the problem, the All-Pair-Ordering problem, where all customers order from all suppliers, is introduced in the paper. A Lagrangean-Relaxation based heuristic is then proposed to solve the general problem. The heuristic is shown to be within a reasonable gap from the optimal solution.

Croxton et al. (2003) studied the problem of distrubtion coordination with merge-in-transit centers. The distribution network studied, which is a proprietary distribution network, consists of multiple customers, a number of suppliers and several merge-in-transit centers. Each customer makes an order which consists of parts from several suppliers. Suppliers send shipments to one of the merge-in-transit centers, which acts as a cross dock, i.e. does not store any products. At the merge-in-transit center, customer orders that arrive from different suppliers are assembled into one product and are shipped to the customer via one of the four different transportation modes considered. The model aims to minimize the total transportation and inventory costs and determine the transportation mode of each shipment, while considering a non-linear cost structure that permits economies of scale. The model also decides on the merge-in-transit center that is used for each customer order. The formulation proposed by Croxton et al. (2003) can be considered an integer multicommodity generalized network flow problem. It differs from the literature in that it considers complex cost structures in a capacitated network optimization problem, while most papers that consider complex cost structures deal with uncapacitated problems. Also, the model addresses operational decisions rather than startegic or tactical decisions.

Berman and Wang (2006) study the problem of selecting a distribution strategy for delivering a number of products from a set of suppliers to a set of plants with the objective of minimizing the total transportation, pipeline inventory and plant inventory costs. The problem is modeled as a nonlinear integer program, where the nonlinearity comes from the objective function that is neither convex nor concave. The model involves only a few details and makes some simplifying assumptions to make the problem manageable. For instance, the model assumes that demand is constant for each product at each plant from each supplier, shipments between suppliers and plants are either direct or go through cross-docks, and that only one truck type with a given capacity level is used. Small instances of the proposed model are solved first using commercial non-linear solvers (CONOPT, DICOPT, and MINOS). These solvers, however, only give local optimal solutions with no information on solution quality. The paper then proposes two heuristics, a greedy heuristic and a Lagrangian Relaxation (LR) heuristic, as well as a branch-and-bound (BB) algorithm to solve the problem.

Table 2.1 compares and contrasts the features of the problems under study in the three papers reviewed above. Furthermore, Guastaroba et al. (2016) review the operations research literature on freight transportation planning with intermediate facilities (such as consolidation or distribution centers or cross-docks). They argue that the literature of distribution planning with intermediate facilities has received much attention mainly because transportation costs account for a high percentage of the cost of a product. Therefore, intermediate facilities that consolidate products from the same origin or to the same destination help in achieving economies of scale in the network through better utilization of the capacity of the transportation modes used in the network. Three different classes of problems are surveyed in detail, namely, intermediate facilities in vehicle routing problems, intermediate facilities in transshipment problems, and intermediate facilities in service network design problems. The class of problems that is most relevant to our problem is the intermediate facilities in transshipment problems. The papers by Croxton et al. (2003), Berman and Wang (2006) and Song et al. (2008) all fall under the same class of problems.

2.3 Two-stage Stochastic Programming for Distribution Coordination

We study the problem of distribution coordination in a two-stage stochastic programming setting, to take into account demand stochasticity, where customer demand is assumed to be uncertain with a finite number of possible realizations, or scenarios. Two-stage stochastic mixed integer programming, which is a way of handling uncertainty, has been applied

Criteria	Song et al. (2008)	Croxton et al. (2003)	Berman and Wang (2006)
Decision level	Tactical	Tactical	Tactical
Planning horizon	Project-based	Few days (1 day time period)	Single time period
No. of consolidation centers	Single	Multiple	Multiple
Capacity at consolidation centers	Uncapacitated	Capacitated	Uncapacitated
Inventory at consolidation centers	Can keep inventory	No inventory	No inventory
Inventory at customer locations	Not considered	Considered	Considered
Pipeline inventory	Not considered	Not considered	considered
Consolidation at suppliers	Consolidate customer orders	No consolidation	Consolidate products
Cost structure	Neither convex nor concave	Non-convex piecewise linear	Neither convex nor concave
Demand	One unit from each supplier	Demand varies	Constant demand
Number of commodities	single-commodity	multi-commodity	multi-commodity
Discrete set of transportation options	Yes	No	No

Table 2.1: Comparison of Features of Similar Problems in the Literature

to many problems, including distribution network problems. The problem categorizes the decisions into two stages, where the first stage handles strategic decisions, that are made before the uncertainty conditions become known, and the second stage, which is modeled by a finite number of scenarios each with an associated probability of occurrence, tackles tactical or operational decisions after the uncertainty is eliminated.

Kılıç and Tuzkaya (2015) use two-stage stochastic mixed integer programming to model a physical distribution network design problem, where the first stage deals with strategic decisions, namely location selection of distribution centers, and the second-stage addresses transportation and inventory decisions, as well as unmet demand. In contrast to our work, the paper considers a one-echelon problem, namely the flow between distribution centers and wholesalers, with no consideration of flow between suppliers and distribution centers.

2.4 Scenario Generation for Stochastic Programming

In stochastic programming, some of the model parameters are uncertain, and therefore, are described by distributions rather than single values. Most stochastic programs, with the exception of some trivial cases, cannot be solved using continuous distributions; instead, the continuous distributions need to be approximated to discrete ones with a preset number of outcomes. The resulting discretization is referred to as a *scenario tree* (Kaut, 2012).

Kaut and Wallace (2003a) evaluate different scenario-generation methods and propose minimal requirements that a certain scenario-generation method must satisfy in order to be used for solving a stochastic programming model. The authors argue that there is no single method that is suitable for all models, and that the choice of the scenario-generation method has to be linked to the specific model under study. In addition, the scenario-generation methods are compared based on their practical performance in real-life problems rather than their theoretical properties.

When measuring the quality of the resulting scenario tree, Kaut and Wallace (2003a) recommend judging a scenario tree not by how well it approximates the continuous probability distribution, but rather by the quality of the decisions it gives. This can be achieved by measuring the stability of the scenario tree, i.e. essentially the same optimal solution is obtained from several scenario trees that are constructed using the same input parameters in the scenario-generation method used, while keeping the rest of the parameters in the problem fixed. In addition, the scenario-generation method should not introduce any bias, i.e. the optimal solution of the stochastic programming model with the scenario tree should be approximately equal to the optimal solution of the model with the "true" distribution.

Kaut and Wallace (2003a) also provides an overview of some commonly used scenario-generation methods. These include methods like conditional sampling, sampling from specified marginals and correlations, moment matching, path-based methods, and optimal discretization. With the exception of conditional sampling, the other four methods have the advantage of being a better fit in generating scenario trees for multivariate random variables.

Kaut (2012, chapter 4) also discuss scenario generation in stochastic programming models. The author argues that having a discretization method that is a very good estimate of the continuous distribution is likely to make the problem intractable, as it would result in a high number of scenarios. In order to tackle this issue, it is important to try to understand the specific needs of the model in hand, i.e. if a certain property of the distribution does not contribute to the solution of the problem, then it is unnecessary to take this property into consideration when the scenario tree is generated. It is, however, not straightforward to know which properties are important for a certain model and which are not. For that reason, after a particular scenario generation procedure has been adopted, it it necessary to test it for stability, as discussed earlier. Note that the solution of the problem may change, which is natural for stochastic programs. However, the objective function value should be similar to a great extent to achieve *in-sample* stability. Additionally, when testing the stability of the results of the model, changing the number of scenarios in a scenario tree should not significantly affect the objective function value.

Lium et al. (2009) discuss how they generate the scenario trees of their problem of interest; Stochastic Service Network Design. They follow the procedure proposed by Høyland et al. (2003) that was initially developed by Høyland and Wallace (2001). For their model, *in-sample* stability is achieved using 50 scenarios. In order to test for *out-sample* stability, the authors sample trees with 20,000 scenarios from the same distribution and assume that those trees represent the "true" distribution.

Høyland and Wallace (2001) propose a methodology for generating scenario trees that is dependent upon the properties of the distribution that we want to take into account, the number of periods of the stochastic process, and the number of conditional outcomes in each stage. Additionally, the authors state that if the random variables are discrete and have few joint outcomes, then scenario tree generation is straightforward and can be completed manually.

Having summarized the literature of similar deterministic and stochastic problems in distribution coordination, as well as the scenario generation methods for stochastic programming, we turn in Chapter 3 to the formulation and solution of the deterministic case of our problem, before stochasticity in demand is introduced. This will provide intuition and greater understanding of the problem before we propose the two-stage stochastic programming model in Chapter 4.

Chapter 3

Deterministic Distribution Planning with Consolidation

3.1 Problem Description and Assumptions

We consider the problem of a third party logistics provider (3PL) that is coordinating distribution between a set of suppliers and a set of customers. Each customer makes an order for a product or a number of products from several suppliers. Each supplier consolidates the orders of customers into one or more consolidated shipments, then the 3PL delivers the consolidated shipments through one or more transportation options to the consolidation center. Each transportation option has a given arrival time at the consolidation center, a vehicle capacity, and can carries the demand or portions of the demand of one or more customers.

At the consolidation center, the orders of each customer from different suppliers are consolidated into as few shipments as possible, and are delivered to the customer using one or more transportation options with a given volumetric capacity and dispatch time from the consolidation center. The 3PL needs to decide (a) the transportation option(s) for each supplier and the corresponding arrival time of each chosen option, (b) the allocation of customer orders to transportation options, (c) the transportation option(s) for each customer and the corresponding dispatch time, and (d) the allocation to transportation options of customer orders of distinct products from different suppliers. The objective of the 3PL is to minimize total transportation costs plus holding costs at the consolidation center.

Problem Assumptions:

1. For each inbound shipment arrival time and for each customer with an order on that shipment, there is at least one outbound shipment that will deliver the order to the customer by the required due date.

2. Holding costs at the consolidation center do not depend on consolidation time. Therefore, consolidating a shipment as soon as all its parts from different suppliers arrive or just before delivering it to the customer would result in the same holding cost at the consolidation center.

3. Demand of different customers for different products from the same supplier can be consolidated into one or more shipments. The demand of a customer for a particular product from a given supplier, however, has to be shipped in one load through a single transportation option with a particular release time, and is therefore not divisible for different transportation options.

4. The time to consolidate customer orders at the consolidation center is negligible.

3.2 Problem Formulation

In more detail, the 3PL provides services to n customers; each customer can place an order with a given demand from a subset of m suppliers. Each supplier provides a set of unique products, i.e. each product is available from one and only one supplier. A customer can order different products from the same supplier. Shipments from suppliers can be sent on various transportation options with distinct release times. However, all orders for a certain product from a given supplier to a customer have to be shipped on the same transportation option. The 3PL needs to decide (a) the transportation option(s) and pickup time(s) for each supplier, (b) the allocation to shipments, of customer orders of products from suppliers, (c) the transportation option(s) and delivery time(s) for each customer, and (d) the allocation of products to shipments from the consolidation center to customers.

Figure 3.1 shows a diagram of the network under study, as well as the decision variables of the corresponding model.

We define the *model parameters* as follows.

Model parameters:

- I = set of all suppliers = $\{1, ..., m\}$;
- J = set of all customers = $\{1, ..., n\}$;
- $I(j)$ = set of suppliers $i \in I$ providing products for customer j;
- $J(i)$ = set of customers $j \in J$ requiring products from supplier i;
- Q_i = the number of possible arrival time slots for supplier $i, i \in I$;

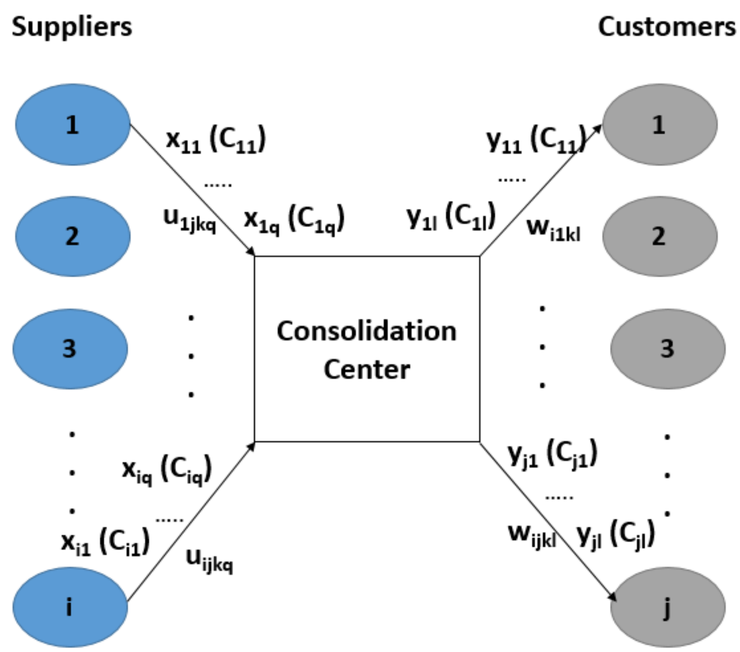

Figure 3.1: Consolidation network under study

- L_j = the number of possible departure time slots for customer $j, j \in J$;
- T_{iQ} = set of all possible shipment arrival times at the center from supplier i;
- T_{jL} = set of all possible shipment departure times from the center to customer j;
- t_{iq} = the arrival time of supplier i's shipment at time q;
- t_{jl} = the departure time of customer j's shipment at time l;
- $f_i(t_{iq})$ = transportation cost associated with receiving one vehicle at arrival time $t_{iq}, t_{iq} \in T_{iQ}$;
- $g_j(t_{jl})$ = transportation cost associated with sending one vehicle at departure time $t_{jl}, t_{jl} \in T_{jL}$;

- $K(ij)$ = the set of products k that supplier i provides for customer j;
- h_k =cost of holding one unit of product k;
- v_k = the volume of product k;
- d_{ijk} = the demand of customer j for product k provided by supplier i;
- C_{iq} = the capacity of one vehicle of the transportation option associated with the arrival time t_{iq};
- C_{jl} = the capacity of one vehicle of the transportation option associated with the arrival time t_{jl};

Decision variables:

$$x_{iq} = \begin{cases} 1, & \text{if transportation option } q \text{ for supplier } i \text{ is chosen, } i \in I, q \in Q \\ 0, & \text{otherwise} \end{cases}$$

$$y_{jl} = \begin{cases} 1, & \text{if transportation option } l \text{ for customer } j \text{ is chosen, } j \in J, l \in L \\ 0, & \text{otherwise} \end{cases}$$

$$u_{ijkq} = \begin{cases} 1, & \text{if customer } j\text{'s of product } k \text{ from supplier } i \text{ is shipped by transportation} \\ & \text{option } q, j \in J,\ i \in I,\ k \in K(ij),\ q \in Q. \\ 0, & \text{otherwise.} \end{cases}$$

$$w_{ijkl} = \begin{cases} 1, & \text{if customer } j\text{'s order of product } k \text{ from supplier } i \text{ is shipped by transportation} \\ & \text{option } l, j \in J, i \in I(j), k \in K(ij), l \in L. \\ 0, & \text{otherwise.} \end{cases}$$

Formulation:

We formulate the deterministic case of our problem as follows:

$$\begin{aligned}
\min \ F(x,y,u,w) = & \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) \\
& + \sum_{j\in J}\sum_{i\in I(j)}\sum_{k\in K(ij)} d_{ijk}v_k h_k\Big(\sum_{l\in L} t_{jl}w_{ijkl} - \sum_{q\in Q} t_{iq}u_{ijkq}\Big) && \text{(3.1a)}\\
\text{s.t.} \quad & \sum_{q\in Q} u_{ijkq} = 1, \qquad i\in I, j\in J(i), k\in K(ij) && \text{(3.1b)}\\
& \sum_{l\in L} w_{ijkl} = 1, \qquad j\in J, i\in I(j), k\in K(ij) && \text{(3.1c)}\\
& \sum_{q\in Q} t_{iq}u_{ijkq} \le \sum_{l\in L} t_{jl}w_{ijkl}, \qquad j\in J, i\in I(j), k\in K(ij) && \text{(3.1d)}\\
& \sum_{j\in J(i)}\sum_{k\in K(ij)} d_{ijk}v_k u_{ijkq} \le C_{iq}x_{iq}, \quad i\in I, q\in Q && \text{(3.1e)}\\
& \sum_{i\in I(j)}\sum_{k\in K(ij)} d_{ijk}v_k w_{ijkl} \le C_{jl}y_{jl}, \quad j\in J, l\in L && \text{(3.1f)}\\
& x_{iq}, y_{jl} \in \{0,1\}, \quad i\in I, j\in J, q\in Q, l\in L \\
& u_{ijkq} \in \{0,1\} \quad i\in I, j\in J(i), k\in K(ij), q\in Q \\
& w_{ijkl} \in \{0,1\} \quad j\in J, i\in I(j), l\in L, k\in K(ij) && \text{(3.1g)}
\end{aligned}$$

The objective function (3.1a) minimizes the total transportation cost of trips to all customers and suppliers plus inventory holding cost at the consolidation center. Constraint (3.1b) ensures that the model chooses exactly one transportation option to carry the demand of each customer for each product. Constraint (3.1c) guarantees that the model selects exactly one transportation option to ship every customer order for each product ordered. Constraint (3.1d) ensures that the shipment for each customer for each product cannot leave the consolidation center until it has arrived at the center. Constraint (3.1e) enforces that for each supplier, the total volume of customer orders that are shipped through a transportation option does not exceed the capacity of that option. Similarly, constraint (3.1f) requires that for each customer, the total volume of orders shipped through a transportation option does not exceed the capacity of that transportation option. Finally, constraint (3.1g) forces all variables to be assigned only binary values.

3.3 Data Generation

3.3.1 Cost Function

The transportation cost functions f_i and g_j in objective function (3.1a) is a linear function of the capacity of the respective option for each supplier and each customer. These functions can be represented as follows:

$$\begin{aligned} f_i(x_{iq}) &= c_i C_{iq} x_{iq} \\ g_j(y_{jl}) &= c'_j C_{jl} y_{jl} \end{aligned} \tag{3.2}$$

Where c_i and c'_j are the variable cost per unit of capacity of shipping from supplier i to the consolidation center and from the consolidation center to customer j, respectively. We set the value of c_i and c'_j to be equal to 10 in our randomly generated instances.

3.3.2 Other Parameters

For our test instances, we generate the data randomly, partly as outlined in Song et al. (2008), but with additional parameters. We create a network of m suppliers and n customers. For each supplier i, we generate a random number of transportation options $\bar{K}_i$ between (20, 30), then the set X_i that consists of $\bar{K}_i$ arrival times generated in the range (100, 500). Similarly, each customer j has a random number of transportation options $\bar{K}_j$ between (20, 30) and a set Y_j, which consists of $\bar{K}_j$ dispatch times from the consolidation center ranging between (100, 500). Each supplier i offers k_i number of unique products, where k_i is uniformly distributed between 1 and 5. The number of supplier-customer pairs in the network is specified for each instance. The capacity of each transportation option for each supplier/customer C_{iq} and C_{jl} is between (500, 1000). The demand of each product d_{ijk} is assumed to be uniform between (1, 10) and the volume of each unit of product v_k is uniform between (10,30). Finally, the holding cost h_k is set as $h_k = \alpha_i * \beta_k * h$, where α_i is the scale factor of supplier i that is supplying product k, and β_k is the scale factor of product k, both of which are randomly generated between (0.5, 1.5), and h is randomly generated in the range (1, 2).

3.4 Numerical experiments

Model (3.1) was tested on randomly generated instances of different sizes. Each instance was run for an hour, two hours, then ten hours. The results are displayed in Table (3.1), where the gap refers to the relative gap from the output of CPLEX that is defined as the gap between the best integer objective function value (incumbent) and the objective of the best node remaining in the branch-and-bound tree. S-C pairs refer to the number of supplier-customer pairs, and demand pairs refer to product demand pairs, i.e. total number of products ordered by all customers from all suppliers.

It can be seen from Table 3.1 that for most instances, no significant improvement in the solution quality in terms of optimality gap is achieved when letting the model run for longer periods of time. For example, instance 1 has a gap of 7.62% after 1 hour, and when letting the model run for 9 more hours, the gap only improves to 6.16%.

Next, we consider the stochastic case of our model in following chapter.

Instance No.	No. of Supp.	No. of Cust.	S-C Pairs	Demand Pairs	Gap (1hr)	Gap (2hr)	Gap (10hr)
1	10	10	50	103	7.62%	6.89%	6.16%
2	10	10	50	99	8.42%	7.73%	6.88%
3	10	10	50	120	6.81%	5.83%	5.21%
Avg				107	7.62%	6.82%	6.08%
4	10	20	100	188	8.48%	7.38%	5.59%
5	10	20	100	231	9.86%	8.96%	7.04%
6	10	20	100	158	6.11%	5.91%	5.34%
Avg				192	8.15%	7.42%	5.99%
7	20	20	100	184	10.13%	8.84%	7.97%
8	20	20	100	195	13.38%	11.33%	7.04%
9	20	20	100	219	13.95%	10.86%	8.14%
Avg				199	12.49%	10.34%	7.72%
10	20	20	300	564	11.09%	9.73%	8.71%
11	20	20	300	560	13.82%	10.68%	9.69%
12	20	20	300	542	11.7%	10.97%	10.21%
Avg				555	12.2%	10.46%	9.54%
13	20	50	300	509	15.17%	14.87%	12.01%
14	20	50	300	585	14.87%	12.51%	11.53%
15	20	50	300	559	12.71%	12.71%	10.64%
Avg				551	14.25%	13.36%	11.39%
16	20	50	500	1088	12.86%	12.77%	9.33%
17	20	50	500	1085	19.33%	19.09%	16.28%
18	20	50	500	1058	14.13%	14.03%	10.93%
Avg				1077	15.44%	15.30%	12.18%
19	50	50	500	1038	15.34%	15.34%	11.65%
20	50	50	500	1014	15.65%	15.62%	14.42%
21	50	50	500	1041	14.89%	14.85%	12.97%
Avg				1031	15.29%	15.27%	13.01%

Table 3.1: Deterministic Model Numerical Testing

Chapter 4

Stochastic Distribution Planning with Consolidation

4.1 Problem Description

In this section, we propose an extension to the deterministic demand case in which the customer demand is stochastic. Given a certain distribution of demand, the model needs to decide on transportation options at both echelons of the distribution network at the beginning of the planning horizon. The chosen transportation options and their associated capacities are fixed for the whole planning horizon. If the demand of a certain customer for a given scenario cannot be fulfilled with the transportation capacities allocated to that customer or the suppliers serving it, then a penalty cost is charged. This cost represents the cost of transporting the unmet demand by a common carrier from the spot market, at a set of discrete possible transportation times. On the other hand, if the total demand of a customer from different suppliers is lower than the allocated capacity of that customer, that demand can of course be fulfilled by the allocated vehicle and no penalty cost would be charged. However, in that case the utilization of the transportation vehicles will be lower.

Two-stage stochastic programming is employed to model this case. The distribution of customer demand is discretized into a finite set of possible realizations, or scenarios, each with a given probability of occurrence. The first stage is to decide on the transportation options and their corresponding capacities, for each supplier and each customer so as to minimize total transportation cost. This is considered a strategic contractual decision, since these transportation options are then fixed for the whole planning horizon. The second stage, on the other hand, decides on operational or tactical decisions that we make after the uncertainty conditions become known. For our application, second stage variables represent for each scenario, the allocation variables that allocate products to available first stage transportation options, and the recourse variables or the variables for allocating extra capacity purchased at the spot market. The two stages are optimized simultaneously in order to minimize the total expected cost for all possible scenarios (Shapiro et al., 2014).

One practical application of this case, amongst others, is that of online shopping or e-retailing. An online retailer can sell a set of products from different suppliers, with no warehouse to keep inventory of those products. A customer, or a customer zone, can place an order for different products from distinct suppliers. Assuming the retailer is outsourcing its logistical needs to a 3PL, the 3PL would arrange the delivery of consolidated customer orders from suppliers to a consolidation center by carriers with which it has set a long term commitment or contract for a pre-specified capacity and period of time. Similarly, the 3PL consolidates customer orders from different suppliers and ships them from the consolidation center to the customer in fewer loads, again through a contracted common carrier. And since demand is uncertain, the 3PL needs to decide which transportation options to reserve for a given period of time in order to minimize expected inventory holding cost at the consolidation center and the expected penalty cost of purchasing extra capacity from the spot market.

4.2 Problem Formulation

We propose two formulations of this case that have different types of recourse variables. In the first formulation, in section (4.2.1), the recourse variables are assumed to be continuous, while in the second, in section (4.2.2) they are assumed to be binary.

Figure 4.1 shows a diagram of the network, as well as the decision variables of the proposed stochastic programming models.

4.2.1 Formulation 1

First, define the following additional parameters:

- S is the set of possible scenarios within the planning horizon $\{1, \ldots, s\}$.
- p^s is the probability of the realization of a scenario $s \in S$.
- π is the penalty cost per unit of transportation through a spot market carrier.
- d^s_{ijk} is the demand of customer j for product k that is provided by supplier i under scenario s.

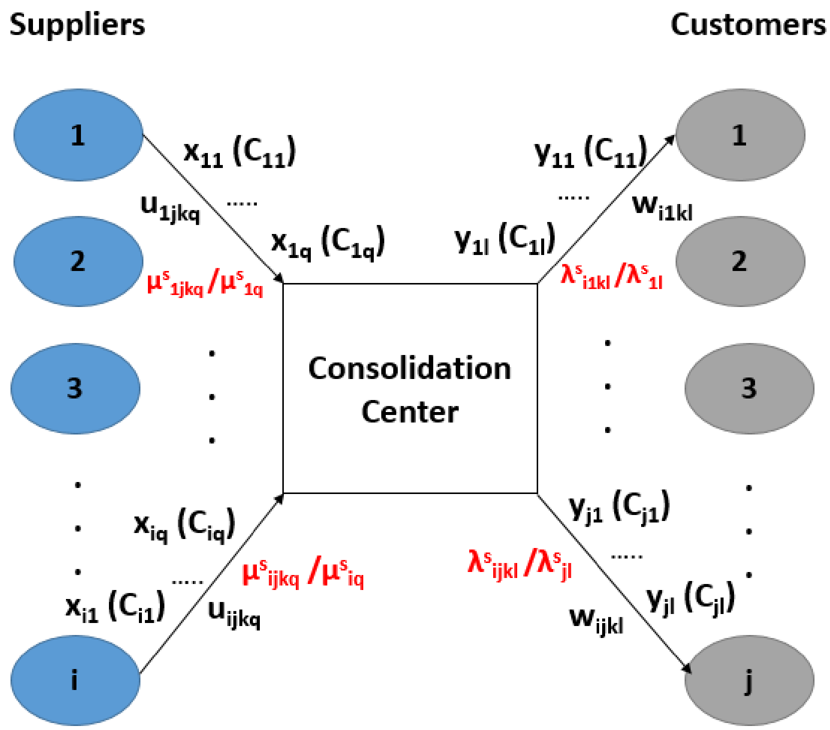

Figure 4.1: Consolidation Network in a Stochastic Programming Setting

In addition, define the following additional decision variables:

$$u^s_{ijkq} = \begin{cases} 1, & \text{if order of customer } j \text{ from supplier } i \text{ for product } k \text{ in scenario } s \text{ is shipped by} \\ & \text{transportation option } q, i \in I, j \in J(i), k \in K(ij), q \in Q, s \in S. \\ 0, & \text{otherwise.} \end{cases}$$

$$w^s_{ijkl} = \begin{cases} 1, & \text{if customer } j\text{'s order of product } k \text{ from supplier } i \text{ in scenario } s \text{ is shipped through} \\ & \text{transportation option } l, j \in J, i \in I(j), k \in K(ij), l \in L, s \in S. \\ 0, & \text{otherwise.} \end{cases}$$

$\mu_{iq}^s =$ the excess capacity needed to satisfy demand of products shipped from supplier i to the consolidation center at the arrival time of transportation option q for a given scenario $s \in S$.

$\lambda_{jl}^s =$ the excess capacity needed to satisfy demand of products shipped from the consolidation center to customer j, at the departure time of transportation option l for a given scenario $s \in S$.

The problem can then be formulated as follows:

$$[4.1] \ \min \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) + \sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)}\sum_{s\in S} p^s[d_{ijk}^s v_k h_k(\sum_{l\in L} t_{jl} w_{ijkl}^s - \sum_{q\in Q} t_{iq} u_{ijkq}^s)] + \sum_{i\in I}\sum_{s\in S}\sum_{q\in Q} p^s \pi(\mu_{iq}^s) + \sum_{j\in J}\sum_{s\in S}\sum_{l\in L} p^s \pi(\lambda_{jl}^s) \quad (4.1a)$$

$$\text{s.t.} \quad \sum_{q\in Q} u_{ijkq}^s = 1, \qquad i\in I, j\in J(i), k\in K(ij), s\in S \quad (4.1b)$$

$$\sum_{l\in L} w_{ijkl}^s = 1, \qquad j\in J, i\in I(j), k\in K(ij), s\in S \quad (4.1c)$$

$$\sum_{q\in Q} t_{iq} u_{ijkq}^s \le \sum_{l\in L} t_{jl} w_{ijkl}^s, \qquad j\in J, i\in I(j), k\in K(ij) \quad (4.1d)$$

$$\sum_{j\in J(i)}\sum_{k\in K(ij)} d_{ijk}^s v_k u_{ijkq}^s \le C_{iq} x_{iq} + \mu_{iq}^s, \qquad i\in I, q\in Q, s\in S \quad (4.1e)$$

$$\sum_{i\in I(j)}\sum_{k\in K(ij)} d_{ijk}^s v_k w_{ijkl}^s \le C_{jl} y_{jl} + \lambda_{jl}^s, \qquad j\in J, l\in L, s\in S \quad (4.1f)$$

$$x_{iq}, y_{jl} \in \{0,1\}, \qquad i\in I, q\in Q, j\in J, l\in L$$

$$u_{ijkq}^s \in \{0,1\}, \qquad i\in I, q\in Q, j\in J(i), k\in K(ij), s\in S$$

$$w_{ijkl}^s \in \{0,1\}, \qquad j\in J, l\in L, k\in K(ij), s\in S$$

$$\mu_{iq}^s \ge 0, \qquad i\in I, q\in Q, s\in S$$

$$\lambda_{jl}^s \ge 0 \qquad j\in J, l\in L, s\in S \quad (4.1g)$$

x_{iq} and y_{jl} represent transportation options for suppliers and customers, respectively, that run daily, which the 3PL has reserved with a common carrier for the whole planning horizon. d^s_{ijk} represents a finite set of possible realizations of demand for the given planning horizon. If the total demand of customer orders from a single supplier exceeds the capacity of the reserved transportation options allocated to that supplier, then the 3PL sends the remaining orders μ^s_{iq} through a spot market carrier at a charge of π per volumetric unit shipped at the same time as transportation option q. Similarly, if the total demand of customer orders from all suppliers exceeds the capacity of the reserved transportation options allocated to that customer, the 3PL sends the remaining orders λ^s_{jl} through a spot market carrier at a charge of π per volumetric unit shipped at the the same time as as the chosen transportation option l.

4.2.2 Formulation 2

An alternative way of formulating the stochastic case is to consider a different type of recourse variables than those proposed in Model (4.1). The new proposed recourse variables do not allow for splitting of demand between stage one transportation options x_{iq} and y_{jl} and recourse variables μ_{iq} and λ_{jl}. Alternatively, if for a given scenario, extra capacity from the spot market is needed, the whole demand of a given customer for a specific product is shipped through that extra capacity. This is a more realistic assumption, since you cannot divide a single package or product over two vehicles. Also, this assumption allows for savings in holding cost at the consolidation center to be achieved, since if some products are shipped through extra capacity, the model will choose the time of shipment that minimizes the holding cost at the consolidation center.

First, define the following new decision variables:

$$\mu^s_{ijkq} = \begin{cases} 1, & \text{if customer } j\text{'s order of product } k \text{ from supplier } i \text{ in scenario } s \text{ is shipped by} \\ & \text{a spot market carrier, with arrival time } q,\ i \in I, j \in J(i), k \in K(ij), q \in Q, s \in S. \\ 0, & \text{otherwise.} \end{cases}$$

$$\lambda^s_{ijkl} = \begin{cases} 1, & \text{if customer } j\text{'s demand of product } k \text{ from supplier } i \text{ in scenario } s \text{ is shipped by} \\ & \text{a spot market carrier, with dispatch time } l,\ j \in J, i \in I(j), k \in K(ij), l \in L, s \in S. \\ 0, & \text{otherwise.} \end{cases}$$

Then the two-stage stochastic programming model can be formulated as in Model 4.2.

Shipments are sent through a spot market carrier ($\mu^s_{ijkq}, \lambda^s_{ijkl}$), in Model 4.2, if (1) the total demand of a given supplier or customer exceeds the capacity of the reserved transportation option(s) of that supplier or customer, or (2) the savings in holding cost at the consolidation center when a shipment is sent at a different time exceed the cost of purchasing extra capacity at the spot market.

For the rest of the thesis, we will carry on with the solution methodology and numerical testing for Model 4.2. Some numerical testing for Model 4.1 is shown in Appendix A. In the following chapter, we outline our proposed solution methodology.

$$
\begin{aligned}
[4.2]\ \min\ & \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) \\
& + \sum_{s\in S}\sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)} p^s[d^s_{ijk} v_k h_k(\sum_{l\in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}) - \sum_{q\in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq}))] \\
& + \sum_{s\in S}\sum_{i\in I}\sum_{q\in Q}\sum_{j\in J(i)}\sum_{k\in K(ij)} p^s \pi d^s_{ijk} v_k(\mu^s_{ijkq}) + \sum_{s\in S}\sum_{j\in J}\sum_{l\in L}\sum_{i\in I(j)}\sum_{k\in K(ij)} p^s \pi d^s_{ijk} v_k(\lambda^s_{ijkl})
\end{aligned} \tag{4.2a}
$$

$$
\text{s.t.} \quad \sum_{q\in Q}(u^s_{ijkq} + \mu^s_{ijkq}) = 1, \qquad i\in I, j\in J(i), k\in K(ij), s\in S \tag{4.2b}
$$

$$
\sum_{l\in L}(w^s_{ijkl} + \lambda^s_{ijkl}) = 1, \qquad j\in J, i\in I(j), k\in K(ij), s\in S \tag{4.2c}
$$

$$
\sum_{q\in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq}) \le \sum_{l\in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}), \qquad j\in J, i\in I(j), k\in K(ij) \tag{4.2d}
$$

$$
\sum_{j\in J(i)}\sum_{k\in K(ij)} d^s_{ijk} v_k u^s_{ijkq} \le C_{iq} x_{iq}, \qquad i\in I, q\in Q, s\in S \tag{4.2e}
$$

$$
\sum_{i\in I(j)}\sum_{k\in K(ij)} d^s_{ijk} v_k w^s_{ijkl} \le C_{jl} y_{jl}, \qquad j\in J, l\in L, s\in S \tag{4.2f}
$$

$$
\begin{aligned}
& x_{iq}, y_{jl} \in \{0,1\}, && i\in I, q\in Q, j\in J, l\in L \\
& u^s_{ijkq}, \mu^s_{ijkq} \in \{0,1\}, && i\in I, q\in Q, j\in J(i), k\in K(ij), s\in S \\
& w^s_{ijkl}, \lambda^s_{ijkl} \in \{0,1\}, && j\in J, i\in I(j), l\in L, k\in K(ij), s\in S
\end{aligned} \tag{4.2g}
$$

Chapter 5

Solution Methodology

5.1 Benders Decomposition and The L-shaped Method

The special structure of the two-stage stochastic programming model (4.2) makes it a good fit for applying Benders decomposition (BD) (Benders, 1962), which is also referred to as the "L-shaped method" in the context of stochastic programming (Van Slyke and Wets, 1969). Classical BD or the linear L-shaped method assume that the subproblem(s) of the decomposition are linear program(s). Since all the decision variables in Model 4.2 are binary, the integer L-shaped method applies (Laporte and Louveaux, 1993). For the rest of the thesis, we will refer to the linear L-shaped method as Benders decomposition (BD).

When applying the integer L-shaped method on Model 4.2, the problem decomposes into a relaxed Master Problem (MP) with first stage variables x_{iq} and y_{jl}, and a set of $|S|$ disaggregated subproblems (SP^s), one for each scenario $s \in S$, with the second stage variables $u^s_{ijkq}, w^s_{ijkl}, \mu^s_{ijkq}$, and λ^s_{ijkl}. The algorithm iteratively solve MP and SPs until it converges to an optimal solution. BD results in the same MP and SPs, except that the

SPs are relaxed to be linear programs instead of binary integer programs.

The resulting decomposition for BD and the integer L-shaped method is outlined in the following sections.

5.1.1 Benders Decomposition

To decompose Model 4.2, we can reformulate it as shown in Model 5.1, where z^s_{LP} is the objective function value of the relaxed subproblem of scenario $s \in S$.

$$\min_{x,y} \quad \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) + \begin{cases} \min \sum_{s\in S} p^s z^s_{LP} \\ \text{s.t. } [4.2b] - [4.2d] \end{cases} \tag{5.1}$$
$$\text{s.t.} \quad x_{iq}, y_{jl} \in \{0,1\}, \quad i \in I, q \in Q, j \in J, l \in L$$

For a fixed value of x_{iq} and y_{jl} variables, which we denote as $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$, respectively, an inner relaxed subproblem (SP^{LP}_s), $\forall s \in S$, can be expressed as:.

$$[5.2] \; z^s_{LP} = \min \sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)} d^s_{ijk} v_k h_k[\sum_{l\in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}) - \sum_{q\in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq})]$$
$$+ \sum_{i\in I}\sum_{q\in Q}\sum_{j\in J(i)}\sum_{k\in K(ij)} \pi d^s_{ijk} v_k(\mu^s_{ijkq}) + \sum_{j\in J}\sum_{l\in L}\sum_{i\in I(j)}\sum_{k\in K(ij)} \pi d^s_{ijk} v_k(\lambda^s_{ijkl}) \tag{5.2a}$$

s.t. [4.2b] - [4.2d]

$$\sum_{j\in J(i)}\sum_{k\in K(ij)} d^s_{ijk} v_k u^s_{ijkq} \le C_{iq}\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \qquad i \in I, q \in Q \tag{5.2b}$$

$$\sum_{i\in I(j)}\sum_{k\in K(ij)} d^s_{ijk} v_k w^s_{ijkl} \le C_{jl}\bar{\boldsymbol{y}}_{\boldsymbol{jl}}, \qquad j \in J, l \in L \tag{5.2c}$$

$$u^s_{ijkq}, w^s_{ijkl}, \mu^s_{ijkq}, \lambda^s_{ijkl} \le 1, \qquad i \in I, j \in J, k \in K(ij), q \in Q, l \in L \tag{5.2d}$$

$$u^s_{ijkq}, w^s_{ijkl}, \mu^s_{ijkq}, \lambda^s_{ijkl} \ge 0, \qquad i \in I, j \in J, k \in K(ij), q \in Q, l \in L \tag{5.2e}$$

Since SP^{LP}_s is a linear program, it can be replaced by its dual, shown in Model 5.3, where $\alpha^s_{ijk}, \beta^s_{ijk}$, and γ^s_{ijk} are the dual variables of constraints (4.2b) - (4.2d), ϵ^s_{iq} and ϕ^s_{jl} are the dual variables of constraints (5.2b) and (5.2c), and $u^{'s}_{ijkq}, w^{'s}_{ijkl}, \mu^{'s}_{ijkq}, \lambda^{'s}_{ijkl}$ are the dual variables of constraint (5.2d).

$$[5.3]\ z^s_{dual} = \max \sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)}(\alpha^s_{ijk} + \beta^s_{ijk}) + \sum_{i\in I}\sum_{q\in Q}(C_{iq}\bar{\boldsymbol{x}}_{\boldsymbol{iq}})\epsilon^s_{iq} + \sum_{j\in J}\sum_{l\in L}(C_{jl}\bar{\boldsymbol{y}}_{\boldsymbol{jl}})\phi^s_{jl}$$
$$+\sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)}\sum_{q\in Q}\sum_{l\in L}(u^{'s}_{ijkq} + w^{'s}_{ijkl} + \mu^{'s}_{ijkq} + \lambda^{'s}_{ijkl}) \tag{5.3a}$$

$$\text{s.t.}\quad \alpha^s_{ijk} + t_{iq}\gamma^s_{ijk} + d^s_{ijk}v_k\epsilon^s_{iq} + u^{'s}_{ijkq} \le -d^s_{ijk}v_k h_k t_{iq}, \qquad \forall i \in I, j \in J(i), k \in K(ij), q \in Q \tag{5.3b}$$

$$\alpha^s_{ijk} + t_{iq}\gamma^s_{ijk} + \mu^{'s}_{ijkq} \le -d^s_{ijk}v_k h_k t_{iq} + \pi d^s_{ijk}v_k, \qquad \forall i \in I, j \in J(i), k \in K(ij), q \in Q \tag{5.3c}$$

$$\beta^s_{ijk} - t_{jl}\gamma^s_{ijk} + d^s_{ijk}v_k\phi^s_{jl} + w^{'s}_{ijkq} \le d^s_{ijk}v_k h_k t_{jl}, \qquad \forall j \in J, i \in I(j), k \in K(ij), l \in L \tag{5.3d}$$

$$\beta^s_{ijk} - t_{jl}\gamma^s_{ijk} + \lambda^{'s}_{ijkl} \le d^s_{ijk}v_k h_k t_{jl} + \pi d^s_{ijk}v_k, \qquad \forall j \in J, i \in I(j), k \in K(ij), l \in L \tag{5.3e}$$

$$\gamma^s_{ijk}, \epsilon^s_{iq}, \phi^s_{jl}, u^{'s}_{ijkq}, w^{'s}_{ijkl}, \mu^{'s}_{ijkq}, \lambda^{'s}_{ijkl} \le 0, \qquad \forall i \in I, j \in J(i), k \in K(ij), q \in Q, l \in L \tag{5.3f}$$

As a result, we can rewrite Model 5.1 as:

$$\min_{x,y} \quad \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) + \begin{cases} \max \sum\limits_{s\in S} p^s z^s_{dual} \\ \text{s.t. } [5.3b] - [5.3f] \end{cases} \tag{5.4}$$

$$\text{s.t.} \quad x_{iq}, y_{jl} \in \{0,1\}, \quad i \in I, q \in Q, j \in J, l \in L$$

Define $\theta^s = \max \ [5.3a]$ Then we can rewrite Model 5.4 as:

$$[5.5] \quad \min \quad z_{MP} = \sum_{i\in I}\sum_{q\in Q} f_i(x_{iq}) + \sum_{j\in J}\sum_{l\in L} g_j(y_{jl}) + \sum_{s\in S} p^s\theta^s \tag{5.5a}$$

$$\text{s.t. } \theta^s \geq \sum_{i\in I}\sum_{j\in J(i)}\sum_{k\in K(ij)} (\alpha^{s\bar{h}}_{ijk} + \beta^{s\bar{h}}_{ijk}) + \sum_{i\in I}\sum_{q\in Q}(C_{iq}\epsilon^{s\bar{h}}_{iq})x_{iq} + \sum_{j\in J}\sum_{l\in L}(C_{jl}\phi^{s\bar{h}}_{jl})y_{jl}, \quad \forall s \in S, \bar{h} \in \bar{H} \tag{5.5b}$$

$$x_{iq}, y_{jl} \in \{0,1\}, \qquad i \in I, q \in Q, j \in J, l \in L \tag{5.5c}$$

We refer to Model 5.5 as the relaxed Master Problem (MP), where $\bar{h}$ is the iteration of the algorithm, and set $\bar{H}$ is the set of all MP solutions $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ for which we have generated cut 5.5b.

At a given iteration $\bar{h}$, and for a given MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$, we solve subproblem (5.3), $\forall s \in S$, to generate cut 5.5b to add to MP. Note that SP^{LP}_s is always feasible for any MP solution, therefore, at any iteration $\bar{h}$, only optimality cuts 5.5b are generated, and there is no need for generating feasibility cuts.

The objective function value of MP (z_{MP}) gives a lower bound (LB) on the optimal objective function value of the original problem, Model 4.2, while the upper bound (UB) is computed as $UB = \sum_{s\in S} p^s z^s_{dual} + \sum_{i\in I}\sum_{q\in Q} f_i(\bar{\boldsymbol{x}}_{\boldsymbol{iq}}) + \sum_{j\in J}\sum_{l\in L} g_j(\bar{\boldsymbol{y}}_{\boldsymbol{jl}})$. The algorithm iterates between solving the MP and SPs creating $|S|$ cuts at every iteration $\bar{h}$ until $(UB - LB)/LB < \bar{\epsilon}$, where $\bar{\epsilon}$ is an acceptable level of error.

5.1.2 Integer L-shaped Method

According to the Laporte/Louveaux L-shaped method (Laporte and Louveaux, 1993), or simply the integer L-shaped method, when the first stage variables are binary, and the subproblem(s) are integer or mixed integer programs, L-shaped cuts 5.7 are sufficient for the algorithm to converge to an integer optimal solution. When applying the integer L-shaped method to Model 4.2, the problem decomposes into MP (5.5) a set of $|S|$ disaggregated subproblems (SP^s), that are similar to SP_s^{LP}, except that the binary requirement of the decision variables is reintroduced as shown in Model 5.6. Note that we keep cuts 5.5b in the MP of the integer L-shaped method since they are valid cuts and can strengthen the MP formulation.

$$
\begin{aligned}
&[5.6]\ z^s = \min\ [5.2a] \\
&\quad \text{s.t.}\ \ [4.2\text{b}]\text{ - }[4.2\text{d}],\ [5.2\text{b}],\text{ and }[5.2\text{c}] \\
&\qquad u^s_{ijkq}, w^s_{ijkl}, \mu^s_{ijkq}, \lambda^s_{ijkl} \in \{0,1\}, \qquad i \in I, j \in J, k \in K(ij), q \in Q, l \in L
\end{aligned}
\tag{5.6}
$$

First, define the following notation. Let $\bar{\boldsymbol{Q}}$ be the set of transportation options for all suppliers that have a value of 1 in the MP solution, i.e. $\bar{\boldsymbol{x}}_{\boldsymbol{iq}} = 1 \ \ \forall\ q \in \bar{\boldsymbol{Q}}$. Similarly, $\bar{\boldsymbol{L}}$ is the set of transportation options for all customers with a value of 1 in the MP solution, i.e. $\bar{\boldsymbol{y}}_{\boldsymbol{jl}} = 1 \ \ \forall\ l \in \bar{L}$. Also, let L^s be a lower bound on the objective function value z^s of each subproblem $s \in S$. L^s is computed as z^s given that all first stage variables take the value of 1, i.e. $\bar{\boldsymbol{Q}} = Q$ and $\bar{\boldsymbol{L}} = L$.

The resulting integer L-shaped cut can be expressed as follows:

$$
\theta^s \geq (L^s - z^s)\Bigg(\Big(\sum_{q \in Q \setminus \bar{\boldsymbol{Q}}} \bar{\boldsymbol{x}}_{\boldsymbol{iq}} + \sum_{l \in L \setminus \bar{\boldsymbol{L}}} \bar{\boldsymbol{y}}_{\boldsymbol{jl}}\Big) - \Big(\sum_{q \in \bar{\boldsymbol{Q}}} \bar{\boldsymbol{x}}_{\boldsymbol{iq}} + \sum_{l \in \bar{\boldsymbol{L}}} \bar{\boldsymbol{y}}_{\boldsymbol{jl}}\Big)\Bigg) + z^s, \quad \forall s \in S \tag{5.7}
$$

For our problem, the integer L-cut (5.7) can be adjusted to take into account the fact that the objective function value of a given subproblem z^s does not decrease with the removal of a first stage transportation option $q \in \bar{\boldsymbol{Q}}$, i.e. removing some transportation options from the solution of the MP, $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$, would never result in a lower objective funciton value z^s. Therefore, constraint (5.7) can be rewritten as follows:

$$\theta^s \geq (L^s - z^s)\Big(\sum_{q \in Q \setminus \bar{\boldsymbol{Q}}} \bar{\boldsymbol{x}}_{\boldsymbol{iq}} + \sum_{l \in L \setminus \bar{\boldsymbol{L}}} \bar{\boldsymbol{y}}_{\boldsymbol{jl}}\Big) + z^s, \quad \forall s \in S \tag{5.8}$$

Similar to Benders decomposition in Section 5.1.1, the L-shaped method iteratively solves the MP and SPs until the algorithm converges to an optimal solution. Algorithm (1) summarizes the steps of the integer L-shaped method.

Algorithm 1 Integer L-shaped Method

$LB = -\inf,\ UB = \inf$
Compute lower bound L^s for each subproblem $s \in S$
while $(UB - LB)/LB > \bar{\epsilon}$ **do**
  Solve MP (5.5)
    Get solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ *and* $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$
    $LB = z_{MP}$
  for $s \in S$ **do**
    Solve dual $\boldsymbol{SP_s^{LP}}$ (5.3)
      Create Benders cut (5.5b).
    Solve $\boldsymbol{SP^s}$ (5.6)
      Get z^s and create integer L-cut (5.8)
  Add all generated cuts to **MP**
  $UB_{new} = \sum_{s \in S} p^s z^s + \sum_{i \in I} \sum_{q \in Q} f_i(\bar{\boldsymbol{x}}_{\boldsymbol{iq}}) + \sum_{j \in J} \sum_{l \in L} g_j(\bar{\boldsymbol{y}}_{\boldsymbol{jl}})$
  $UB = \min(UB, UB_{new})$
end while

5.2 Enhancement Techniques for Benders Decomposition and the L-shaped Method

One common problem that is likely to arise with the implementation of Benders decomposition is the slow convergence of the algorithm for a reasonable size problem. This problem can get worse for the integer L-shaped method, where the subproblem(s) are integer programs (IP) or mixed integer programs (MIP) since an IP or MIP is solved for each subproblem at each iteration of the algorithm. In our application, we employ some algorithm enhancement techniques to speed up the convergence of the L-shaped method. We implement partial decomposition (5.2.1), alternating cut strategy (5.2.2) and the use of a single branch-and-bound search tree through a callback routine (5.2.3).

5.2.1 Partial Decomposition

Crainic et al. (2014) propose the idea of partial decomposition for two-stage stochastic integer programs, which refers to keeping a subset of scenarios, $\bar{S} \subset S$, in the master problem, then solving for the remaining scenarios, $s \in S \backslash \bar{S}$, as subproblems. From those subproblems, we obtain the necessary cuts and proceed with the usual Benders decomposition or integer L-shaped method. Partial decomposition has been shown to improve the efficiency of the algorithm greatly and reduce the number of optimality and feasibility cuts needed to reach convergence (Crainic et al., 2014).

An important question that arises with partial decomposition is which scenarios to keep in the master problem. Crainic et al. (2014) suggest different methods for deciding on the subset of scenarios ($\bar{S}$) to maintain in the MP. For our implementation, we choose $\bar{S}$ either randomly, or using k-means clustering.

With the use of partial decomposition, Model 4.2 decomposes differently than the decomposition proposed in Section (5.1). While the formulation of the subproblems (5.2 and 5.6) stays the same, the MP becomes equivalent to the full formulation of the original problem but only for a subset of scenarios $\bar{S} \subset S$. This results in the following formulation of the relaxed master problem (MP).

$$[5.9]: \quad z'_{MP} = \min \sum_{s \in S \setminus \bar{S}} p^s \theta^s + \sum_{i \in I} \sum_{q \in Q} f_i(x_{iq}) + \sum_{j \in J} \sum_{l \in L} g_j(y_{jl})$$
$$+ \sum_{s \in \bar{S}} \sum_{i \in I} \sum_{j \in J(i)} \sum_{k \in K(ij)} p^s [d^s_{ijk} v_k h_k (\sum_{l \in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}) - \sum_{q \in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq}))]$$
$$+ \sum_{s \in \bar{S}} \sum_{i \in I} \sum_{q \in Q} \sum_{j \in J(i)} \sum_{k \in K(ij)} p^s \pi d^s_{ijk} v_k (\mu^s_{ijkq}) + \sum_{s \in \bar{S}} \sum_{j \in J} \sum_{l \in L} \sum_{i \in I(j)} \sum_{k \in K(ij)} p^s \pi d^s_{ijk} v_k (\lambda^s_{ijkl}) \tag{5.9a}$$

$$\text{s.t. } \theta^s \geq (L^s - z^s)\Big(\sum_{q \in Q \setminus \bar{\boldsymbol{Q}}} \bar{\boldsymbol{x}}_{\boldsymbol{iq}} + \sum_{l \in L \setminus \bar{\boldsymbol{L}}} \bar{\boldsymbol{y}}_{\boldsymbol{jl}} \Big) + z^s, \qquad \forall s \in S \setminus \bar{S} \tag{5.9b}$$

$$\theta^s \geq \sum_{i \in I} \sum_{j \in J(i)} \sum_{k \in K(ij)} (\alpha^{s\bar{h}}_{ijk} + \beta^{s\bar{h}}_{ijk}) + \sum_{i \in I} \sum_{q \in Q} (C_{iq} \epsilon^{s\bar{h}}_{iq}) x_{iq} + \sum_{j \in J} \sum_{l \in L} (C_{jl} \phi^{s\bar{h}}_{jl}) y_{jl}, \quad \forall s \in S \setminus \bar{S} \tag{5.9c}$$

$$\sum_{q \in Q} (u^s_{ijkq} + \mu^s_{ijkq}) = 1, \qquad i \in I, j \in J(i), k \in K(ij), s \in \bar{S} \tag{5.9d}$$

$$\sum_{l \in L} (w^s_{ijkl} + \lambda^s_{ijkl}) = 1, \qquad j \in J, i \in I(j), k \in K(ij), s \in \bar{S} \tag{5.9e}$$

$$\sum_{q \in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq}) \leq \sum_{l \in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}), \qquad j \in J, i \in I(j), k \in K(ij) s \in \bar{S} \tag{5.9f}$$

$$\sum_{j \in J(i)} \sum_{k \in K(ij)} d^s_{ijk} v_k u^s_{ijkq} \leq C_{iq} x_{iq}, \qquad i \in I, q \in Q, s \in \bar{S} \tag{5.9g}$$

$$\sum_{i \in I(j)} \sum_{k \in K(ij)} d^s_{ijk} v_k w^s_{ijkl} \leq C_{jl} y_{jl}, \qquad j \in J, l \in L, s \in \bar{S} \tag{5.9h}$$

$$x_{iq}, y_{jl}, u^s_{ijkq}, \mu^s_{ijkq}, w^s_{ijkl}, \lambda^s_{ijkl} \in \{0, 1\}, \qquad i \in I, q \in Q, j \in J(i), k \in K(ij), s \in \bar{S} \tag{5.9i}$$

5.2.2 Alternating Cut Strategy for Integer L-shaped Method

Angulo et al. (2016) suggest the alternating cut strategy for the integer L-shaped method. Instead of solving both the LP and IP subproblems at each iteration, the authors suggest alternating between them. For a given MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$, and for a subproblem $s \in S$, we start by solving the LP subproblem (5.2) we only solve the IP subproblem (5.6) if the Benders cut (5.5b) has already been generated at a previous iteration, or if $\theta^s > z^s_{LP}$. The steps of the algorithm are summarized in Algorithm (2), where $\bar{V}^s$ is the set of MP solutions $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ at which the IP subproblem $s \in S$ was solved, and $\bar{V}^s_{LP}$ is the set of MP solutions $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ at which the LP subproblem $s \in S$ was solved. This ensures that we do not assess the same solution twice in order to save on computational time.

Algorithm 2 Alternating Cuts Strategy (Angulo et al., 2016)

Solve relaxed Master Problem [MP] (5.9)
    Get $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ *and* $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$
for $s \in S$ **do**
    if $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}} \in \bar{V}^s$ **then**
        return
    end if
    if $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}} \notin \bar{V}_{LP}$ **then**
        Compute z^s_{LP} (5.2)
        $\bar{V}_{LP} \leftarrow \bar{V}_{LP} \cup \{\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}\}$
        Add BD cut (5.5b)
        if $\theta^s < z^s_{LP}$ **then**
            return
        end if
    end if
    Compute z^s (5.6)
    $\bar{V}^s \leftarrow \bar{V}^s \cup \{\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}\}$
    if $\theta^s < z^s$ **then**
        Add Integer L-cut (5.8)

5.2.3 Single Tree Search for the MP with a Callback Routine

In the classic implementation of BD, the relaxed master problem (MP) is solved from scratch at every iteration. This means that a new branch-and-bound tree is composed at every iteration with no regards to solutions of earlier iterations. This, in turn, results in a lot of computational demand and inefficiency in the algorithm, especially that some of the generated cuts may not change the feasible region of the MP.

Nonetheless, advanced commercial solvers, like CPLEX, allow for the use of a single branch-and-bound search tree for mixed integer programs with the lazy constraint callback routine, where generated cuts are added lazily, i.e. only if they change the feasible region of the MP. The callback routine also guarantees that a given node in the branch-and-bound tree is only explored once throughout the whole algorithm. This could result in significant improvements in the efficiency of the algorithm.

We implement all of the three enhancement techniques in our numerical testing in Chapter 6. In addition to that, in the following section, we experiment with a number of valid cuts for Model 4.2, with the goal of generating stronger cuts than the L-shaped cut, that would ultimately improve the time it takes for our algorithm to converge to an optimal solution.

5.3 Additional Valid Cuts

5.3.1 Valid Cut 1

The integer L-shaped cuts by Laporte/Louveaux, introduced in Section 5.1.2, are general cuts that can be applied to any stochastic program when the first stage variables are all binary. The cuts, however, may be weak as they set the value of θ^s to its lower bound (L^s) or a lower value, if there is a change in MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$.

We propose a set of valid optimality cuts to better approximate the change in subproblem objective values θ^s when the MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ changes. In contrast to the L-shaped cut that bounds θ^s by its lower bound (L^s), the proposed cut estimates the maximum impact of opening transportation options with the same dispatch time as the occupied extra capacity, i.e. extra capacity bought when demand cannot be satisfied with the given available transportation options. We compute δ^s, the maximum improvement in subproblem objective function value if such options become available.

Let $\hat{\boldsymbol{x}}_{\boldsymbol{iq}} = \bar{\boldsymbol{x}}_{\boldsymbol{iq}}$, and $\hat{\boldsymbol{y}}_{\boldsymbol{jl}} = \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$. Also define $\boldsymbol{Q'} \subseteq Q \backslash \bar{\boldsymbol{Q}}$ and $\boldsymbol{L'} \subseteq L \backslash \bar{\boldsymbol{L}}$ as the sets of transportation options such that $\mu^s_{ijkq} = 1$ and $\lambda^s_{ijkl} = 1$, respectively, for any product k and any demand pair (i, j). Then, for all $q \in \boldsymbol{Q'}$, we set $x_{iq} = 1,\ x_{iq} \in \hat{\boldsymbol{x}}_{\boldsymbol{iq}}$, and for all $l \in \boldsymbol{L'}$, we set $y_{jl} = 1,\ y_{jl} \in \hat{\boldsymbol{y}}_{\boldsymbol{jl}}$. Also let $SP^s(\hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \hat{\boldsymbol{y}}_{\boldsymbol{jl}})$ denote subproblem SP^s given the MP solution $\hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \hat{\boldsymbol{y}}_{\boldsymbol{jl}}$. We solve $SP^s(\hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \hat{\boldsymbol{y}}_{\boldsymbol{jl}})$ to get objective function value z^s_{new}. Then we calculates $\delta^s = z^s_{new} - z^s$. Valid cut 1 is expressed as:

$$\theta^s \geq \delta^s(\sum_{q \in \boldsymbol{Q'}} x_{iq} + \sum_{l \in \boldsymbol{L'}} y_{jl}) + (L^s - z^s)(\sum_{q \in Q \backslash \{\bar{\boldsymbol{Q}} \cup \boldsymbol{Q'}\}} x_{iq} + \sum_{l \in L \backslash \{\bar{\boldsymbol{L}} \cup \boldsymbol{L'}\}} y_{jl}) + z^s, \quad \forall s \in S \tag{5.10}$$

Algorithm 3 summarizes the steps of generating valid cut 1.

Justification:

The values of δ^s are computed exactly as the improvement in objective function value when additional transportation options become available, i.e. $\bar{\boldsymbol{Q}} \leftarrow (\bar{\boldsymbol{Q}} \cup \boldsymbol{Q'})$ and $\bar{\boldsymbol{L}} \leftarrow (\bar{\boldsymbol{L}} \cup \boldsymbol{L'})$. Therefore, in contrast to the L-shaped cut, if one or more of the transportation

Algorithm 3 Valid Cut 1

$\hat{\boldsymbol{x}}_{\boldsymbol{iq}} = \bar{\boldsymbol{x}}_{\boldsymbol{iq}}$, $\hat{\boldsymbol{y}}_{\boldsymbol{jl}} = \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$
for $i \in I,\ q \in Q \backslash \bar{\boldsymbol{Q}}$ **do**
  if $x_{iq} = 0$ & $\sum_{j \in J(i)} \sum_{k \in K(ij)} \mu^s_{ijkq} > 0$ **then**
    Set $x_{iq} = 1$, for $x_{iq} \in \hat{\boldsymbol{x}}_{\boldsymbol{iq}}$
for $j \in J,\ l \in L \backslash \bar{\boldsymbol{L}}$ **do**
  if $y_{jl} = 0$ & $\sum_{i \in I(j)} \sum_{k \in K(ij)} \lambda^s_{ijkl} > 0$ **then**
    Set $y_{jl} = 1$, for $y_{jl} \in \hat{\boldsymbol{y}}_{\boldsymbol{jl}}$
$z^s_{new} = SP^s(\hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \hat{\boldsymbol{y}}_{\boldsymbol{jl}})$
$\delta^s = z^s_{new} - z^s$
if $\delta^s < L^s - z^s$ **then**
  $\delta^s = L^s - z^s$
Generate valid cut 1 (5.10)

options $\boldsymbol{Q'}$ and $\boldsymbol{L'}$ take a value of 1 in the MP solution in a subsequent iteration, valid cut 1 sets the value of θ^s to δ^s or a lower value, rather than setting it to its lower bound L^s or a lower value. Note that $\delta^s \geq L^s - z^s$, and therefore valid cut 1 is at least as strong as the L-cut.

5.3.2 Valid Cut 2

In this section, we propose another valid cut with the goal of estimating the decrease in the objective function value of the subproblem with the addition of one transportation option at a time. The transportation option could correspond to a particular supplier or customer. In other words, instead of a single δ^s that estimates the change in θ^s, as in valid cut 1 (5.3.1), we calculate $\delta^s_{iq},\ \forall q \in Q \backslash \bar{\boldsymbol{Q}}$ and $\delta^s_{jl},\ \forall l \in L \backslash \bar{\boldsymbol{L}}$ for each supplier and customer and each transportation option that is unavailable at that given iteration. Our goal is to measure how much the objective function value of the subproblem z^s improves if we are to add a transportation option that is originally unavailable, i.e. $q \in Q \backslash \bar{\boldsymbol{Q}}$ or $l \in L \backslash \bar{\boldsymbol{L}}$, given the solution of the MP, $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$. As a result, the cut tightly underestimate the value function θ^s. Valid cut 2 can be expressed as follows:

$$\theta^s \geq \sum_{q \in Q \backslash \bar{\boldsymbol{Q}}} \delta^s_{iq} x_{iq} + \sum_{l \in L \backslash \bar{\boldsymbol{L}}} \delta^s_{jl} y_{jl} + z^s, \quad \forall s \in S \tag{5.11}$$

In order to calculate δ^s_{iq} and δ^s_{jl}, it is important to note that the objective function value of a subproblem z^s at a specific iteration is dependent upon the particular set of available transportation options $\bar{\boldsymbol{Q}}$ and $\bar{\boldsymbol{L}}$. Therefore, when computing δ^s_{iq} for a particular transportation option q and supplier i, we compute it independent of the customer side $J(i)$, i.e. we assume that all transportation options for all customers $J(i)$ are available. Similarly, we compute δ^s_{jl} for a particular transportation option l and customer j independent of the supplier side $I(j)$. Let $\hat{\boldsymbol{x}}_{\boldsymbol{iq}} = \bar{\boldsymbol{x}}_{\boldsymbol{iq}}$, and $\hat{\boldsymbol{y}}_{\boldsymbol{jl}} = \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$. To calculate δ^s_{iq} for a given $q \in Q \backslash \bar{\boldsymbol{Q}}$, we set $x_{iq} = 1, x_{iq} \in \hat{\boldsymbol{x}}_{\boldsymbol{iq}}$. We also set $y_{jl} = 1,\ y_{jl} \in \hat{\boldsymbol{y}}_{\boldsymbol{jl}},\ \forall j \in J(i),\ \forall l \in L$. Then we solve $SP^s(\hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \hat{\boldsymbol{y}}_{\boldsymbol{jl}})$ to get z^s_{new} and we compute $\delta^s_{iq} = z^s_{new} - z^s$. δ^s_{jl} is computed in a similar manner.

It is worth noting that resolving SP^s for every $q \in \bar{\boldsymbol{Q}}$ and $l \in \bar{\boldsymbol{L}}$ would be inefficient and computationally demanding. Therefore, instead of computing the exact values of δ^s_{iq} and δ^s_{jl}, we compute lower bounds on those values that are obtained by relaxing the capacity restrictions of the subproblem. For a given supplier i and a given transportation option $q \in Q \backslash \bar{\boldsymbol{Q}}$, we solve for the best allocation of all products $k \in K(ij)$ ordered by all customers $j \in J(i)$ to available transportation options from the supplier side $(\bar{\boldsymbol{Q}} \cup \{q\})$ or extra capacity options $(Q \backslash (\bar{\boldsymbol{Q}} \cup \{q\}))$, as well as all transportation options from the customer side (L). So, for every $j \in J(i)$ and every $k \in K(ij)$, we choose q and l that minimize the following expression: $\min_{q,l}\ [(t_{jl} - t_{iq}) + \pi(1 - \hat{x}_{iq})]$, such that $t_{jl} \geq t_{iq}$. This is shown in Figure 5.1. Then the resulting cost of the chosen allocation would be:

$$\begin{aligned} \bar{z}_{ijk} = \min_{q,l} \ & [d^s_{ijk} v_k h_k (t_{jl} - t_{iq}) + \pi d^s_{ijk} v_k (1 - \hat{x}_{iq})] \\ \text{s.t.} \ \ & t_{jl} \geq t_{iq} \end{aligned} \tag{5.12}$$

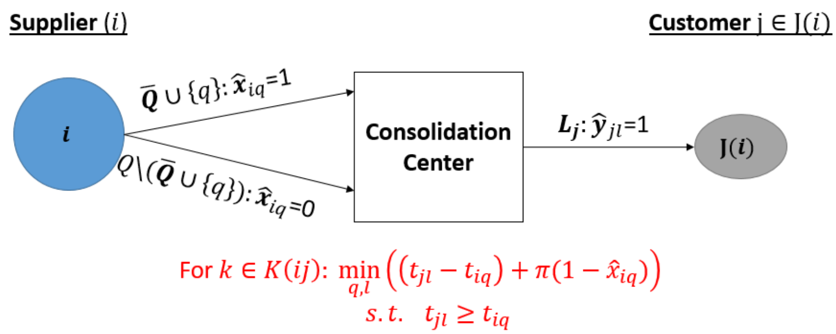

Figure 5.1: Optimal Allocation for Uncapacitated Case - Valid Cut 2

Denote $\bar{u}^s_{ijkq}, \bar{w}^s_{ijkl}, \bar{\mu}^s_{ijkq}$, and $\bar{\lambda}^s_{ijkl}$ as the solution of SP^s at a given iteration. δ^s_{iq} is then computed as:

$$\begin{aligned} \delta^s_{iq} = & \sum_{j \in J(i)} \sum_{k \in K(ij)} \bar{z}_{ijk} - [\sum_{j \in J(i)} \sum_{k \in K(ij)} d^s_{ijk} v_k h_k (\sum_{l \in L} t_{jl} (\bar{w}^s_{ijkl} + \bar{\lambda}^s_{ijkl}) - \sum_{q \in Q} t_{iq} (\bar{u}^s_{ijkq} + \bar{\mu}^s_{ijkq})) \\ & + \sum_{q \in Q} \sum_{j \in J(i)} \sum_{k \in K(ij)} \sum_{l \in L} \pi d^s_{ijk} v_k (\bar{\mu}^s_{ijkq} + \bar{\lambda}^s_{ijkl})] \end{aligned} \tag{5.13}$$

Since the computed δ^s_{iq} value is a lower bound on the value actually obtained if the SP^s is solved to optimality, we check if $\delta^s_{iq} < L^s - z^s$. If so, we set $\delta^s_{iq} = L^s - z^s$, since the objective function value of a given subproblem with the addition of a transportation option cannot be lower than the lower bound of the subproblem L^s. It should be noted that relaxing the capacities of transportation options when calculating the values of δ^s_{iq} ensures that the resulting cut does not eliminate the optimal solution, without having to solve for the exact values of δ^s_{iq}. This is because we are overestimating the maximum improvement of the objective function value of a given subproblem, and consequently, underestimating the value function θ^s. A setback of this approach, however, is that the generated cuts may be weak.

In the following section, we experiment with a different set of valid cuts.

5.3.3 Valid Cut 3

Valid Cut 3A

To address the issues that contribute to the weakness of valid cut 2, valid cut 3 is proposed, where we create a separate cut for each transportation option $q \in Q \backslash \bar{\boldsymbol{Q}}$ and $l \in L \backslash \bar{\boldsymbol{L}}$, and assume that the rest of the MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ stays the same. Therefore, if in a subsequent iteration, the MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ adds only one additional transportation option $q \in Q \backslash \bar{\boldsymbol{Q}}$ or $l \in L \backslash \bar{\boldsymbol{L}}$, we know that the objective function value of the subproblem z^s would decrease by a maximum of δ^s_{iq} and as a result, the cut would set θ^s to the value of $z^s + \delta^s_{iq}$. Otherwise, if additional changes take place, θ^s would take a value less than or equal to the subproblem lower bound L^s.

Denote $z^s(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}})$ as the objective function value of subproblem s, given that a subset of transportation options $\bar{\boldsymbol{Q}}$ and $\bar{\boldsymbol{L}}$ have a value of 1 in the MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$. Then for any subset $\bar{\boldsymbol{Q}} \subseteq Q$ and for any $q^1 \subseteq Q \backslash \bar{\boldsymbol{Q}}$, or for any subset $\bar{\boldsymbol{L}} \subseteq L$ and for any $l^1 \subseteq L \backslash \bar{\boldsymbol{L}}$, we have:

$$
\begin{aligned}
z^s(\bar{\boldsymbol{Q}} \cup \{q^1\}, \bar{\boldsymbol{L}}) &\geq z^s(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}}) + \delta^s_{iq}, \quad \forall q^1 \in Q \backslash \bar{\boldsymbol{Q}},\ \forall s \in S \\
z^s(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}} \cup \{l^1\}) &\geq z^s(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}}) + \delta^s_{jl}, \quad \forall l^1 \in L \backslash \bar{\boldsymbol{L}},\ \forall s \in S
\end{aligned}
\tag{5.14}
$$

and therefore, inequalities of the form:

$$\begin{aligned}
\theta^s &\geq \delta^s_{iq} x_{iq^1} + (L^s - z^s)[\sum_{q \in Q \backslash \{\bar{\boldsymbol{Q}} \cup \{q^1\}\}} x_{iq} + \sum_{l \in L \backslash \bar{\boldsymbol{L}}} y_{jl}] + z^s, \forall q^1 \in Q \backslash \bar{\boldsymbol{Q}}, \forall s \in S \\
\theta^s &\geq \delta^s_{jl} y_{jl^1} + (L^s - z^s)[\sum_{q \in Q \backslash \bar{\boldsymbol{Q}}} x_{iq} + \sum_{l \in L \backslash \{\bar{\boldsymbol{L}} \cup \{l^1\}\}} y_{jl}] + z^s, \forall l^1 \in L \backslash \bar{\boldsymbol{L}}, \forall s \in S
\end{aligned} \tag{5.15}$$

are valid in the relaxed MP. We refer to inequalities 5.15 as valid cut 3A.

Similar to valid cut 2, the values of δ^s_{iq} and δ^s_{jl} can be obtained by either resolving SP^s to obtain the exact values, or obtaining lower bounds by relaxing capacity restrictions. Since we create a separate cut for each unavailable transportation option $q \in Q \backslash \bar{\boldsymbol{Q}}$ and $l \in L \backslash \bar{\boldsymbol{L}}$, we do not need to compute the values of δ^s_{iq} and δ^s_{jl} independent of the other side of the network, as done with valid cut 2. That is because valid cut 3A would set θ^s to the value of $z^s + \delta^s_{iq}$ only if a single transportation option is added to a given MP solution $\bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \bar{\boldsymbol{y}}_{\boldsymbol{jl}}$. Otherwise, similar to the L-cut, the cut would set the value of θ^s to the subproblem lower bound L^s or a lower value.

To compute the values of δ^s_{iq} while relaxing capacity, for a particular supplier i and a transportation option $q^1 \in Q \backslash \bar{\boldsymbol{Q}}$, let $\hat{\boldsymbol{x}}_{\boldsymbol{iq}} = \bar{\boldsymbol{x}}_{\boldsymbol{iq}}$, and set $x_{iq} = 1, x_{iq} \in \hat{\boldsymbol{x}}_{\boldsymbol{iq}}$. Then for each customer $j \in J(i)$ that orders from supplier i and each product $k \in K(ij)$ ordered by customer j, we solve for the best allocation from the supplier side to available transportation options $(\bar{\boldsymbol{Q}} \cup \{q^1\})$ or extra capacity options $(Q \backslash (\bar{\boldsymbol{Q}} \cup \{q^1\}))$, as well as available transportation options for the customer side $(\bar{\boldsymbol{L}})$ or extra capacity options $(L \backslash \bar{\boldsymbol{L}})$. So, for every $j \in J(i)$ and every $k \in K(ij)$, we choose q and l that minimize the following expression: $\min_{q,l} \; [(t_{jl} - t_{iq}) + \pi\big((1 - \hat{x}_{iq}) + (1 - \bar{y}_{jl})\big)]$, such that $t_{jl} \geq t_{iq}$. This is shown in Figure 5.2. Then the resulting cost of the chosen allocation would be:

$$\bar{z}_{ijk} = \min_{q,l} \; [d^s_{ijk} v_k h_k (t_{jl} - t_{iq}) + \pi d^s_{ijk} v_k \big((1 - \hat{x}_{iq}) + (1 - \bar{y}_{jl})\big)] \quad (5.16)$$
$$\text{s.t.} \quad t_{jl} \geq t_{iq}$$

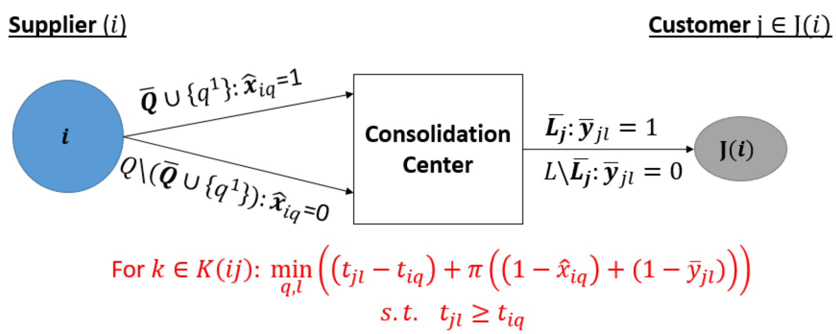

Figure 5.2: Optimal Allocation for Uncapacitated Case - Valid Cut 3A

δ^s_{iq} is then computed as:

$$\delta^s_{iq} = \sum_{j \in J(i)} \sum_{k \in K(ij)} \bar{z}_{ijk} - [\sum_{j \in J(i)} \sum_{k \in K(ij)} d^s_{ijk} v_k h_k (\sum_{l \in L} t_{jl} (\bar{w}^s_{ijkl} + \bar{\lambda}^s_{ijkl}) - \sum_{q \in Q} t_{iq} (\bar{u}^s_{ijkq} + \bar{\mu}^s_{ijkq}))$$
$$+ \sum_{q \in Q} \sum_{j \in J(i)} \sum_{k \in K(ij)} \sum_{l \in L} \pi d^s_{ijk} v_k (\bar{\mu}^s_{ijkq} + \bar{\lambda}^s_{ijkl})] \quad (5.17)$$

An alternative way of computing the values of δ^s_{iq} is to solve a relaxed MIP formulation of SP^s. For a given supplier i, denote it as SP^s_i, for every $q \in Q \backslash \bar{\boldsymbol{Q}}$. δ^s_{iq} is then calculated by solving SP^s_i and computing the difference in the objective function values, $z^s_i(\{q^1\} \cup \bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}}) - z^s_i(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}})$, where $z^s_i(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}})$ is computed as the objective function value (5.18a) given the solution $\bar{u}^s_{ijkq}, \bar{w}^s_{ijkl}, \bar{\mu}^s_{ijkq}$, and $\bar{\lambda}^s_{ijkl}$ to $SP^s(\bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}})$.

$$z_i^s(\{q^1\} \cup \bar{\boldsymbol{Q}}, \bar{\boldsymbol{L}}) = \min \sum_{j\in J(i)} \sum_{k\in K(ij)} d_{ijk}^s v_k h_k [(\sum_{l\in L} t_{jl}(w_{ijkl}^s + \lambda_{ijkl}^s) - (\sum_{q\in Q} t_{iq}(u_{ijkq}^s + \mu_{ijkq}^s)]$$

$$+ \sum_{q\in Q} \sum_{j\in J(i)} \sum_{k\in K(ij)} \pi d_{ijk}^s v_k (\mu_{ijkq}^s) + \sum_{j\in J(i)} \sum_{l\in L} \sum_{k\in K(ij)} \pi d_{ijk}^s v_k (\lambda_{ijkl}^s) \tag{5.18a}$$

$$\text{s.t. } \sum_{q\in Q} (u_{ijkq}^s + \mu_{ijkq}^s) = 1, \qquad j \in J(i), k \in K(ij) \tag{5.18b}$$

$$\sum_{l\in L} (w_{ijkl}^s + \lambda_{ijkl}^s) = 1, \qquad j \in J(i), k \in K(ij) \tag{5.18c}$$

$$\sum_{q\in Q} t_{iq}(u_{ijkq}^s + \mu_{ijkq}^s) \le \sum_{l\in L} t_{jl}(w_{ijkl}^s + \lambda_{ijkl}^s), \qquad j \in J(i), k \in K(ij) \tag{5.18d}$$

$$\sum_{j\in J(i)} \sum_{k\in K(ij)} d_{ijk}^s v_{ik} u_{ijkq}^s \le S_{iq} \hat{\boldsymbol{x}}_{\boldsymbol{iq}}, \qquad q \in Q \tag{5.18e}$$

$$\sum_{k\in K(ij)} d_{ijk}^s v_{ik} w_{ijkl}^s \le S_{jl} \bar{\boldsymbol{y}}_{\boldsymbol{jl}}, \qquad j \in J(i),\ l \in L \tag{5.18f}$$

$$u_{ijkq}^s, w_{ijkl}^s, \mu_{ijkq}^s, \lambda_{ijkl}^s \in \{0,1\}, \qquad j \in J(i), k \in K(ij), q \in Q, l \in L \tag{5.18g}$$

Similarly, $\delta_{jl^1}^s$ can be computed by solving the relaxation in Formulation 5.19, where $\hat{\boldsymbol{y}}_{\boldsymbol{jl}}$ is equivalent to $\bar{\boldsymbol{y}}_{\boldsymbol{jl}}$ except that we set the value of $y_{jl} = 1$ for $l^1 \in L \backslash \bar{\boldsymbol{L}}$.

$$z_j^s(\bar{\boldsymbol{Q}}, \{l^1\} \cup \bar{\boldsymbol{L}}) = \min \sum_{i\in I(j)} \sum_{k\in K(ij)} d_{ijk}^s v_k h_k [(\sum_{l\in L} t_{jl}(w_{ijkl}^s + \lambda_{ijkl}^s) - (\sum_{q\in Q} t_{iq}(u_{ijkq}^s + \mu_{ijkq}^s)]$$

$$+ \sum_{i\in I(j)} \sum_{q\in Q} \sum_{k\in K(ij)} \pi d_{ijk}^s v_k (\mu_{ijkq}^s) + \sum_{l\in L} \sum_{i\in I(j)} \sum_{k\in K(ij)} \pi d_{ijk}^s v_k (\lambda_{ijkl}^s) \tag{5.19a}$$

$$\text{s.t.} \quad \sum_{q \in Q} (u^s_{ijkq} + \mu^s_{ijkq}) = 1, \qquad i \in I(j), k \in K(ij) \quad (5.19b)$$

$$\sum_{l \in L} (w^s_{ijkl} + \lambda^s_{ijkl}) = 1, \qquad i \in I(j), k \in K(ij) \quad (5.19c)$$

$$\sum_{q \in Q} t_{iq}(u^s_{ijkq} + \mu^s_{ijkq}) \leq \sum_{l \in L} t_{jl}(w^s_{ijkl} + \lambda^s_{ijkl}), \qquad i \in I(j), k \in K(ij) \quad (5.19d)$$

$$\sum_{k \in K(ij)} d^s_{ijk} v_{ik} u^s_{ijkq} \leq S_{iq} \bar{\boldsymbol{x}}_{\boldsymbol{iq}}, \qquad i \in I(j), q \in Q \quad (5.19e)$$

$$\sum_{i \in I(j)} \sum_{k \in K(ij)} d^s_{ijk} v_{ik} w^s_{ijkl} \leq S_{jl} \hat{\boldsymbol{y}}_{\boldsymbol{jl}}, \qquad l \in L \quad (5.19f)$$

$$u^s_{ijkq}, w^s_{ijkl}, \mu^s_{ijkq}, \lambda^s_{ijkl} \in \{0, 1\}, \qquad i \in I(j), k \in K(ij), q \in Q, l \in L \quad (5.19g)$$

In the following section, we propose an additional valid cut that is an extension of valid cut 3A.

Valid Cut 3B

Similar to valid cut 3A, we now define $q^2 \subset Q \backslash \bar{\boldsymbol{Q}}$ and $l^2 \subset \bar{\boldsymbol{L}}$ as sets of two transportation options that are not available in a given MP solution, and are to be added at the same time to a given set of available options $\bar{\boldsymbol{Q}}$ and $\bar{\boldsymbol{L}}$. We then calculate the maximum improvement in the objective function value of a subproblem SP^s when q^2 and l^2 are added to the MP solution as $\delta^s_{iq^2}$ and $\delta^s_{jl^2}$ for a given supplier i and a given customer j, respectively.

Note that, instead of enumerating all possible combinations of two options in $\bar{\boldsymbol{Q}}$ for a specific supplier i, we focus on the single transportation option $q \in Q \backslash \bar{\boldsymbol{Q}}$ with the highest possible impact, i.e. the min δ^s_{iq}, and estimate the total impact if it is to be added with another option $q \in Q \backslash \bar{\boldsymbol{Q}}$ with a $\delta^s_{iq} < 0$. Since we want the cut to hold if both options are chosen at the same time and to be voided if only one is chosen, we add $3\delta^s_{iq^2}$ to the right hand side of the cut, expressed in Equations 5.20. Note that $\delta^s_{iq^2}$, similar to δ^s_{iq}, is a negative number. Note that $\delta^s_{iq^2}$ is always less than or equal to δ^s_{iq} for each of the two options. In other words, the combined benefit of having the two options q^2 available is greater than or equal to the benefit of having only one of them available. When just one of the transportation options becomes available, Inequalities 5.20 become redundant with the presence of valid cut 3A (5.15), and therefore cut 3B is valid.

$$\begin{aligned}
\theta^s &\geq -\delta^s_{iq^2}\sum_{q\in q^2} x_{iq} + (L^s - z^s)[\sum_{q\in Q\backslash\{q^2\cup\bar{\boldsymbol{Q}}\}} x_{iq} + \sum_{l\in L\backslash\bar{\boldsymbol{L}}} y_{jl}] + z^s + 3\delta^s_{iq^2}, \;\; \forall q^2 \subset Q\backslash\bar{\boldsymbol{Q}}, \forall s \in S \\
\theta^s &\geq -\delta^s_{jl^2}\sum_{l\in l^2} y_{jl} + (L^s - z^s)[\sum_{q\in Q\backslash\bar{\boldsymbol{Q}}} x_{iq} + \sum_{l\in L\backslash\{l^2\cup\bar{\boldsymbol{L}}\}} y_{jl}] + z^s + 3\delta^s_{jl^2}, \;\; \forall l^2 \subset L\backslash\bar{\boldsymbol{L}}, \forall s \in S
\end{aligned} \tag{5.20}$$

Similar to valid cut 3A, to compute the values of $\delta^s_{iq^2}$ while relaxing capacity, for a particular supplier i and a set of two transportation options $q^2 \subset Q\backslash\bar{\boldsymbol{Q}}$, let $\hat{\boldsymbol{x}}^{\boldsymbol{s}}_{\boldsymbol{iq}} = \bar{\boldsymbol{x}}^{\boldsymbol{s}}_{\boldsymbol{iq}}$ and set $x_{iq} = 1, x_{iq} \in \hat{\boldsymbol{x}}^{\boldsymbol{s}}_{\boldsymbol{iq}}$ for each $q \in q^2$. Then for each customer $j \in J(i)$ that orders from supplier i, and each product $k \in K(ij)$ that customer j orders, we solve for the best allocation from the supplier side to available transportation options ($\bar{\boldsymbol{Q}} \cup q^2$) or extra capacity options ($Q\backslash(\bar{\boldsymbol{Q}} \cup q^2)$), as well as available transportation options for the customer side ($\bar{\boldsymbol{L}}$) or extra capacity options ($L\backslash\bar{\boldsymbol{L}}$). So, for every $j \in J(i)$ and every $k \in K(ij)$, we choose q and l that minimize the following expression: $\min_{q,l}\; [(t_{jl} - t_{iq}) + \pi\big((1 - \hat{x}_{iq}) + (1 - \bar{y}_{jl})\big)]$, such that $t_{jl} \geq t_{iq}$.
Then the resulting cost of the chosen allocation would be:

$$\begin{aligned}
\bar{z}_{ijk} = &\min_{q,l}\; [d^s_{ijk} v_k h_k (t_{jl} - t_{iq}) + \pi d^s_{ijk} v_k \big((1 - \hat{x}_{iq}) + (1 - \bar{y}_{jl})\big)] \\
&\text{s.t. } \; t_{jl} \geq t_{iq}
\end{aligned} \tag{5.21}$$

$\delta^s_{iq^2}$ is then computed as:

$$\delta^s_{iq^2} = \sum_{j\in J(i)} \sum_{k\in K(ij)} \bar{z}_{ijk} - [\sum_{j\in J(i)} \sum_{k\in K(ij)} d^s_{ijk} v_k h_k (\sum_{l\in L} t_{jl}(\bar{w}^s_{ijkl} + \bar{\lambda}^s_{ijkl}) - \sum_{q\in Q} t_{iq}(\bar{u}^s_{ijkq} + \bar{\mu}^s_{ijkq})) + \sum_{q\in Q} \sum_{j\in J(i)} \sum_{k\in K(ij)} \sum_{l\in L} \pi d^s_{ijk} v_k (\bar{\mu}^s_{ijkq} + \bar{\lambda}^s_{ijkl})] \tag{5.22}$$

In the following chapter, we conduct some numerical testing to evaluate the performance of our proposed solution methodology and of the valid cuts.

Chapter 6

Numerical Testing

In this chapter, we outline numerical testing results of our proposed solution methodology in Chapter 5 as well as for CPLEX and the L-shaped method for Model 4.2. First, we start with discussing how the data is generated in our test instances.

6.1 Data Generation

The data generation method outlined in Section (3.3) is adjusted to take different sets of values for two main reasons. First of all, after observing the results in Section (3.4), we noticed that for all suppliers, the total demand from all customers is small enough to fit in a single transportation option. This is also the case for the total demand of customers from all suppliers. This results in a special case of the model that may be affecting the results or making the model easier to solve. Also, since Model 4.2 has a very high number of binary variables, we reduce parameters $\bar{K}_i$ and $\bar{K}_j$ to be in the range [10, 15]. Additionally, the capacity of each transportation option for each supplier/customer C_{iq} and C_{jl} is uniformly distributed in the range [1000, 5000]. Moreover, each product, k_i, is assumed to have a certain uniform distribution of demand, where the range of that distribution is randomly generated. The lower bound of the distribution for a specific product k_i is between [1, 3], and the width of the distribution is between [1, 5]. Consequently, for a particular supplier i, a given customer j and product k, the demand d^s_{ijk} is uniformly distributed between the ranges of the particular distribution of product k_i. Finally, the volume of each unit of product v_k is uniform in the range [50, 150].

Furthermore, the number of scenarios S represents all possible realizations of demand with their associated probabilities. Section (6.2) outlines how the we generate scenarios and what is an acceptable number of scenarios. The probability of the realization of each scenario is also randomly generated with the sum of all probabilities equal to 1. The penalty cost π is assumed to be 500 for each unit of demand that exceeds the capacity of the reserved transportation option.

6.2 Scenario Generation

As discussed in Chapter 2, Kaut and Wallace (2003a) provide a brief comparison of different commonly used scenario-generation techniques. Since in Formulation 4.2 the demand parameter (d^s_{ijk}) is the only random variable in the model, i.e. the model has a univariate random variable, Conditional Sampling is considered a good fit for generating the scenario tree. After the scenario tree is generated, the model is tested for stability to ensure that it provides quality solutions to the stochastic programming model and that the results of the model are not the effect of randomness.

To test the stability of the model, for a given instance, we create three sub-instances by only changing the demand parameter (d^s_{ijk}) and randomly generating it again following the same discrete uniform distribution of the particular product k. Then the model is solved again for all the sub-instances. The objective function value of all sub-instances are compared and the gap is calculated. If the gap between the highest and lowest objective function value is less than or equal to 1%, then we would conclude that in-sample stability has been reached, and the associated number of scenarios would be considered a good estimation of the distribution for the specific model in hand.

Since the model itself is hard to solve, when measuring *in-sample stability* we choose arbitrary values for first stage variables, and refer to the fixed values as $\bar{x}_{iq}$ and $\bar{y}_{jl}$, then the rest of the model decomposes into $|S|$ small size subproblems. The sum of the objective function values of all $|S|$ subproblems and the cost of the arbitrarily chosen transportation options $\bar{x}_{iq}$ and $\bar{y}_{jl}$ is the final objective function of the instance or sub-instance. We start with an S equal to 50 scenarios, then repeat the test by increasing S by an increment of 5, until we reach an acceptable maximum gap (1% or lower). *In-sample* stability testing for multiple instances is shown in Table 6.1.

Since having 50 scenarios results in a maximum *in-sample* gap of 0.41% for the instances we tested, and we determined that this is an acceptable level of gap, we conclude that our stochastic model with $S = 50$ scenarios achieves *in-sample* stability.

As mentioned in Section 2.4, in addition to *in-sample* stability, we need to ensure that the model has *out-sample* stability, which means the model gives approximately the same result if the "true" distribution is used to represent the uncertain parameters in the stochastic programming model. We assume that generating a scenario tree with 20,000 scenarios would represent the "true" distribution, as has been done in Lium et al. (2009).

For testing the *out-sample* stability of the model, we use some of the instances in Table 6.1 with $S = 50$ and compare their objective function value and that of their sub-instances to the objective function value of a sub-instance with a scenario tree that consists of 20,000 scenarios. To calculate the objective function value of the model with $S = 20,000$, we follow the same procedure as in *in-sample* stability testing, i.e. fixing x_{iq} and y_{jl} variables and solving $|S|$ subproblems. We report the maximum gap of the tested instances in Table 6.2.

Finally, we conclude from the results that $S = 50$ achieves both *in-sample* and *out-sample* stability for the stochastic programming model under study.

Instance No.	No. of Suppliers	No. of Customers	Demand Pairs	S	In- Sample Gap
1	10	10	50	50	0.41%
2	10	10	50	60	0.10%
3	10	10	50	70	$5.94e^{-5}$%
4	10	10	50	80	0.36%
5	10	10	50	90	$2.44e^{-5}$%
6	10	10	50	90	0.14%
7	10	20	100	50	0.24%
8	10	20	100	60	0.61%
9	10	20	100	70	0.48%
10	20	20	100	80	$7.31e^{-5}$%
11	20	20	300	50	$6.28e^{-5}$%
12	20	50	300	50	0.15%
13	20	50	500	50	$6.43e^{-5}$%
14	50	50	500	50	0.19%

Table 6.1: In-Sample Stability Testing

Instance No.	No. of Suppliers	No. of Customers	Demand Pairs	S	Out-Sample Gap
1	10	10	50	20,000	$1.02e^{-4}\%$
2	10	20	100	20,000	$1.86e^{-5}\%$
3	20	20	100	20,000	$1.42e^{-4}\%$

Table 6.2: Out-Sample Stability Testing

6.3 Numerical Testing

6.3.1 Valid Cut 1

We test the performance of valid cut 1 over multiple instances of varying sizes and compare it to the performance of CPLEX and the integer L-shaped method. We conduct some analysis on the effects of changing the penalty cost (π) and the number of scenarios in the MP ($\bar{S}$). Note that the instance number reflects its size. For example, instance number s5_c5_d20_sc20_1 is of size 5 suppliers, 5 customers, 20 supplier-customer demand pairs, and 20 scenarios.

Effects of Changing Penalty Cost and Number of Scenarios in MP

Numerical testing has been completed to study the effect of changing the penalty cost (π) as well as the number of scenarios maintained in the MP ($\bar{S}$) for valid cut 1, the L-shaped method and CPLEX. Table (6.3) summarizes the results of the testing for the case of partial decomposition with 10 scenarios maintained in the MP ($\bar{S} = 10$) for both the

L-shaped cut and valid cut 1. Also, the comparison includes varying the penalty cost of purchasing extra capacity to be in the between [100,1000].

It can be observed from Table (6.3) that valid cut 1 does not outperform the L-shaped cut for most cases. That is likely due to the fact that for each subproblem at each iteration, two IP's are solved for the valid cut 1 to be computed, while only one is solved for the integer L-shaped cut. To further analyze this, Table (6.4) also varies the penalty cost and the number of scenarios maintained in the master problem and shows the percentage of time that valid cut 1 is used in each case. Recall from Algorithm (3) that the computed value δ^s is only used if it is greater than the value $L^s - z^s$. In other words, if the value δ^s is less than or equal to the value $L^s - z^s$, the resulting cut would be the L-shaped cut. As a result, keeping track of the percentage of time that valid cut 1 is used in the algorithm could clarify why, in many instances, the algorithm performs worse than the L-shaped cut. It can also be noted from the results in Table (6.4) that as the number of scenarios maintained in the MP increases, the percentage of valid cut 1 used in the algorithm decreases. That can be justified by pointing out that as you maintain more scenarios in the MP, the MP starts off closer to the original problem than it would be with a smaller set of scenarios, and therefore, it would reach a good solution sooner.

Likewise, as the penalty cost (π) increases, the percentage of time valid cut 1 is used in the algorithm decreases. This is because naturally as the penalty cost of purchasing extra capacity increases, the model would prefer to use less extra capacity and open more transportation options and therefore most of the shipments would be allocated to first stage transportation options (x_{iq}, y_{jl}) .

Instance No.	Penalty Cost	CPLEX Gap	Valid Cut 1 Gap	L-Cut Gap
s5 c5 d20 sc20_1	100	0.00%	5.20%	3.05%
s5 c5 d20 sc20_1	300	0.00%	5.15%	3.66%
s5 c5 d20 sc20_1	500	0.00%	3.21%	3.16%
s5 c5 d20 sc20_1	1000	0.00%	0.07%	0.00%
s5 c5 d20 sc20_2	100	0.5%	0.00%	2.3%
s5 c5 d20 sc20_2	300	4.36%	0.27%	0.00%
s5 c5 d20 sc20_2	500	0.00%	0.00%	0.00%
s5 c5 d20 sc20_2	1000	0.00%	0.00%	0.00%
s10 c10 d50 sc50_1	100	21.88%	24.49%	13.53%
s10 c10 d50 sc50_1	300	13.17%	22.16%	21.10%
s10 c10 d50 sc50_1	500	16.53%	20.10%	21.12%
s10 c10 d50 sc50_1	1000	15.83%	14.10%	17.06

Table 6.3: Comparing Performance of CPLEX, Valid Cut 1 and L-Shaped Cut. Varying Penalty Cost (π) with 1 Hour Time Limit

6.3.2 Comparing the Performance of Valid Cuts

We test the performance of valid cuts 1, 2, 3A and 3B over multiple instances and compare it to the performance of CPLEX and the integer L-shaped method. Table 6.5 does some initial comparison of performance of valid cuts 1, 2, and 3A with CPLEX and the integer L-cut. Recall that solving two IP's for each subproblem at every iteration when computing the value δ^s in valid cut 1 slows down the algorithm and negatively affects its overall performance. Additionally, computing the maximum improvement $\delta^s_{iq}, \delta^s_{jl}$ in valid cut 2 independent of the other side of the network as well as relaxing the capacity restrictions

Instance No.	Penalty Cost (π)	Scenarios in MP ($\bar{S}$)	Gap Valid Cut 1	Percentage Used Valid Cut 1	Gap L-Cut
s5 c5 d20 sc20_1	100	2	14.28%	45.14%	12.90%
s5 c5 d20 sc20_1	100	5	10.00%	10.39%	9.57%
s5 c5 d20 sc20_1	100	10	5.20%	8.73%	3.05%
s5 c5 d20 sc20_1	300	2	5.15%	11.49%	5.41%
s5 c5 d20 sc20_1	300	5	6.94%	10.67%	3.66%
s5 c5 d20 sc20_1	300	10	7.87%	14.29%	9.16%
s5 c5 d20 sc20_1	500	2	7.76%	7.28%	4.08%
s5 c5 d20 sc20_1	500	5	8.39%	5.00%	4.23%
s5 c5 d20 sc20_1	500	10	3.21%	8.11%	3.16%
s5 c5 d20 sc20_1	1000	2	5.77%	5.45%	6.28%
s5 c5 d20 sc20_1	1000	5	0.07%	2.98%	0.00%
s5 c5 d20 sc20_1	1000	10	0.167%	0.00%	0.00%
s5 c5 d20 sc20_2	100	2	1.17%	11.39%	3.02%
s5 c5 d20 sc20_2	100	5	0.00%	0.95%	0.00%
s5 c5 d20 sc20_2	100	10	2.01%	3.95%	2.30%
s5 c5 d20 sc20_2	300	2	1.90%	7.32%	1.80%
s5 c5 d20 sc20_2	300	5	1.12%	0.00%	0.00%
s5 c5 d20 sc20_2	300	10	0.27%	0.17%	0.00%
s5 c5 d20 sc20_2	500	2	2.69%	0.91%	2.45%
s5 c5 d20 sc20_2	500	5	1.186%	0.03%	0.14%
s5 c5 d20 sc20_2	500	10	0.00%	0.00%	0.00%
s5 c5 d20 sc20_2	1000	2	0.00%	0.103%	0.00%
s5 c5 d20 sc20_2	1000	5	0.00%	0.00%	0.00%
s5 c5 d20 sc20_2	1000	10	0.00%	0.00%	0.00%
s10 c10 d50 sc50_1	100	2	50.52%	53.07%	45.98%
s10 c10 d50 sc50_1	100	5	42.52%	77.99%	38.61%
s10 c10 d50 sc50_1	100	10	24.49%	7.16%	13.53%
s10 c10 d50 sc50_1	300	2	69.59%	29.44%	40.28%
s10 c10 d50 sc50_1	300	5	34.00%	25.23%	28.19%
s10 c10 d50 sc50_1	300	10	22.16%	1.26%	21.10%
s10 c10 d50 sc50_1	500	2	34.93%	21.72%	27.05%
s10 c10 d50 sc50_1	500	5	21.57%	21.96%	29.23%
s10 c10 d50 sc50_1	500	10	20.10%	1.12%	21.12%
s10 c10 d50 sc50_1	1000	2	14.10%	5.09%	21.16%
s10 c10 d50 sc50_1	1000	5	16.78%	10.95%	17.05%
s10 c10 d50 sc50_1	1000	10	18.53%	0.00%	7.06%

Table 6.4: Comparing Performance of CPLEX, Valid Cut 1 and L-Shaped Cut. Varying Penalty Cost (π) and Number of Scenarios is MP ($\bar{S}$) with 1-Hour Time Limit

result in making the cut relatively weak. This can be noticed in the results of Table 6.5 as valid cut 3A outperforms valid cuts 1 and 2 for most cases.

Table 6.6, compares the performance of valid cuts 3A and 3A with 3B, to that of CPLEX and the L-cut with a 2-hour time limit. We notice that the addition of valid cut 3B to valid cut 3A results in better optimality gaps for some instances. However, when taking the average over multiple instances of the same size, we see that the addition of valid cut 3B does not always result in a better gap. Furthermore, the results also show that the performance of the proposed cuts 3A and 3B is somewhat comparable to that of CPLEX and the Integer L-cut, where in some instances valid cuts 3A and 3B result in a smaller gap in the given time limit, while for other instances, the CPLEX or L-cut outperform cuts 3A and 3B.

We can also infer from the results that cuts 3A and 3B show some promising research direction. Though they do not outperform the L-cut in all cases, this is due to the fact that at a single iteration for a specific subproblem, the number of optimality cuts of our proposed algorithm are a lot higher than the L-cut, which results in the algorithm going through a fewer number of iterations in the 2-hour time limit as compared to the L-cut algorithm. Additionally, the high number of cuts generated at every iteration results in making the MP harder to solve, which could make the algorithm more computationally demanding.

Instance No.	CPLEX Gap	Valid Cut 1 - Gap	Valid Cut 2 - Gap	Valid Cut 3A - Gap	L-Cut Gap
s5 c5 d20 sc5	0% (150 s)	0% (142 s)	0% (58 s)	0% (161 s)	0% (57 s)
s5 c5 d20 sc20_1	0.00%	3.21%	2.81%	2.57%	3.16%
s5 c5 d20 sc20_2	0.00% (3215 s)	0.00%	0.50%	0.40%	1.41%
s5 c5 d20 sc20_3	11.00%	11.55%	9.00%	5.35%	11.12%

Table 6.5: Comparing the Performance of CPLEX, Valid Cuts 1, 2, and 3A, and the L-Cut with 1-Hour Time Limit

In the next chapter, we conclude the findings of our research and propose some modifications of our problem for future research.

Instance No.	Gap CPLEX	Gap Valid Cut 3A	Gap Valid Cut 3A & 3B	Gap L-cut
s5 c5 d20 sc20_1 ($\bar{S} = 5$)	0.00%	3.09%	1.97%	3.94%
s5 c5 d20 sc20_2 ($\bar{S} = 5$)	0.00%	0.01%	0.07%	0.003%
s5 c5 d20 sc20_3 ($\bar{S} = 5$)	9.544%	0.44%	4.83%	8.09%
Average	3.18%	1.18%	2.29%	4.011%
s10 c10 d50 sc20_1 ($\bar{S} = 10$)	7.59%	12.28%	10.51%	11.86%
s10 c10 d50 sc20_2 ($\bar{S} = 10$)	13.77%	14.00%	13.62%	16.19%
s10 c10 d50 sc20_3 ($\bar{S} = 10$)	11.22%	7.73%	9.47%	7.72%
s10 c10 d50 sc20_4 ($\bar{S} = 10$)	6.80%	12.64%	8.58%	11.86%
s10 c10 d50 sc20_5 ($\bar{S} = 10$)	14.07%	14.00%	14.00%	15.89%
s10 c10 d50 sc20_6 ($\bar{S} = 10$)	10.05%	8.39%	9.47%	7.72%
s10 c10 d50 sc20_7 ($\bar{S} = 10$)	7.08%	8.20%	8.20%	9.41%
s10 c10 d50 sc20_8 ($\bar{S} = 10$)	10.77%	8.05%	8.02%	11.00%
Average	10.17%	10.66%	10.23%	11.46%
s10 c10 d50 sc50_1 ($\bar{S} = 20$)	11.39%	11.48%	11.34%	14.07%
s10 c10 d50 sc50_2($\bar{S} = 20$)	10.17%	14.91%	14.39%	14.79%
s10 c10 d50 sc50_3 ($\bar{S} = 20$)	16.75%	17.40%	16.72%	17.07%
s10 c10 d50 sc50_4 ($\bar{S} = 20$)	11.18%	12.87%	14.13%	14.26%
s10 c10 d50 sc50_5 ($\bar{S} = 20$)	20.82%	23.83%	24.33%	21.93%
Average	14.06%	16.10%	16.18%	16.42%

Table 6.6: Comparing the Performance of CPLEX, Valid Cuts 3A, 3B and the L-cut with 2-Hour Time Limit

Chapter 7

Conclusion and Future Work

This thesis studies a distribution planning problem with consolidation in a stochastic programming setting. The assumption of stochastic demand extends the deterministic problems in the literature to account for demand fluctuations. We study the problem from the point of view of a 3PL, who needs to decide on a set of transportation options for each supplier and customer in the network to reserve for a given period of time, subject to stochastic demand. We therefore formulate the problem as a two-stage stochastic programming model, where the choice of transportation options are the stage one variables. The second stage variables, on the other hand, represent the allocation of orders to available transportation options as well as purchasing extra capacity by shipping orders through a spot-market carrier, at a higher transportation cost.

Because of the high computational demand of the problem, we apply the integer L-shaped method to decompose it. To speed up the convergence of the algorithm, we apply three algorithm enhancement techniques. Additionally, we experiment with three valid cuts with the goal of generating stronger cuts than the L-cut, in order to increase the efficiency of the algorithm. The valid cuts are based on the idea of estimating the impact of making a transportation option available on the subproblem objective function value. We perform a number of numerical tests and compare the performance of our proposed algorithm with valid cuts, with that of CPLEX and the L-shaped cut. We focus on valid cuts 3A (Eq. 5.15) and 3B (Eq. 5.20) as they are shown to provide tighter gaps as compared to valid cuts 1 and 2 (Eq. 5.10 and 5.11 respectively). Numerical results suggest that the performance of our proposed algorithm with valid cuts 3A and 3B is comparable to that of CPLEX and the L-cut. Upon further investigation, we notice that even though valid cuts 3A and 3B are stronger than the L-cut, because of the high number of cuts the algorithm generates at every iteration, with a given time limit, our algorithm does not always outperform the L-shaped algorithm. This is because in a 2-hour time limit, the L-shaped algorithm goes through more iterations than our proposed algorithm with valid cuts 3A and 3B.

There is a number of possible future research directions. We notice that calculating the values of δ in our valid cuts, while relaxing capacity requirements, is contributing to the weakness of the cuts. Therefore, efficient greedy heuristics may be developed to calculate tighter values of δ that are close to the optimal values. Furthermore, additional numerical analysis may be conducted, while varying key parameter values, to better understand what is causing the problem to be hard to solve. Additionally, varying the penalty cost (π) to make it a function of the particular arrival/dispatch time q or l, and also allowing it to vary by scenario is an interesting estimate that accounts for natural fluctuations in spot market prices. Finally, limiting the number of transportation options available for each customer order based on the dispatch time from the consolidation center, i.e. creating some due date requirements for customer orders based on when the orders leave the consolidation center, is a reasonable assumption that would create sparsity in the problem. This, in turn, may reduce the computational difficulty of our problem.

References

W. M. Abdelwahab and M. Sargious. Freight rate structure and optimal shipment size in freight transportation. *The Logistics and Transportation Review*, 6(3):271–292, 1990.

G. Angulo, S. Ahmed, and S. S. Dey. Improving the integer l-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

O. Berman and Q. Wang. Inbound logistic planning: Minimizing transportation and inventory cost. *Transportation Science*, 40(3):287–299, 2006.

D. E. Blumenfeld, L. D. Burns, J. D. Diltz, and C. F. Daganzo. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research Part B: Methodological*, 19(5):361 – 380, 1985.

J. H. Bookbinder and J. K. Higginson. Probabilistic modeling of freight consolidation by private carriage. *Transportation Research Part E: Logistics and Transportation Review*, 38(5):305 – 318, 2002.

J. H. Bookbinder, Q. Cai, and Qi-M. He. Shipment consolidation by private carrier: The discrete time and discrete quantity case. *Stochastic Models*, 27(4):664–686, 2011.

L. D. Burns, R. W. Hall, D. E. Blumenfeld, and C. F. Daganzo. Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469–490, 1985.

Q. Cai, Qi-M. He, and J. H. Bookbinder. A tree-structured markovian model of the shipment consolidation process. *Stochastic Models*, 30(4):521–553, 2014.

D.J. Closs and R.L. Cook. Multi-stage transportation consolidation analysis using dynamic simulation. *International Journal of Pysical Distribtuion and Materials Management*, 17 (3):28–45, 1987.

M.C. Cooper. Freight consolidation and warehouse location strategies in physical distribution systems. *Journal of Business Logistics*, 4(2):53–74, 1983.

T. G. Crainic, M. Hewitt, and W. Rei. *Partial decomposition strategies for two-stage stochastic integer programs*. CIRRELT, 2014.

K. L. Croxton, B. Gendron, and T. L. Magnanti. Models and methods for merge-in-transit operations. *Transportation Science*, 37(1):1–22, 2003.

C. Daganzo. *Logistics Systems Analysis*. Springer, 1999. ISBN 9783540655336.

C. F. Daganzo. Shipment composition enhancement at a consolidation center. *Transportation Research Part B: Methodological*, 22(2):103 – 124, 1988.

J. Dupačová, G. Consigli, and S. W. Wallace. Scenarios for multistage stochastic programs. *Annals of operations research*, 100(1-4):25–53, 2000.

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

G. Guastaroba, M. G. Speranza, and D. Vigo. Intermediate facilities in freight transportation planning: A survey. *Transportation Science*, 2016.

Y.P. Gupta and P.K. Bagchi. Inbound freight consolidation under just-in-time procurement: application of clearing models. *Journal of Business Logistics*, 8(2):74–94, 1987.

K. Hyon. Ha, S.. Khasnabis, and G. Jackson. Impact of freight consolidation on logistics system performance. *Journal of Transportation Engineering*, 114(2):173–193, 1988.

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

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

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

K. Høyland and S. W. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, 2001.

K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, 24(2-3):169–185, 2003.

G.C. Jackson. Evaluating order consolidation strategies using simulation. *Journal of Business Logsitics*, 2(2):110–138, 1982.

M. Kaut. Scenario-tree generation: With michal kaut. In *Modeling with stochastic programming*, chapter 4, pages 77–102. Springer; New York, 2012.

M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. 2003a.

M. Kaut and S. W. Wallace. Evaluation of scenario-generation methods for stochastic programming. 2003b.

Wallace SW King AJ.

Y. E. Kılıç and U. R. Tuzkaya. A two-stage stochastic mixed-integer programming approach to physical distribution network design. *International Journal of Production Research*, 53(4):1291–1306, 2015.

G. Laporte and F. V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13(3):133 – 142, 1993.

Arnt-G. Lium, T. G. Crainic, and S. W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157, 2009.

M. A. Ülkü and J. H. Bookbinder. Modelling shipment consolidation and pricing decisions for a manufacturer-distributor. *Revenue Management*, 6(1):62–76, 2012a.

M. A. Ülkü and J. H. Bookbinder. Optimal quoting of delivery time by a third party logistics provider: The impact of shipment consolidation and temporal pricing schemes. *European Journal of Operational Research*, 221(1):110 – 117, 2012b.

T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

J.M. Masters. The effects of freight consolidation on customer service. *Journal of Business Logistics*, 2(1):55–74, 1980.

R. E. De Matta, V. N. Hsu, and Chung-L. Li. Coordinated production and delivery for an exporter. *IIE Transactions*, 47(4):373–391, 2015.

R. Musa, Jean-P. Arnaout, and H. Jung. Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*, 59(1):85 – 92, 2010.

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

Y. Pan, M. Zhou, Z. Chen, and H. Tan. A simulation optimization framework for shipment planning at rdc considering time and quantity consolidation with uncertain demands. In *Service Systems and Service Management (ICSSSM), 2011 8th International Conference on*, pages 1–6, June 2011.

J. Penuel, J. C. Smith, and Y. Yuan. An integer decomposition algorithm for solving a two-stage facility location problem with second-stage activation costs. *Naval Research Logistics (NRL)*, 57(5):391–402, 2010.

R.M. Russell and L. Krajewski. Optimal purchase and transportation cost lot sizing for a single item. *Decision Science*, 22:940–952, 1991.

A. Shapiro, D. Dentcheva, et al. *Lectures on stochastic programming: modeling and theory*, volume 16. Siam, 2014.

H. Song, V. N. Hsu, and R. K. Cheung. Distribution coordination between suppliers and customers with a consolidation center. *Operations Research*, 56(5):1264–1277, 2008.

T. Tan and O. Alp. Optimal sourcing from alternative capacitated suppliers with general cost structures. *Omega*, 58:26 – 32, 2016.

R. M Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4): 638–663, 1969.

Kai-L. Yung, J. Tang, A. W. H. Ip, and D. Wang. Heuristics for joint decisions in production, transportation, and order quantity. *Transportation Science*, 40(1):99–116, 2006.

G. Zhang and L. Ma. Optimal acquisition policy with quantity discounts and uncertain demands. *International Journal of Production Research*, 47(9):2409–2425, 2009.

Ju-l. Zhang and Ming-y. Zhang. Supplier selection and purchase problem with fixed cost and constrained order quantities under stochastic demand. *International Journal of Production Economics*, 129(1):1–7, 2011.

APPENDICES

Appendix A

Formulation 1 Numerical Testing

A.1 Solving Directly with CPLEX

Model 4.1 was tested on several problem instances of varying sizes. Table A.1 summarizes the results of running 10 instances of the size 10 customers, 10 suppliers, 50 demand pairs and 50 scenarios for a maximum of a 2-hour time limit. It compares the performance of the model when solved as-is with binary $\boldsymbol{u}$ and $\boldsymbol{w}$ variables, and when we relax $\boldsymbol{u}$ and $\boldsymbol{w}$ variables to be continuous and solve using default CPLEX settings, and computes the gap between objective function value of the original model with that of the relaxation.

Additionally, Table A.2 shows the result of solving the 10 instances for a time limit of 2 hours, with the original formulation, with the relaxed $\boldsymbol{u}$ and $\boldsymbol{w}$ variables using default CPLEX settings, and with the relaxed $\boldsymbol{u}$ and $\boldsymbol{w}$ variables using CPELX built-in Benders Strategy. It can be noted that CPLEX Benders Strategy outperforms the default settings for our problem for most of the instances.

Next we compare the performance of the original model with that of the relaxation

Instance No.	Total demand of all products	Incumbent Cont $\boldsymbol{u}, \boldsymbol{w}$	Gap	Incumbent Bin $\boldsymbol{u}, \boldsymbol{w}$	Gap	Gap Cont/Bin
1 (10,10,50)	109	$2.78e^5$	0% (112 s)	$2.81e^5$	0% (3929 s)	1.07%
2 (10,10,50)	127	$3.26e^5$	0.76%	$3.33e^5$	1.21%	2.1%
3 (10,10,50)	81	$2.66e^5$	0% (2,041 s)	$2.69e^5$	0%(335 s)	1.12%
4 (10,10,50)	99	$2.64e^5$	0% (919 s)	$2.71e^5$	0.17%	2.58%
5 (10,10,50)	110	$3.03e^5$	0% (43 s)	$3.04e^5$	0% (3902 s)	0.33%
6 (10,10,50)	109	$2.54e^5$	1.9%	$2.65e^5$	0.3%	4.15%
7 (10,10,50)	106	$3.01e^5$	0% (775 s)	$3.045e^5$	0% (3768 s)	1.16%
8 (10,10,50)	113	$2.84e^5$	0.78%	$2.912e^5$	0%(3865 s)	2.41%
9 (10,10,50)	92	$2.58e^5$	0% (34 s)	$2.594e^5$	0% (150 s)	0.54%
10 (10,10,50)	103	$2.87e^5$	1.44%	$2.935e^5$	1.5%	2.21%
11 (10,20,100)	232	$5.363e^5$	2.35%	$6.32e^5$	15.65%	15.19%
12 (10,20,100)	200	$4.98e^5$	4.28%	$5.386e^5$	9.13%	7.54%
13 (10,20,100)	198	$5.205e^5$	3.88%	$5.45e^5$	5.00%	4.50%

Table A.1: Performance of Original Model vs. Relaxation with Continuous $\boldsymbol{u}$ and $\boldsymbol{w}$ Variables with Default CPLEX Settings using CPLEX Benders Strategy in Table A.3 and calculate the percentage of $\boldsymbol{u}$ and $\boldsymbol{w}$ variables that have binary values in the solution of the relaxation. We notice that for all instances, most $\boldsymbol{u}$ and $\boldsymbol{w}$ variables end up having binary values in the solution which results in a small optimality gap between the relaxation and the original model. However, upon careful examination of those instances, it appears that for most suppliers and customers in the network, the total demand of all orders for a given supplier/customer is less than the capacity of a single transportation option and therefore resulting in the model choosing most u and w allocation variables close to 1 or 0 even when those variables are relaxed, and the optimality gap being small between the original model and the relaxation. This is, therefore, a special case of the model that occurred because the parameters used in the model were generated using the first proposed data generation method explained in Section 3.3, and for that reason, we adjusted the data generation method, as outlined in Section 6.1, for the computational testing of Model 4.2.

Thus, for the case when the total demand of a given supplier/customer exceeds the capacity of a single transportation option and multiple options are needed, solving the relaxed model with continuous $\boldsymbol{u}$ and $\boldsymbol{w}$ variables would result in $\boldsymbol{u}$ and $\boldsymbol{w}$ allocation variables being assigned fractional numbers, which in turn would increase the optimality gap between the relaxation and the original model.

Instance No.	Total Demand of All Products	Gap Binary $\boldsymbol{u}, \boldsymbol{w}$ (Default Options)	Gap Relaxed $\boldsymbol{u}, \boldsymbol{w}$ (Benders Strategy)	Gap Relaxed $\boldsymbol{u}, \boldsymbol{w}$ (Default Options)
1	109	0% (3929 s)	0% (35 s)	0% (112 s)
2	127	1.21%	0.45%	0.76%
3	81	0% (335 s)	0% (1,904 s)	0% (2,041 s)
4	99	0.17%	0% (477 s)	0% (919 s)
5	110	0% (3902 s)	0% (27 s)	0% (43 s)
6	109	0.3%	2.72%	1.9%
7	106	0% (3768 s)	0% (170 s)	0% (775 s)
8	113	0% (3865 s)	0% (5,257 s)	0.78%
9	92	0% (150 s)	0% (20 s)	0% (34 s)
10	103	1.5%	1.29%	1.44%

Table A.2: CPLEX Performance - Original Model vs. Relaxation with Continuous $\boldsymbol{u}$ and $\boldsymbol{w}$ Using Default CPLEX Settings and Benders Strategy

Instance No.	Total Demand of All Products	Incumbent Benders Strategy	Gap	Percentage Binary $\boldsymbol{u}, \boldsymbol{w}$	Incumbent (Binary $\boldsymbol{u}, \boldsymbol{w}$	Gap	Gap Cont/Bin
1	109	$2.785e^5$	0% (35 s)	99.98%	$2.81e^5$	0% (3929 s)	0.89%
2	127	$3.258e^5$	0.45%	99.98%	$3.33e^5$	1.21%	2.16%
3	81	$2.663e^5$	0% (1,904 s)	99.97%	$2.69e^5$	0% (335 s)	1.00%
4	99	$2.641e^5$	0 (477 s)	99.97%	$2.71e^5$	0.17%	2.55%
5	110	$3.03e^5$	0% (27 s)	99.98%	$3.04e^5$	0% (3902s)	0.33%
6	109	$2.550e^5$	2.72%	99.94%	$2.65e^5$	0.3%	0.39%
7	106	$3.010e^5$	0% (170 s)	99.97%	$3.045e^5$	0% (3768 s)	0%
8	113	$2.840e^5$	0% (5,257 s)	99.98%	$2.912e^5$	0%(3865 s)	0%
9	92	$2.585e^5$	0% (20 s)	99.97%	$2.594e^5$	0% (150 s)	0%
10	103	$2.869e^5$	1.29%	99.94%	$2.935e^5$	1.5%	0.034%

Table A.3: Comparing the Performance of CPLEX Automated Benders Decomposition with Relaxed $\boldsymbol{u}$ and $\boldsymbol{w}$ Variables to CPLEX Default Settings with Binary $\boldsymbol{u}, \boldsymbol{w}$