An Inventory Routing Problem with Cross-Docking

by

Forough Pourhossein

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, 2014

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

This thesis concerns inventory-transportation tradeoffs in which a number of suppliers serve multiple customers, each ordering several product types. The goal is to design optimal routes to satisfy the customers' demands. In the proposed approach, the products are shipped to cross-docks from the suppliers, and several customers will be served by each route beginning at a cross-dock. The objective is to minimize the total cost, beginning with summing the transportation costs on those edges through which trips may go, times the shipment frequencies. The holding costs at customers, and the pipeline inventory costs on the routes, take into account that various products may have different carrying-cost parameters. Based on some analytical results, the developed model is reformulated in terms of a single set of decision variables.

The holding cost makes the objective function highly nonlinear. In addition, transportation cost and pipeline inventory cost are quadratic. After linearization of the objective function, a column generation algorithm is proposed to solve the nonlinear mixed-integer programming model. The holding cost, which is the sum of a set of fractions, is linearized after objective-function decomposition. Each of the decomposed sub-problems has only one fraction, which can be linearized by replacing that fraction by a new decision variable and adding some constraints to the formulation. To linearize the quadratic parts of the objective function, we substitute a new variable for the multiplication of each pair of decision variables, and add some new constraints.

We provide computational results for the model with a single product. All parameters are generated randomly. Our proposed algorithm can optimally solve some problems with up to 626 edges. However, CPU time might be very high. For instances with 500 edges, CPU time can be up to 20 hours depending on the number of iterations the algorithm needs to find the optimal solution. Instances with up to 300 edges are solved to optimality within a CPU time of only one hour on a computer with 16 GB RAM and 3.40 GHz CPU.

Acknowledgments

Foremost, I would like to express my greatest indebtedness to my supervisors, Prof. James Bookbinder and Prof. Hossein Abouee Mehrizi, for their continuous support of my study and research. I appreciate all their contributions of time, ideas and funding as well as their guidance while allowing me to be independent with my research. They made my MASc. experience productive and exciting.

Prof. Bookbinder, thank you for offering valuable advice, and sharing your vast knowledge and experience. Your support and warm encouragement during the whole period of the study, and especially your indescribable patience and guidance during the writing process, deserve an everlasting appreciation.

Prof. Abouee, I truly admire your patience, positive attitude, enthusiasm, and immense knowledge. Thanks a lot for the insightful and valuable discussions, continuous encouragement and constructive suggestions. I have learned so much from you. It is an honor for me to be your first student to graduate!

I would like to thank the members of the reading committee, Prof. Benny Mantin and Prof. Stan Dimitrov for their helpful comments and suggestions on my thesis.

My fellow officemates, Tiffany, Nidarsha, Saul and Meisam, thank you so much for inspiring discussions and being supportive in every way.

Ata, Amin, Bahar and Majid, thanks for those sleepless nights we were working together before deadlines, and for all the fun we have had in the past two years of movies and popcorn, Xbox, board games and game cards. I know there will be many more chances to make great memories in the future. Gizem, Ru, I will never forget all fun and laughter we had in your office.

I have very special friends to thank, for their continued friendship since middle school. Mona, Faegheh, your support and encouragement have always been so heartwarming for me. Distance couldn't stop us from having long fun talks almost every week! You were always there through the good times and bad, cheering me up. I love you.

I warmly thank my biggest supporters, my mom and dad for their unconditional love and encouragement. Even though we are thousands of miles away, they were always there to cherish with me every wonderful moment of my life, and supported me whenever I needed them. Mom, I love you so much. Dad, I will always make you proud. I also would like to thank my lovely sister and dearest brother, Parisa and Amir, for their endless love and prayers for me.

I finish with my loving husband, Mahdi, whom I believe I could put in every single category of my acknowledgments! Darling, thank you for making me smile and for standing by me in every moment of my life. With you, I look forward to every day. You are incredibly supportive. You gave me love, hope and courage whenever I felt it was impossible to continue. Thanks for staying with me at school very late at night. Thanks for delicious dishes you cooked for me when I was busy with my studies. I owe you lots of fun, love, hugs, *fesenjun* and *khoresh karafs*. I love you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Background

In every supply chain, there is a group of customers who demand a variety of products from a set of suppliers. For example, consider a series of franchise convenience stores. Those stores are the demand points; the suppliers provide several types of products for them. Each of these convenience stores should carry a certain level of inventory relative to the demand of each product. The total cost of the supply chain has always been a major concern. Companies place great emphasis on the shipping costs from suppliers to customers and the inventory carrying costs at the stores, hence those are the most important costs that researchers take into account.

The shipping costs include transportation costs and the in-transit inventory carrying cost (also called pipeline inventory cost). Because some products need to be refrigerated during transit, pipeline inventory cost should be considered in addition to vehicle costs such as the fuel, driver's wage and the vehicle depreciation.

Inventory level at the stores should be considered in light of total cost. Thus, low stock levels result in low carrying costs. However, to secure the demand satisfaction at the stores while keeping inventory levels down, the deliveries of products must become more regular. Higher frequencies of deliveries increase the transportation cost. Therefore, the decisions on inventory management and transportation strategies should be integrated. There must be a tradeoff between the inventory levels and the replenishment frequencies, so that the total cost of the supply chain is minimized.

Various delivery strategies can have a significant influence on shipment costs. For example, 7-Eleven Japan and Wal-Mart use cross-docking to reduce transportation costs. Shipments coming from different suppliers are aggregated at the cross-docks (CDs), and mixed loads are then delivered to individual stores. Products for different customers that are provided by the same supplier can be shipped to a CD and aggregated there. These strategies help the supply chain to dispatch near-truckload shipments most of the time, as opposed to having less-than-truckload (LTL) ones. Therefore, if the demands are usually LTL, the shipping costs decrease significantly.

Vehicle routing is another useful transportation strategy. That is especially true when most shipments to each customer are LTL and the suppliers are far from the customers. In this case, demands of a sub-set of customers are consolidated at a supplier; the vehicle visits all those customers, delivers their demands, and goes back to the supplier.

1.2 Objectives

There is much research in the areas of cross-docking, vehicle routing and inventory management. However, there is so far no published research that considers those three topics together. We are interested in formulating and solving a model in which vehicle routing and cross-docking can be considered simultaneously as transportation strategies. In addition, our goal is to minimize the supply chain's total inventory costs plus transportation costs. We note that there is no paper in the literature addressing pipeline inventory in vehicle routing problems; the present work takes this into account as well.

The shipment frequencies for each route will be considered as decision variables, allowing the sum of demands on a route to exceed the vehicle capacity. This idea was first introduced by Berman and Wang (2006) for a supply chain with direct shipment strategy. However, it has not yet been applied to routing models.

The remainder of the thesis is organized as follows: Chapter 2 is a literature survey on cross-docking, vehicle routing and inventory-routing models. In Chapter 3, the problem definition and the model formulation are provided. Chapter 4 addresses the model reformulation and the solution method. In Chapter 5, we present computational results and some sensitivity analyses. Chapter 6 considers the conclusion of the thesis, and suggestions for future work.

Chapter 2

Literature Review

2.1 Vehicle Routing Problems (VRP)

The classical VRP consists of planning optimal delivery or pick-up routes for a set of customers, originating from a depot. Through imposing different constraints, such as time windows, vehicle capacity and travel time, various problems in vehicle routing can be defined. We introduce some famous ones and then provide a literature review for them.

The problem was first introduced by Dantzig and Ramser (1959) in which a gasoline truck delivery is to be optimized by designing routes from a bulk terminal to a large set of service stations supplied by the terminal. The demands for multiple products are known; the goal is to assign a number of stations to each truck, minimizing the total travel mileage, such that all demands are satisfied. They provide a linear programming formulation for the problem. They introduce their model as a generalization of the Traveling Salesman Problem (TSP).

Capacitated VRP (CVRP) is a vehicle routing problem in which the only constraint is on the capacity of the vehicle. CVRP is definitely harder to solve than the TSP. Instances with up to thousands of vertices are solvable for the TSP. However, the best algorithms can generally solve the CVRP for at most 100 customers (Cordeau et al. 2007). VRP with Time Windows (VRPTW) is another variant of the vehicle routing problem. The customers can be served only within a specific time interval, and the trips are to be scheduled. However, researchers have been working on more complex problems that are considered as "rich" VRP. Various assumptions such as considering multiple depots, several routes per vehicle, a variety of products, and non-identical vehicle types help to define the types of rich VRP.

2.1.1 VRP Solved with Exact Algorithms

As a literature review, we now describe some recent CVRP models solved using exact methods, especially column generation and branch-and-price. For a review of previous exact methods in CVRP, see Laporte and Nobert (1987) and Cordeau et al. (2007).

Mingozzi et al. (2013) present a multi-trip VRP in which their goal is to schedule several trips for each vehicle to minimize the total number of vehicles used in a period. This is applicable when the period is large enough compared to the travel times. The capacity of each vehicle is Q and the corresponding maximum driving time is T. The sum of demands of the customers on each route should not exceed the capacity of the vehicle. A schedule of a vehicle is defined as a set of routes assigned to that vehicle, such that the total travel times of those routes is less than or equal to T.

Mingozzi et al. (2013) provide exact algorithms to solve the problem, and state that no exact algorithm has been proposed for this problem so far. They develop two set-partitioning formulations; one is based on producing all feasible routes, and the other is based on producing all feasible schedules. They introduce a two-phase route-generation algorithm to solve the pricing problem corresponding to the first formulation, and another algorithm as a schedule generator to solve the pricing problem of the second formulation. In their computational results, they show that their algorithms can solve 42 out of 52 instances with up to 120 customers, examples used in testing heuristic algorithms in the literature.

Muter et al. (2014) develop a column generation algorithm for a multi-depot VRP (MDVRP) in which going through intermediate depots is allowed so that the vehicles may replenish. For instance, considering an ordered set of vertices on a route as $(S_1, S_2, \dots, S_n)$, S_1 and S_n in MDVRP refer to the same depot, whereas in their model, S_1 and S_n can be two different depots. The vehicles can thus stop at S_n and replenish, and then continue serving another route. Therefore, Muter et al. (2014) define rotation as a sequence of ordered indices $(S_1, S_2, \dots, S_N)$ for each vehicle, in which S_1 and S_N correspond to a given depot, and the remaining vertices can be either customers or depots. A rotation is feasible if the total travel time is less than or equal to the maximum travel time allowed, and the sum of demands on each route of the rotation does not exceed the vehicle capacity. In their column generation algorithm, the authors formulate the problem as a set-covering model in which the decision variables are the rotations. They provide two pricing problems to generate the rotations. For the first, Muter et al. (2014) use the approach of the Elementary Shortest Path Problem with Resource Constraints, employing the label-correcting algorithm to solve. In the second approach, the authors decompose each rotation into a set of routes to model the pricing problem and provide a two-phase method to solve it. The solution of the column generation algorithm is an LP relaxation, so they employ branch-and-price to find the optimal integer solution. Muter et al. (2014) can provide exact solutions for instances with up to 50 customers.

2.1.2 VRP Solved with Heuristic Methods

Many heuristics have been presented for the VRP. These can be categorized into route construction heuristics, two-phase methods and route improvement methods (Laporte and Semet, 2002). For the route construction approach, we can point to the savings algorithm proposed by Clarke and Wright (1964), in which cost reduction is achieved by connecting two customers to each other in the same route instead of considering them in two separate routes. The savings method is fast but it has a poor performance, i.e., at the beginning, it generates good routes, but the routes get less interesting towards the end. Other researchers improved the performance of the algorithm, however with higher computational times (See Cordeau et al., 2007).

The two-phase method decomposes the VRP into two sub-problems: clustering and routing. As for clustering, there are several algorithms proposed such as the Sweep algorithm. Then, the routing phase is treated as a TSP.

For route improvement heuristics, local search is often used to improve the solutions of other heuristic methods (See Cordeau et al., 2007). Metaheuristics such as simulated annealing, genetic algorithm and neural networks have also been applied to the VRP. The metaheuristic ideas help to build powerful heuristics (See Cordeau and Laporte, 2004).

2.2 Inventory Routing Problems (IRP)

The IRP considers inventory management and vehicle routing simultaneously. The inventory routing problem aims to deliver products from a supplier to a group of customers on several routes, with a set of side constraints. Some decisions are to be made by the supplier, e.g., when to serve each customer, how to build the routes, etc. to minimize the total cost of the system. Researchers have been working on the IRP for more than thirty years. Before this period, there were many publications in the area of inventory management and VRP separately. However, due to the limited computing power and lack of proper algorithms to solve large and complex combinatorial problems, the IRP appeared too difficult to handle (Coelho et al. 2014).

The IRP can be classified based on inventory policy, structure of the supply chain or the time horizon. In addition, information about the customers' demand can be known or unknown. If the demand information is available from the beginning of the horizon, or the probability distribution of demand is available, then the demand information is classified as "known". Otherwise, the "Dynamic IRP" should be considered when the demand information is not available in advance (Coelho et al. 2014).

We next provide a literature review of some articles relevant to our research. See Andersson et al. (2010) and Coelho et al. (2014) for a more thorough literature review on the IRP.

Archetti et al. (2007) propose the first branch-and-cut algorithm for IRP with one vehicle. They develop an IRP in which a product is to be shipped from a supplier to a set of customers within a time horizon. The supplier has to monitor the level of inventory at the customers, and replenish their stock such that the inventory level reaches its maximum allowed. The supplier guarantees that there will be no stock-outs at the customers. The customers themselves define their maximum inventory level. Therefore, this problem has a vendor-managed, order-up-to-level inventory policy. The demands are known, and the vehicle has a certain capacity. The goal is to determine the quantity of shipments in each time period, and to design the routes. Archetti et al.

(2007) present a mixed integer linear programming model, and solve it by adding some valid inequalities to the LP relaxed model, then applying an exact branch-and-cut algorithm.

In their branch-and-cut algorithm, the authors first relax the sub-tour elimination constraints and add some of the defined valid inequalities. At each node, they call the separation algorithm presented by Padberg and Rinaldi (1991). Whenever the sub-tour elimination constraint is violated, the corresponding constraint is added to that sub-problem. Otherwise, they branch based on the values of the variables. Archetti et al. (2007) solve instances with up to 50 customers when the time horizon is 3, and up to 30 customers when the time horizon is 6.

Coelho and Laporte (2013) extend the model of Archetti et al. (2007) by considering multiple vehicles. They also propose a branch-and-cut algorithm to solve the model.

2.3 Cross-Docking

Cross-docking involves the receipt of goods from suppliers, and preparing these items for shipment to retailers within a short time with no storage. Cross-docking reduces the logistics costs and may provide more customer satisfaction. Gumus and Bookbinder (2004) introduce several approaches to cross-docking. They consider a network with a group of suppliers, a set of customers with known demands, and the sites of potential CDs. Decisions are to be made on the locations of the CDs, the numbers of trucks to be used both for direct shipment and through CDs, and on shipment consolidation. Gumus and Bookbinder (2004) propose a model for a sole supplier serving a single product to multiple customers. The authors provide some observations, such as on the efficiency of sending near-truckload shipments directly compared to cross-docking. In addition, they observe that after subtracting the truckload shipments from the total demand of a single customer, the entire remainder should be sent (whether through CD or directly) without splitting into more than one shipment.

After solving the problem, Gumus and Bookbinder (2004) find that the demands are mostly consolidated and shipped through CDs if the fixed cost per truck is high, and the direct shipment is preferable only for near-truckload shipments. The authors also generalize their proposed model to multiple products, and provide a heuristic considering possible consolidation by product for the given customer. Then, they consider multiple manufacturers in their model, where each manufacturer can consolidate the demands of several customers, and each CD consolidates the demands on various products of particular customers.

For a detailed literature survey on cross-docking, see the review articles of Stephan and Boysen, (2011), Van Belle et al. (2012) and Buijs et al. (2014). We next provide a literature review on articles related to our work in terms of cross-docking.

Berman and Wang (2006) integrate transportation and inventory management to reduce the supply chain costs significantly. They consider a supply chain with a set of suppliers and a group of plants with demands on multiple products. In this supply chain, the total transportation cost, inventory carrying cost at the plants and the pipeline inventory cost is to be minimized. Also, the best distribution strategy (either direct shipment or cross-docking) is to be determined. The authors consider the frequencies of shipment as decision variables, too. Berman and Wang (2006) provide a heuristic and a branch-and-bound algorithm to solve the highly nonlinear mixed-integer programing model. Their algorithms are based on Lagrangean relaxation (LR). They also provide a greedy heuristic to find an initial feasible solution and also a proper upperbound to the problem.

These authors make some assumptions about the frequencies of shipment and the quantities of products so that the problem is solvable:

1) The quantities of products in a shipment do not have to be integer.

2) The shipment frequencies also can be any number and not necessarily an integer.

3) Products are always available at the suppliers.

4) The inbound-outbound coordination at the CDs is not considered.

5) All the quantities in a flow are shipped by the same transportation strategy, whether direct shipment or cross-docking.

6) When loading a truck, only the volume of products is considered. Transportation costs do not depend on weight.

In this model, the periods are identical. The transportation time parameters are in fact the ratios of transportation times to the length of the time horizon. Berman and Wang (2006) define one set of binary decision variables. Each variable is equal to 1 if the corresponding flow of product p from supplier i to plant j is going to be sent through CD k ($k = 0$ if it is a direct shipment). To write the objective function, the authors first formulate the frequencies of shipment. Then, the transportation cost is the sum of all transportation costs of each flow multiplied by the corresponding shipment frequency. The inventory cost at each customer is the total holding cost of that customer divided by its shipment frequency.

The only constraint that they have is on demand satisfaction. Those authors relax the constraint using LR, and then the LR formulation becomes decomposable. After decomposition, they come up with some analytical results, and using the results of those, Berman and Wang

(2006) develop a branch-and-bound algorithm to solve the sub-problems. They also provide a Lagrangean heuristic. They provide some numerical results and show that their Lagrangean heuristic is very fast and the lower bound is tight. The gap between the Lagrangean heuristic solution and the optimal solution is less than 1%.

Abouee-Mehrizi et al. (2014) propose a column generation algorithm to solve a two-echelon model consisting of a number of suppliers, a group of capacitated CDs and a set of customers. The number and the locations of the capacitated CDs are to be determined, and the demands of customers must be satisfied by minimizing the pipeline inventory cost, inventory cost at the customers and the transportation cost. The proposed model is a nonlinear mixed integer program. The authors show that the "Capacitated Plant Fixed-Charge Transport Location Problem" is a special case of their problem. To solve it, they show that their model can be written as a cutting stock problem. The authors also solve the problem when the vehicle capacities are decision variables. They provide some numerical results to show the efficiency of their algorithm. As in Berman and Wang (2006), Abouee-Mehrizi et al. (2014) also consider the frequencies of shipment. In the column generation algorithm provided by Abouee-Mehrizi et al. (2014), the CD capacity constraints are relaxed using Lagrangean relaxation. It is shown that the structure of the pricing problem is similar to the one in Berman and Wang (2006), except that the latter authors dealt with binary variables while Abouee-Mehrizi et al. (2014) are dealing with integer variables. Since the sub-problem is a bounded integer program, they transform it into a binary integer program. Therefore, those authors use the branch-and-bound algorithm of Berman and Wang (2006) and find it efficient for their model, as well.

2.4 Vehicle Routing with Cross-Docking

Wen et al. (2009) introduce the VRP with cross-docking (VRPCD) in which the suppliers ship products to customers through a CD. Their objective is to minimize the total travel time, considering time window constraints for the customers. They propose a mixed integer programming model and develop a solution algorithm based on Tabu Search. The authors test their model on data provided by Danish Consultancy Transvision involving up to 200 arcs. They show that the gap of the model is less than 5%. Wen et al. (2009) consider the loading and unloading of trucks at the CD as decision variables, so that they can consider direct shipment, too.

Santos et al. (2011) reformulate the VRPCD and propose a branch-and-price algorithm to solve the model. They state that they are the first authors that provide an exact algorithm for VRPCD. Those authors also include the cost of changing loads at the CD in their objective function. They formulate the pricing problem as a Resource Constrained Elementary Shortest Path Problem. To solve the pricing problem, they use dynamic programming.

Baldacci et al. (2013) present an exact algorithm for the two-echelon CVRP (2E-CVRP). They consider intermediate capacitated depots called “satellites,” in which shipments are managed from a depot to customers. Although they name the intermediate depots as satellites, their application seems to be more like cross-docking, because they consider only a handling cost, proportional to the quantities loaded or unloaded at the satellites. Therefore, it is worth mentioning their article in this section. The customers are to be visited exactly once. However, the satellites can be visited by more than one route originating from the depot.

Baldacci et al. (2013) introduce a new formulation and apply both integer and continuous relaxations to it. They use dynamic programming to find better bounds to the problem. They also decompose the 2E-CVRP into a set of "Multi-depot Capacitated Vehicle Routing Problems" with side constraints. 207 instances are run with up to 100 customers and 6 satellites.

In the next chapter we will provide the problem definition and the formulation of our problem. We consider an IRP model with cross-docking (IRPCD), considering frequencies of shipment. To the best of our knowledge, there is no article in the literature addressing the IRPCD and shipment frequencies for the VRP.

Chapter 3

Problem Formulation

3.1 Problem Definition

We consider a number of suppliers who serve multiple customers, each ordering various product types. The goal is to design optimal routes to satisfy the customers' demands, routed through one or more CDs. Multiple customers will be served by each route originating from CDs after consolidating the shipments that arrived from several suppliers. Shipments from suppliers to the CDs are assumed to be direct.

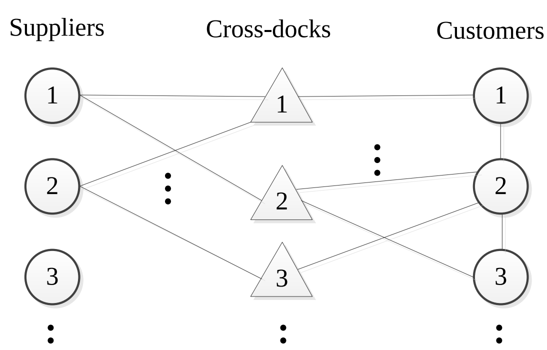

Figure 1 – Distribution strategy

Apart from designing optimal routes with minimum costs, we are also interested in finding the optimal frequencies of shipment on each route. The vehicles have a finite capacity, and the total demands delivered on each route are allowed to be greater than the vehicle capacity; in that case, the vehicle assigned to that route has to deliver those demands in more than one trip to customers. Therefore, the frequencies of shipments are also treated as decision variables. This applies as well to the trips between suppliers and CDs, hence those frequencies of shipment are to be determined.

Berman and Wang (2006) ship multiple products through CDs, considering frequencies of shipment, but to a single customer at a time. Moreover, they assume that the supplier-product combinations are decided in advance for each customer. Abouee-Mehrizi et al. (2014) find optimal shipment frequencies in a supply chain with capacitated CDs. However, they too consider only direct shipments. To the best of our knowledge, there is no article aimed at shipment frequencies in the context of vehicle routing.

The total cost in our model consists of five parts: transportation cost from CDs to the customers (TC), pipeline inventory cost from CDs to customers (PC), inventory holding cost at the customers (HC), transportation cost from suppliers to the CDs (TC'), and pipeline inventory cost from suppliers to CDs (PC').

No publication in the literature addresses pipeline inventory in VRP. Indeed, calculation of the pipeline inventory cost requires that the routes be designed in advance: the aggregate demands of all the customers on a specific route must be known to determine how much holding cost these demands incur on that route; the sequence of customers visited must be known because those are the edges on which each demand travels to get to the corresponding customer.

This preceding limitation of pipeline inventory cost makes the modeling quite challenging. That is why we put an assumption here: indices of customers always increase in the direction of the route, meaning that going from one customer to another is allowed only if the index of the first customer is smaller than the latter.

The above assumption can be interpreted in another way. We can say the cost matrix is such that in the upper triangular portion, each element is the cost of traveling over the corresponding edge, but in the lower-triangular parts, all the elements are infinite. That assumption can be operationalized by applying a heuristic to sort the customers in advance, to get a solution that is close to optimal.

As a summary, the goal of our model is to minimize the sum of total cost by designing proper routes and finding optimal shipment frequencies, satisfying all the customers' demands.

We assume that products are always available at the suppliers. In addition, there are no capacity constraints for the CDs. All products of the same type to be shipped to a specific customer are assumed to be delivered through a single CD and on a single route. Two types of vehicles are available, one for the trips from suppliers to CDs, and one for routes originating from the CDs.

Similar to Berman and Wang (2006), when loading a vehicle, the product volumes are considered. Based on vehicle capacity, the sum of the demands for a route is divided by the vehicle capacity to find the shipment frequency. Therefore, the product units are "infinitely splittable", meaning that the quantity of product in a shipment does not have to be integer. In addition, the shipment frequency can be any number, not necessarily integer.

Our simplifying assumptions could mean that the model might not apply to some real systems. However, solution of our model can approximate the costs of those real systems and the corresponding distribution strategies, and provide insights on real world systems.

3.2 Model Formulation

3.2.1 Notation

Let S,G, I and P be the sets of available suppliers, CDs, customers and products respectively. Table 1 introduces the notation used in the model formulation. More than one route can originate from a particular CD. Thus, to be able to write a separate cost function for each route, we establish that each CD can be referred by a set of "dummy CDs", such that each dummy CD serves at most one route. Therefore, we define set K as the set of dummy CDs for CD g. This helps us to deal with each route separately, in terms of design and relative costs. As a result, the process of decomposition of the objective function will be possible. This will be elaborated later.

The maximum number of routes that can be assigned to a CD is equal to $|I| \times |P|$ because there are $|I|$ customers in the system, and each of them can potentially order at most $|P|$ types of products. For the worst case, each customer would be served by its own direct shipment, and each product would be sent on its own separate route. Hence, the maximum number of dummy CDs equals $|I| \times |P|$.

In fact, each dummy CD refers to one route. However, in the next chapter, we show that the number of dummy CDs is not a concern for us as we decompose the objective function over g and k. Also let $\bar{I} = I \cup G$, which is the set of all customers and CDs together. When writing the equations, we assume the indices of the CDs are smaller than the indices of the customers in $\bar{I}$.

As an example, concerning the CDs and customers as a sub-system, consider two CDs with three customers and a single product. We will have three dummy CDs per CD. In Fig. 2, the first customer (Customer 4) is served by CD 1, and two other customers are served by CD 2. Note that in this figure, dummy CD $i - j$ refers to the jth dummy of CD i.

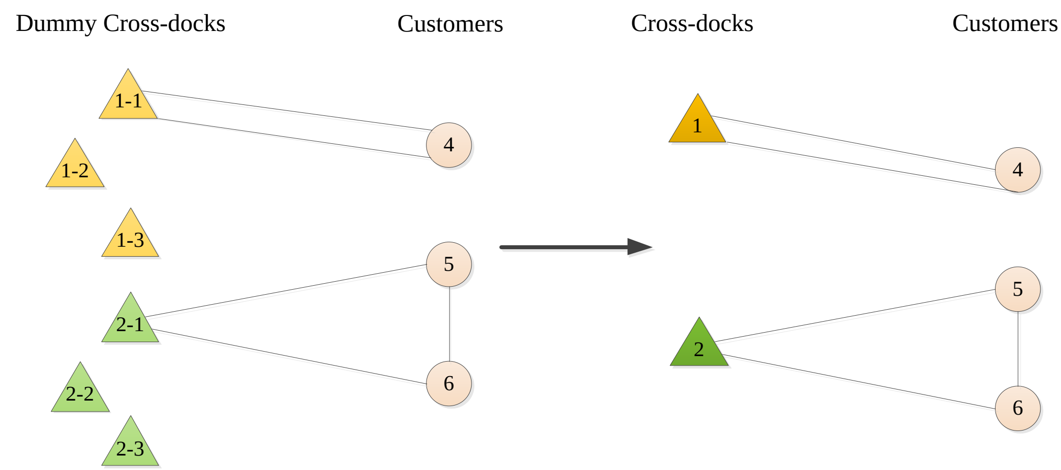

Figure 2 – Routes assigned to CDs

Also in Fig. 3, all customers are served by direct shipments originating from CDs. The first two customers are served by CD 1 and the third one by CD 2.

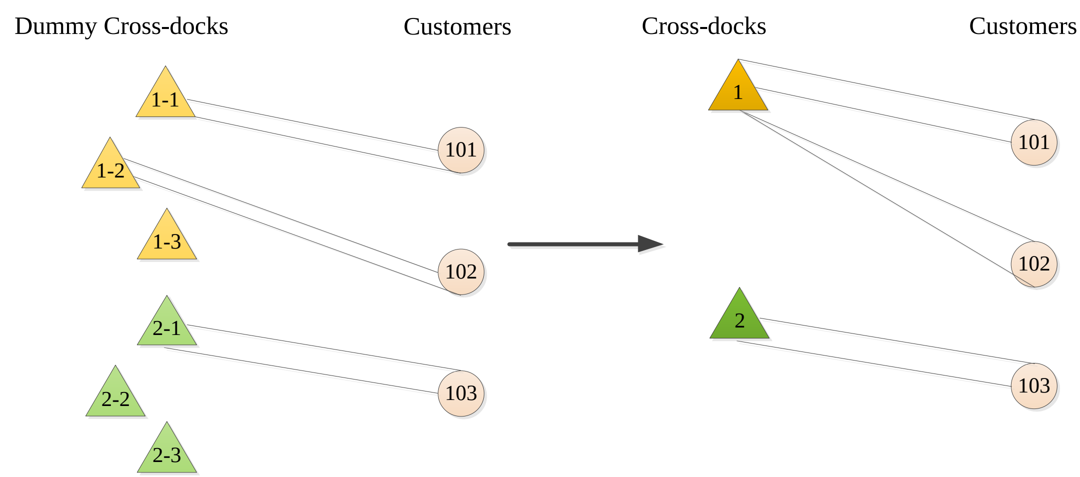

Figure 3 – Routes assigned to CDs

Notation	Description
C_T	Capacity of the vehicles travelling from CDs to customers
C'_T	Capacity of the vehicles travelling from suppliers to CDs
b_p	Volume of a vehicle occupied by one unit of product $p, p \in P$
d_{ip}	Demand of customer i for product $p, i \in I, p \in P$
c_{ij}	Transportation cost of a trip on edge $(i, j), (i \in \bar{I}, j \in I, i < j)$
c'_{sg}	Transportation cost of a trip on edge $(s, g), (s \in S, g \in G)$
h_p	Cost to hold one unit of product p over the time horizon (one period)
t_{ij}	Travel time (periods) on edge $(i, j), (i \in \bar{I}, j \in I, i < j)$
t'_{sg}	Travel time (periods) on edge $(s, g), (s \in S, g \in G)$
f_{gk}	Frequency of shipment on kth route of CD g, $(k \in K, g \in G)$
f'_{sg}	Frequency of shipment on edge $(s, g), (s \in S, g \in G)$

Table 1 – Notation

3.2.2 Decision Variables

Our formulation of this model involves four sets of variables defined as:

$$y_{sip}^{gk} = \begin{cases} 1 & \text{if customer } i \text{ gets product } p \text{ from supplier } s \text{ through CD } g, \text{ on the } k\text{th route} \\ 0 & \text{otherwise} \end{cases} \qquad (i \in I)$$

$$y_{sip}^{gk} = \begin{cases} 1 & \text{if dummy CD } k \ (k\text{th route}) \text{of CD } g \text{ is operating to deliver product } p \text{ from supplier } s \\ 0 & \text{otherwise} \end{cases} \qquad (i \in G)$$

$$x_{ijp}^{gk} = \begin{cases} 1 & \text{if edge } (i, j) \text{is a part of route } k \text{ related to CD } g, \text{delivering product } p \\ 0 & \text{otherwise} \end{cases} \qquad (i, j \in I)$$

$$x_{gjp}^{gk} = \begin{cases} 1 & \text{if edge } (g, j) \text{is a part of the route related to dummy} \\ & \text{CD } k \text{ containing more than one customer on the} \\ & \text{route to deliver product } p \\ 2 & \text{if only the single customer } j \text{ is served by CD } (g, k) \\ & \text{to deliver product } p \\ 0 & \text{otherwise} \end{cases} \qquad (j \in I)$$

We also need to define a variable as x_{ij0}^{gk} , which shows whether the edge (i, j) is going to be used or not:

$$x_{ij0}^{gk} = \max\{x_{ijp}^{gk} : p \in P\}$$

The frequency of shipment from a CD to a set of customers is calculated as below:

$$f_{gk} = \frac{\sum_{j \in I} \sum_{p \in P} \sum_{s \in S} b_p d_{jp} y_{sjp}^{gk}}{C_T} \qquad (1)$$

This means that the frequency of trips through CD g on route k is equal to sum of the demands of customers' volume on all products that are going to be served by CD g on route k, from all suppliers, divided by the vehicle capacity.

3.2.3 Cost Function

Now that we have defined variables and parameters, we can write the cost function. This consists of five different parts, and is formulated as follows:

3.2.3.1 Formulation of Transportation Cost from CDs to Customers

TC is computed by adding up the costs of the edges through which the trips have gone, times the frequencies of those shipments for all CDs.

As mentioned before, x_{ij0}^{gk} shows whether the edge (i, j) is going to be used or not. If more than one product is assigned to a given route k, the corresponding volumes of those products will be summed up in the frequency equation; taking the maximum of x_{ijp}^{gk} shows that this specific edge is going to be traversed on only one route, i.e. whatever product it is, it is going to be delivered on route k of CD g. If there is a direct shipment from a CD to a customer, although that edge may be part of other routes as well, index k makes them independent of each other. Taking the maximum value for that specific edge of the route with a specific k will result in the value of 2 in direct shipment, meaning that there are several products shipping directly from a CD to a supplier; thus the variable (x_{ij0}^{gk}) works properly. Therefore, the transportation cost is:

$$TC = \sum_{g} \sum_{k} \sum_{j \in I} \sum_{i \in \bar{I}, i<j} c_{ij} x_{ij0}^{gk} f_{gk} \qquad (2)$$

Substituting Eq. (1) in Eq. (2), we have:

$$\begin{aligned} TC &= \sum_{g} \sum_{k} \sum_{j \in I} \sum_{i \in \bar{I}, i<j} c_{ij} x_{ij0}^{gk} \frac{\sum_{j' \in I} \sum_{p \in P} \sum_{s \in S} b_p d_{j'p} y_{sj'p}^{gk}}{C_T} \\ &= \frac{1}{C_T} \sum_{s \in S} \sum_{g \in G} \sum_{k \in K} \sum_{j \in I} \sum_{i \in \bar{I}, i<j} \sum_{j' \in I} \sum_{p \in P} b_p c_{ij} d_{j'p} y_{sj'p}^{gk} \, x_{ij0}^{gk} \qquad (3) \end{aligned}$$

Note that when $y_{sj'p}^{gk} = 1$, based on the constraints defined for designing the routes, x_{ijp}^{gk} will be equal to 1 if edge (i, j) is a part of route (g, k). This means that $x_{ijp}^{gk} = 1$ if $x_{ij0}^{gk} = 1$ and $y_{sj'p}^{gk} = 1$. Also, $x_{ijp}^{gk} = 2$ if $x_{ij0}^{gk} = 2$ and $y_{sj'p}^{gk} = 1$ since in this case route (g, k) is a direct shipment to customer j for product p. Therefore, we can write Eq. (3) as follows:

$$TC = \frac{1}{C_T} \sum_{s \in S} \sum_{g \in G} \sum_{k \in K} \sum_{j \in I} \sum_{i \in \bar{I}, i < j} \sum_{j' \in I} \sum_{p \in P} c_{ij} b_p d_{j'p} y_{sj'p}^{gk} x_{ijp}^{gk} \qquad (4)$$

Eq. (4) is true because $y_{sj'p}^{gk} x_{ijp}^{gk} = 0$ if $y_{sj'p}^{gk} = 0$, no matter what the value of x_{ijp}^{gk} is, and $y_{sj'p}^{gk} x_{ijp}^{gk} = 1$ if and only if $y_{sj'p}^{gk} x_{ij0}^{gk} = 1$. Also, $y_{sj'p}^{gk} x_{ijp}^{gk} = 2$ if and only if $y_{sj'p}^{gk} x_{ij0}^{gk} = 2$.

Eq. (4) can now be interpreted in another way, distinct from that presented at the beginning. Whatever the frequency of shipment for route (g, k), all demands of the route must be delivered at the end, and all product quantities must traverse the full set of edges. The reason is that since TC is based on the vehicle capacity travelling over every edge, even if the vehicle becomes partially or totally empty, TC is still considered for the whole vehicle capacity. Therefore, the total volume of each product demand on a route should be multiplied by the cost of all edges of that route, which is in fact done in Eq. (4).

Note that TC is nonlinear as variables y and x are multiplied by each other.

3.2.3.2 Formulation of Holding Cost at the Customers

The customers follow EOQ policy for their inventory. Each type of product has a different holding cost. Therefore, before considering shipment frequencies, HC of product p for customer j is equal to $\frac{h_p d_{jp}}{2}$. Considering all the products demanded by customer j, the total HC for customer j is $\frac{\sum_{p \in P} h_p d_{jp}}{2}$. In our formulation, we wish to consider the holding costs of each route separately, to employ the frequencies of shipment. Thus, using y_{sip}^{gk} to decide which demand should be satisfied by route (g, k), the total HC for all customers on route (g, k) is equal to $\sum_{i \in I} \sum_p \sum_s \frac{h_p d_{ip} y_{sip}^{gk}}{2}$.

The latter equation is true when no frequencies of shipment are allowed. To implement the idea of shipment frequency, we have to divide that equation by f_{gk} because depending on the shipment frequency, the maximum quantity of products delivered at the customer in each trip, would be the maximum inventory level. Note we assume that the next shipment arrives at each customer whenever their inventory level is zero (EOQ policy). Since the maximum quantity shipped equals total demands divided by f_{gk}, the total HC over all routes is

$$HC = \sum_{g}\sum_{k}\sum_{i\in I}\sum_{p}\sum_{s}\frac{h_p d_{ip} y_{sip}^{gk}}{2f_{gk}} \qquad (5)$$

Since f_{gk} itself is a decision variable, HC is nonlinear, too. After substituting f_{gk} in Eq. (5), we have:

$$HC = \sum_{g}\sum_{k}\frac{\sum_{i\in I}\sum_{p\in P}\sum_{s\in S} h_p d_{ip} y_{sip}^{gk}}{\frac{2}{C_T}\sum_{i\in I}\sum_{p\in P}\sum_{s\in S} b_p d_{ip} y_{sip}^{gk}} \qquad (6)$$

Let $\bar{b}_p = \frac{2b_p}{C_T}$, so HC is:

$$HC = \sum_{g}\sum_{k}\frac{\sum_{i\in I}\sum_{p\in P}\sum_{s\in S} h_p d_{ip} y_{sip}^{gk}}{\sum_{i\in I}\sum_{p\in P}\sum_{s\in S} \bar{b}_p d_{ip} y_{sip}^{gk}} \qquad (7)$$

3.2.3.3 Formulation of Pipeline Inventory Cost from CDs to Customers

To calculate PC, the travel time on edge (i,j) should be considered. Treating the total time horizon as one period, t_{ij} is the ratio of the travel time on edge (i,j) to the period length. As mentioned before, each customer can be connected only to a customer with larger index or to a CD. Therefore, for all the demands, the holding cost should be considered on all edges, traversed by each route (g,k) until reaching its destination. Demand of p for customer j should be multiplied by $h_p t_{ij'}$ if edge (i,j') is part of route (g,k) and $j' \leq j$. As a result, for all the routes (g,k), PC is calculated by the following equation:

$$PC = \sum_{g} \sum_{k} \sum_{p} \sum_{j} d_{jp} h_p y_{sjp}^{gk} \sum_{i<j} \sum_{j'=i}^{j} t_{ij'} x_{ij'p}^{gk} \qquad (8)$$

Note that, since all demands of customers must traverse a route to be delivered, no matter what the shipment frequency is, we have built Eq. (8) without using the idea of shipment frequency. PC is also nonlinear.

3.2.3.4 Formulation of Transportation Cost from Suppliers to CDs

As noted previously, the shipment strategy considered for suppliers is direct to CDs. We thus need to calculate the transportation cost of the product quantities shipped from supplier s to cross-cock g. To formulate TC', the approach is to obtain the shipment frequency and then multiply it by the cost to traverse edge (s,g). Demands for each product shipped from supplier s to CD g are added up; the frequency of shipment is calculated by dividing that result by vehicle capacity. The sum of all shipment costs over all edges is the total transportation cost between suppliers and CDs.

$$f'_{sg} = \frac{\sum_i \sum_p \sum_k b_p d_{ip} y_{sip}^{gk}}{C'_T} \qquad (9)$$

$$TC' = \sum_{s} \sum_{g} c'_{sg} f'_{sg} \qquad (10)$$

Substituting Eq. (9) into Eq. (10), we have:

$$TC' = \sum_s \sum_g c'_{sg} \frac{\sum_i \sum_p \sum_k b_p d_{ip} y_{sip}^{gk}}{C'_T} = \frac{1}{C'_T} \sum_s \sum_g \sum_k \sum_i \sum_p c'_{sg} b_p d_{ip} y_{sip}^{gk} \qquad (11)$$

3.2.3.5 Formulation of Pipeline Inventory Cost from Suppliers to CDs

With the total time horizon again as one period, t_{sg} is the proportion of the period length represented by travel time on edge (s, g). PC' is the sum of holding costs incurred during each trip:

$$PC' = \sum_s \sum_i \sum_p \sum_g \sum_k t_{sg} h_p d_{ip} y_{sip}^{gk} \qquad (12)$$

3.2.3.6 Objective Function

The goal of the model is to minimize the total supply chain cost, which equals

$$\text{Min } \mathbb{C} = TC + HC + PC + TC' + PC' \qquad (13)$$

The objective function is highly nonlinear since TC, HC and PC are all nonlinear.

3.2.4 Constraints

We must ensure that demands of the customers are satisfied, and also proper routes are designed such that there is at least one dummy CD on each route. Constraints (15) and (16), which are respectively route designers and cycle eliminators, are borrowed from Archetti et al. (2007).

The constraints of this model are as follows:

$$\sum_{s}\sum_{g}\sum_{k} y_{sip}^{gk} \geq 1 \qquad i \in I\,, p \in P \qquad (14)$$

$$\sum_{j \in I, j>i} x_{ijp}^{gk} + \sum_{j \in \bar{I}, j<i} x_{jip}^{gk} = 2\sum_{s} y_{sip}^{gk} \qquad i \in \bar{I}, p \in P, k \in K, g \in G \qquad (15)$$

$$\sum_{i \in v}\sum_{j \in v, i<j} x_{ijp}^{gk} \leq \sum_{s}\sum_{i \in v} y_{sip}^{gk} - \sum_{s} y_{smp}^{gk} \qquad v \subset I, k \in K, g \in G, for\ some\ m \in v, p \in P \qquad (16)$$

$$\sum_{i<j} x_{ijp}^{gk} \leq 1 \quad \forall j \in I, p \in P, g \in G, k \in K \quad (17)$$

$$\sum_{j>i} x_{ijp}^{gk} \leq 1 \quad \forall i \in I, p \in P, g \in G, k \in K \quad (18)$$

$$\sum_{p}\sum_{i} x_{gip}^{gk} \leq \sum_{p}\sum_{i<j, i \in I} x_{ijp}^{gk} + Mv_j \;\; \forall j \in I, g \in G, k \in K \qquad (19)$$

$$\sum_{p}\sum_{i<j, i \in I} x_{ijp}^{gk} \leq M\left(1 - v_j\right) \;\; \forall j \in I, g \in G, k \in K \qquad (20)$$

$$\forall i, g \in G, i \neq g :\; x_{ijp}^{gk} = 0 \qquad (21)$$

$$\forall i, g \in G, i \neq g :\; y_{sip}^{gk} = 0 \qquad (22)$$

$$x_{ijp}^{gk} \in \{0,1\}, x_{igp}^{gk} \in \{0,1,2\}\,, i, j \;\in I, s \in S, g \in G, p \in P \; (23)$$

$$y_{sip}^{gk} = \{0,1\}, i \in \bar{I}, s \in S, g \in G, p \in P \qquad (24)$$

$$v_j \in \{0,1\}\,, j \in I \qquad (25)$$

Constraint (14) indicates that all customers should receive their requirements of product p from at least one CD. This constraint also means that at least one supplier should supply each customer, and therefore, all demands should be satisfied.

Constraint (15) is considered to make sure each customer is connected to two other nodes (if a vehicle enters that node, it should be able to exit it). Constraint (16) is a cycle elimination constraint.

Constraints (17) and (18) make sure that the costumers are connected to each other in an ascending order. Constraints (19) and (20) ensure that in a multi-product problem, all products delivered on a given route, traverse all the edges.

Constraints (23), (24) and (25) are related to feasible values of variables. The direct shipment from a CD to a customer is possible, so x_{gjp}^{gk} can equal 2. When index i in variables x_{ijp}^{gk} and y_{sip}^{gk} is an element of G, i must equal index g. Otherwise, the corresponding variables does not make sense, meaning that both indices should refer to the same CD involved in route k. (21) and (22) force the values of any "nonsense" variables to be zero.

To be able to solve the nonlinear mixed integer programming model with large number of variables introduced above, we next provide some analytical results.

3.3 Analytical Results

In this section, we demonstrate two analytical results. Theorem 1 is related to some solutions that seem to exist in the feasible region, but they should have been considered as infeasible. We prove that these solutions will never be optimal. As we know, each set of dummy CDs represents a single CD. At most one route is assigned to each dummy CD. Therefore, if two or more dummy CDs in a set serve different products for identical customers on the same edges, then it means that there are two or more vehicles traveling on identical routes, both starting from a specific CD. However, our purpose is to assign only one vehicle to each route. Therefore, normally, this condition should be considered as an infeasible solution. In Theorem 1, we will prove that although these solutions exist in the feasible region, they will not appear in the optimal solution because the cost of the route resulting from merging those identical routes will be lower than the sum of the costs of all individual routes. Theorem 2 helps us to reformulate our model with only one set of variables and simpler edge-based constraints.

Theorem 1. Let $E_1 = \{(i,j): (i,j) \in \text{Route}(g, k_1)\}$ and $E_2 = \{(i,j): (i,j) \in \text{Route}(g, k_2)\}$ such that $k_1 \neq k_2$ and $E_1 = E_2$. Then Routes (g, k_1) and (g, k_2) will not appear in the optimal solution together.

Proof: There might be two dummy CDs representing one specific route at the same time in feasible solutions, as can be seen in Fig. 4.

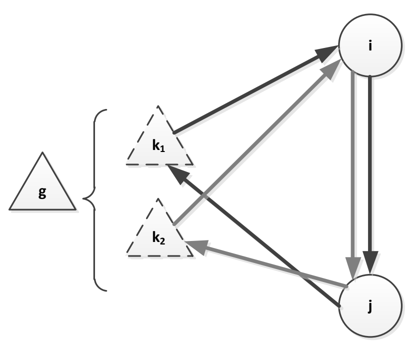

Figure 4 – Dummies of a same CD with identical routes

Routes 1 and 2 with the same edges and nodes may appear to be feasible, according to the constraints of the model. However, we will prove that the sum of the costs of these two routes is always higher than the cost of having all products shipped on a single route ($route_{merged}$). Solutions similar to Fig. 4 will not thus appear in the optimal solution.

According to the cost function, Eq. (13), the total cost consists of five parts. Three are related to trips between CDs and customers, and two parts involve supplier - CD trips. It is obvious that PC' and TC' for a specific CD, are the same for the two scenarios we compare. Consider the sum of PC' and TC':

$$TC' + PC' = \sum_{s}\sum_{g} c'_{sg} \frac{\sum_i \sum_p \sum_k b_p d_{ip} y_{sip}^{gk}}{C'_T} + \sum_{s}\sum_{i}\sum_{p}\sum_{g}\sum_{k} t_{sg} h_p d_{ip} y_{sip}^{gk} \quad (26)$$

Let k_a represent the indices of the two dummy CDs with identical routes. For CD g, each dummy CD k_a, and with I^* the set of customers served by those two dummy CDs, Eq. (26) becomes

$$TC'_a + PC'_a = \sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_p b_p d_{ip} y_{sip}^{gk_a}}{C'_T} + \sum_{s}\sum_{i\in I^*}\sum_{p} t_{sg} h_p d_{ip} y_{sip}^{gk_a} \quad (27)$$

The sum of corresponding costs for the two identical routes is:

$$\begin{aligned} &TC'_{a_1} + PC'_{a_1} + TC'_{a_2} + PC'_{a_2} \\ &= \sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_p b_p d_{ip} y_{sip}^{gk_{a_1}}}{C'_T} + \sum_{s}\sum_{i\in I^*}\sum_{p} t_{sg} h_p d_{ip} y_{sip}^{gk_{a_1}} \\ &+ \sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_p b_p d_{ip} y_{sip}^{gk_{a_2}}}{C'_T} + \sum_{s}\sum_{i\in I^*}\sum_{p} t_{sg} h_p d_{ip} y_{sip}^{gk_{a_2}} \\ &= \sum_{k_a}\left(\sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_p b_p . d_{ip} . y_{sip}^{gk}}{C'_T} + \sum_{s}\sum_{i\in I^*}\sum_{p} t_{sg} h_p d_{ip} y_{sip}^{gk}\right) \quad (28) \end{aligned}$$

Eq. (26) for the merged route ($route_{merged}$) will be as follows:

$$TC'_{merged} + PC'_{merged} = \sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_{p} \sum_{k_a} b_p d_{ip} y_{sip}^{gk}}{C'_T} + \sum_{s} \sum_{i\in I^*} \sum_{p} \sum_{k_a} t_{sg} h_p d_{ip} y_{sip}^{gk} \quad (29)$$

Equations (28) and (29) are equal:

$$\sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_{p} \sum_{k_a} b_p . d_{ip} . y_{sip}^{gk}}{C'_T} + \sum_{s} \sum_{i\in I^*} \sum_{p} \sum_{k_a} t_{sg} h_p d_{ip} y_{sip}^{gk}$$
$$= \sum_{k_a} \left(\sum_{s} c'_{sg} \frac{\sum_{i\in I^*} \sum_{p} b_p . d_{ip} . y_{sip}^{gk}}{C'_T} + \sum_{s} \sum_{i\in I^*} \sum_{p} t_{sg} h_p d_{ip} y_{sip}^{gk} \right) \quad (30)$$

Therefore, PC' and $TC'$are the same for both scenarios.

The rest of the cost function in the model consists of three parts, based on Eq. (13): TC, HC and PC. We look at each part separately:

Comparison of TCs: Consider Eq. (4). For identical routes 1 and 2, we have:

$$TC_a = \frac{1}{C_T} \sum_{(i,j)\in E_1} \sum_{p\in P_a} \sum_{j'\in I^*} c_{ij} b_p d_{j'p} \quad a = 1,2 \qquad (31)$$

In terms of shipment frequency, since the two dummy CDs are actually a single CD, the edge travelling costs are the same for both routes. This result can also be derived from the following equation

$$TC_a = f_{g1} \sum_{(i,j)\in E_1} c_{ij}\, , \quad a = 1,2 \quad (32)$$

The cost of a single trip for each CD is $\sum_{(i,j)\in E_1} c_{ij}$, and multiplying that cost by shipment frequency yields the total transportation cost for that route.

TC_{merged}, the transportation cost of the merged route, is as follows:

$$TC_{merged} = \frac{1}{C_T} \sum_{(i,j)\in E_1} \sum_{p\in P_1 \cup P_2} \sum_{j' \in I^*} c_{ij} b_p d_{j'p} = f_{g,merged} \sum_{(i,j)\in E_1} c_{ij} \qquad (33)$$

Since the total shipment quantity for the merged route is equal to the sum of the quantities shipped on Routes 1 and 2, and shipment frequency is quantities shipped divided by the vehicle capacity, we have:

$$f_{g1} + f_{g2} = f_{g,merged} \qquad (34)$$

Thus, $TC_{merged} = TC_1 + TC_2$.

Comparison of $\boldsymbol{HC}$s: For CD g and dummy CD $k = 1$, the holding cost is:

$$HC_1 = \sum_{i\in I} \sum_{p} \sum_{s} \frac{h_p d_{ip} y_{sip}^{g1}}{2 f_{g1}} \qquad (35)$$

For $k = 2$ and the same CD, the holding cost is:

$$HC_2 = \sum_{i\in I} \sum_{p} \sum_{s} \frac{h_p d_{ip} y_{sip}^{g2}}{2 f_{g2}} \qquad (36)$$

The sum of the cost of two separate routes is:

$$HC_1 + HC_2 = \sum_{i\in I} \sum_{p} \sum_{s} \frac{h_p d_{ip} y_{sip}^{g1}}{2 f_{g1}} + \sum_{i\in I} \sum_{p} \sum_{s} \frac{h_p d_{ip} y_{sip}^{g2}}{2 f_{g2}} \qquad (37)$$

For the merged route, the holding cost is as follows:

$$HC_{merged} = \sum_{i \in I} \sum_{p} \sum_{s} \frac{h_p d_{ip} y_{sip}^{g1} + h_p d_{ip} y_{sip}^{g2}}{2(f_{g1} + f_{g2})} \qquad (38)$$

Eq. (38) means that all demands which were delivered by two separate vehicles (two separate dummy CDs) on the same route are now summed up to be delivered on one route. The frequencies are also added up.

We need to to prove that $HC_1 + HC_2 > HC_{merged}$, which is in fact the case:

$$\frac{a+c}{b+d} < \frac{a}{b} + \frac{c}{d} \qquad (39)$$

where

$$\sum_{i \in I} \sum_{p} \sum_{s} h_p d_{ip} y_{sip}^{g1} = a \qquad (40)$$

$$2f_{g1} = b \qquad (41)$$

$$\sum_{i \in I} \sum_{p} \sum_{s} h_p d_{ip} y_{sip}^{g2} = c \qquad (42)$$

$$2f_{g2} = d \qquad (43)$$

and a, b, c and d are all positive.

To prove that Eq. (39) is valid, consider that equation:

$$\frac{a+c}{b+d} = \frac{a}{b+d} + \frac{c}{b+d} \qquad (44)$$

We know that:

$$\frac{a}{b+d} < \frac{a}{b}, \qquad \frac{c}{b+d} < \frac{c}{d} \qquad (45)$$

By adding up the two inequalities in Eq. (45), we have:

$$\frac{a}{b+d} + \frac{c}{b+d} < \frac{a}{b} + \frac{c}{d} \qquad (46)$$

Therefore,

$$\frac{a+c}{b+d} < \frac{a}{b} + \frac{c}{d} \qquad (47)$$

This result proves that the holding cost related to the merged route is lower than the sum of the holding costs of the separate routes:

$$\sum_{i\in I}\sum_{p}\sum_{s}\frac{h_p d_{ip} y_{sip}^{g1} + h_p d_{ip} y_{sip}^{g2}}{2(f_{g1}+f_{g2})} < \sum_{i\in I}\sum_{p}\sum_{s}\frac{h_p d_{ip} y_{sip}^{g1}}{2f_{g1}} + \sum_{i\in I}\sum_{p}\sum_{s}\frac{h_p d_{ip} y_{sip}^{g2}}{2f_{g2}} \qquad (48)$$

Comparison of $\boldsymbol{PC}$s: For CD g and dummy CD $k=1$, let P_1 and P_2 be the sets of products shipping on routes 1 and 2 respectively. Then, PC_1 is:

$$PC_1 = \sum_{j\in I^*}\sum_{p\in P_1} d_{jp} h_p \sum_{i\in I^*, i<j}\;\sum_{j'\in I^*, j'\le j} t_{ij'} \qquad (49)$$

And for $k=2$ we have:

$$PC_2 = \sum_{j\in I^*}\sum_{p\in P_2} d_{jp} h_p \sum_{i\in I^*, i<j}\;\sum_{j'\in I^*, j'\le j} t_{ij'} \qquad (50)$$

For route_{merged} , the total demand shipped is the sum of all demands shipped on the two separate routes, and the pipeline inventory cost equals the sum of PC_1 and PC_2.

$$PC_{merged} = \sum_{j \in I^*} \sum_{p \in P_1 \cup P_2} d_{jp} h_p \sum_{i \in I^*, i < j} \sum_{j' \in I^*, j' \leq j} t_{ij'} = PC_1 + PC_2 \qquad (51)$$

As explained above, the transportation cost and the pipeline inventory cost are the same for both scenarios, and the only difference is in holding costs at the customers. Since the holding cost related to the merged route is always lower than the sum of the holding costs of the separate routes, two independent routes corresponding to two dummy CDs with the same edges and nodes will never appear in the optimal solution. □

Theorem 2. All products of one type p, sent through a specific CD g, are provided by that single supplier s which yields the minimum value for $\left(\frac{c'_{sg}}{C'_T} b_p + t_{sg} h_p\right)$.

Proof. Consider the optimal solution with a set of customers served, whether totally or partially, by a specific CD. By analyzing the cost of shipments from suppliers to that CD, we want to see which supplier has provided those products to that CD. Using Equations (11) and (12), for a specific product p demanded by customer i and shipped through CD g, the shipment cost (γ) from supplier s is the sum of the corresponding transportation cost and pipeline inventory cost:

$$\gamma_{sipg} = \frac{c'_{sg}}{C'_T} b_p d_{ip} + t_{sg} h_p d_{ip} \; = d_{ip} \left(\frac{c'_{sg}}{C'_T} b_p + t_{sg} h_p\right) \qquad (52)$$

We take the minimum of this cost among all suppliers, and name the corresponding supplier as s^*. Let γ_{s^*ipg} be the minimum of the cost among all suppliers sending demand d_{ip} to CD g.

When $s = 1$, $s = 2$,... and $s = n$, Eq. (52) is:

$$\gamma_{1ipg} = d_{ip}\left(\frac{c'_{1g}}{C'_T}b_p + t_{1g}h_p\right) \quad (53)$$

$$\gamma_{2ipg} = d_{ip}\left(\frac{c'_{2g}}{C'_T}b_p + t_{2g}h_p\right) \qquad (54)$$

$$\gamma_{2ipg} = d_{ip}\left(\frac{c'_{ng}}{C'_T}b_p + t_{ng}h_p\right) \qquad (55)$$

Comparing Equations (53), (54) and (55), we can see that since the demand d_{ip} is constant, we have:

$$\left(\frac{c'_{s^*g}}{C'_T}b_p + t_{s^*g}h_p\right) = \min_{s \in S}\left\{\left(\frac{c'_{sg}}{C'_T}b_p + t_{sg}h_p\right)\right\} \qquad (56)$$

Now, consider one other customer (i') who also receives product p through CD g. This customer will get served by a supplier with minimum cost as well. Suppose the supplier with minimum cost serving customer i' through CD g is $s^*_{i'}$. Therefore, based on Eq. (54), we have:

$$d_{i'p}\left(\frac{c'_{s^*_{i'}g}}{C'_T}b_p + t_{s^*_{i'}g}h_p\right) \leq d_{i'p}\left(\frac{c'_{s^*g}}{C'_T}b_p + t_{s^*g}h_p\right) \qquad (57)$$

Since $d_{i'p}$ is a positive value, it will cancel out from the two sides of the inequality. Therefore, inequality (57) will change to:

$$\left(\frac{c'_{s^*_{i'}g}}{C'_T}b_p + t_{s^*_{i'}g}h_p\right) \leq \left(\frac{c'_{s^*g}}{C'_T}b_p + t_{s^*g}h_p\right) \qquad (58)$$

Based on Eq. (55), inequality (58) will become

$$\left(\frac{c'_{s^*_{i'}g}}{C'_T}b_p + t_{s^*_{i'}g}h_p\right) = \left(\frac{c'_{s^*g}}{C'_T}b_p + t_{s^*g}h_p\right) \qquad (59)$$

As a result, s^* is the supplier that will serve customer i' for product p, the same supplier that provides product p for customer i. Therefore, the entire product p that goes through CD g will be provided by a single supplier, the one with the minimum value for $\left(\frac{c'_{sg}}{C'_T}b_p + t_{sg}h_p\right)$. □

Suppose s^* is the single supplier for product p_1 shipped through CD g. s^* is not always the best supplier for other types of products because choosing the best supplier depends on p.

Corollary 1. Consider an ascending order for $b_p, p \in P$ as $b_{(1)} \leq b_{(2)} \leq \cdots \leq b_{(|P|)}$ such that product (i) is the ith smallest product in terms of volume. If $h_{(1)} \leq h_{(2)} \leq \cdots \leq h_{(|P|)}$, then s^* is the single source for all the products shipping to CD g. □

Using Theorem 2 in the next chapter, the model is formulated in terms of a single set of variables, without changing any of the assumptions.

Chapter 4

Solution Method

4.1 Reformulation

4.1.1 Modifying Decision Variables

Based on the analytical results provided in Chapter 3, we modify the initial formulation by omitting the set of variables y_{sip}^{gk}. Reformulation and removing y_{sip}^{gk} has many advantages. First, the number of variables reduces significantly. Second, we provide an edge-based formulation for the model. Also, we demonstrate that we can linearize the nonlinear objective function, which enables a solution method based on model decomposition and a column generation algorithm.

y_{sip}^{gk} indicates whether each node is on a particular route or not. To remove that set of variables, we need to relate the suppliers to x_{ijp}^{gk}. Based on Theorem 2, we define $s(g, p)$ as the supplier who provides product p to CD g. In that case, since s depends only on p and g, we can remove the summation over index s in the model formulation. Moreover, when $x_{ijp}^{gk} = 1$, we know that product p is sent though edge (i,j) by $s(g,p)$. Therefore, we modify variables x_{ijp}^{gk} such that they carry the information corresponding to the suppliers. The new set of edge decision variables is $x_{s(g,p)ijp}^{gk}$ which we refer to it as x_{sijp}^{gk} for convenience because the information about g and p is already there. We add the subscript s to the existing set of variables without changing any of the assumptions, and this will be the only set of decision variables in the model. For other parameters, we use $s(g,p)$ instead of s.

In addition, we define $x_{sgjp}^{gk} = x_{sg_1jp}^{gk} + x_{sg_2jp}^{gk}$ such that $x_{sg_1jp}^{gk} = 1$ if customer j is the first customer on the route starting at CD g, and $x_{sg_2jp}^{gk} = 1$ if customer j is the last customer that the vehicle visits before returning to CD g. Note that g_1 and g_2 both refer to CD g. In addition, in case of having a direct shipment from CD g to Customer j, both $x_{sg_1jp}^{gk}$ and $x_{sg_2jp}^{gk}$ will be equal to 1, which is equivalent to having $x_{sgjp}^{gk} = 2$. Separation of x_{sgjp}^{gk} into two variables has two advantages: first, all variables will be binary, and second, the process of rewriting the edge-based routing constraints will be easier.

When removing y_{sip}^{gk}, we now need to include x_{sijp}^{gk} such that it takes the role of y_{sip}^{gk}. Since x_{sijp}^{gk} refers to edge (i,j), it can represent both nodes i and j; whereas, y_{sip}^{gk} only points to node i. Hence, we need to rewrite the equations, so that they become compatible with this substitution. This is elaborated in the next sections.

After modifying the decision variables, we reformulate the problem. Each part of the new formulation is now given.

4.1.2 Reformulation of TC

After substituting x_{sijp}^{gk} for y_{sip}^{gk}, we take into account d_{jp} as the demand of Customer j on edge (i,j) because if we consider d_{ip}, the demand of the last customer of the route will not be counted. Note that $i \neq g_2$ because if we consider edge (g_2, j), then d_{jp} will be counted twice. Hence, based on Eq. (4), we have:

$$TC = \frac{1}{2C_T} \sum_{g\in G} \sum_{k\in K} \sum_{j\in I} \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_{j'\in I} \sum_{\substack{(i'=g_1)\\ \vee(i'=g_2)\\ \vee(i'\in I, i'<j')}} \sum_{p\in P} b_p c_{i'j'} d_{jp} x_{sijp}^{gk} x_{si'j'p}^{gk} \quad (60)$$

Let us define a new parameter $c''_{ij} = \frac{c_{ij}}{2C_T}$, the shipment cost per unit volume divided by 2 on edge (i,j). Therefore, TC becomes:

$$TC = \sum_{g} \sum_{k} \sum_{j\in I} \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_{j'\in I} \sum_{\substack{(i'=g_1)\\ \vee(i'=g_2)\\ \vee(i'\in I, i'<j')}} \sum_{p\in P} b_p c''_{i'j'} d_{jp} x_{sijp}^{gk} x_{si'j'p}^{gk} \qquad (61)$$

x_{sijp}^{gk} refers to the customers i and j that are on route (g,k) receiving product p from supplier s. If those customers are parts of route (g,k), then this variable equals 1. Now variable $x_{si'j'p}^{gk}$ will be equal to 1 if edge (i',j') is a part of route (g,k). Note that the demands of customer j should be considered to traverse all edges (i',j'), if (i',j') is a part of the corresponding route. This is because TC accounts for all vehicle trips, whether empty or full.

4.1.3 Reformulation of HC

As in the case of TC, we consider edges instead of nodes to formulate HC. The idea is to add up the holding costs of Customer j for edge (i,j) for all edges of the route except the last edge which is edge (g_2, j). The reason is the demand of the last customer of the route is already considered. This applies as well to frequencies of shipment. HC based on Eq. (7) is modified as

$$HC = \sum_g \sum_k \frac{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i\in I, i<j)}} \sum_p h_p d_j x_{sijp}^{gk}}{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i\in I, i<j)}} \sum_p \bar{b}_p d_j x_{sijp}^{gk}} \quad (62)$$

4.1.4 Reformulation of PC

Pipeline inventory cost does not change significantly. There is no pipeline inventory for the last edge of a particular route, since the vehicle is empty. Since we do not take the last edge into account when calculating PC, the reformulated equation becomes

$$PC = \sum_g \sum_k \sum_p \sum_j \sum_{\substack{(i=g_1) \\ \vee(i\in I, i<j)}} d_{jp} h_p x_{sijp}^{gk} \sum_{\substack{(i'=g_1) \\ \vee(i'\in I, i'<j)}} \sum_{j'>i}^{j} t_{i'j'} x_{si'j'p}^{gk} \quad (63)$$

As $t_{i'j'}$ is an element of a matrix in which the lower triangle is zero, we can modify the last summation of Eq. (63) as:

$$PC = \sum_g \sum_k \sum_p \sum_j \sum_{\substack{(i=g_1) \\ \vee(i\in I, i<j)}} d_{jp} h_p x_{sijp}^{gk} \sum_{\substack{(i'=g_1) \\ \vee(i'\in I, i'<j)}} \sum_{j'=1}^{j} t_{i'j'} x_{si'j'p}^{gk} \quad (64)$$

4.1.5 Reformulation of TC'

To deliver all demands of the customers, the product quantities should first traverse the edges connecting the suppliers to CDs. Using the same variable sets, we calculate the costs thus incurred; based as before on shipment frequencies. Therefore, the new equation for TC' is:

$$f'_{sg} = \frac{1}{2C'_T}\sum_{k}\sum_{j}\sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}}\sum_{p} b_p d_{jp} x_{sijp}^{gk} \qquad (65)$$

$$TC' = \sum_{s}\sum_{g} c'_{sg} f'_{sg} \qquad (66)$$

Thus, we have:

$$TC' = \sum_{g}\sum_{k}\sum_{j}\sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}}\sum_{p} \frac{c'_{s(g,p),g}}{2C'_T} b_p d_{jp} x_{sijp}^{gk} \qquad (67)$$

4.1.6 Reformulation of PC'

Using the same approach for reformulation of TC', pipeline inventory cost from suppliers to the cross-docks is:

$$PC' = \sum_{g}\sum_{k}\sum_{j}\sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}}\sum_{p} \frac{t_{s(g,p),g} h_p d_{jp} x_{sijp}^{gk}}{2} \qquad (68)$$

Note that in the PC' and TC' equations, x_{sijp}^{gk} is interpreted as customers i and j who are served by supplier s on route g and product p; that variable does not refer to edge(i, j).

4.1.7 Edge-Based Constraints

Having removed the node decision variables, we now write the constraints in terms of x_{sijp}^{gk} which can be considered as edge-based constraints. The new edge-based constraints are

$$\sum_{g}\sum_{k}\sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} x_{sijp}^{gk} \geq 1 \quad \forall j \in I, p \in P \qquad (69)$$

$$\forall j, p : d_{jp} > 0, \sum_{\substack{(i=g_1) \\ \vee(i=g_2) \\ \vee(i \in I)}} x_{sijp}^{gk} = 2 \qquad \text{for one route } (g,k) \quad (70)$$

$$M \sum_{j \in I} \left(x_{sg_1jp}^{gk} + x_{sg_2jp}^{gk} \right) - \sum_{i' \in I} \sum_{j \in I} x_{si'jp}^{gk} \geq 0 \quad \forall g \in G, p \in P, k \in K \quad (71)$$

$$\sum_{i<j} x_{sijp}^{gk} \leq 1 \quad \forall j \in I, p \in P, g \in G, k \in K \quad (72)$$

$$\sum_{j>i} x_{sijp}^{gk} \leq 1 \quad \forall i \in I, p \in P, g \in G, k \in K \quad (73)$$

$$\sum_{p} \sum_{i} x_{sgip}^{gk} \leq \sum_{p} \sum_{i<j, i \in I} x_{sijp}^{gk} + Mv_j \quad \forall j \in I, g \in G, k \in K \quad (74)$$

$$\sum_{p} \sum_{i<j, i \in I} x_{sijp}^{gk} \leq M(1 - v_j) \quad \forall j \in I, g \in G, k \in K \quad (75)$$

$$\forall i, g \in G, i \neq g : \; x_{sijp}^{gk} = 0 \quad (76)$$

$$x_{sijp}^{gk} \in \{0,1\}, \forall g \in G: \; i \in I \cup \{g_1, g_2\}, i < j, j \in I, p \in P \quad (77)$$

$$v_j \in \{0,1\}, j \in I \quad (78)$$

Relations (69) indicate that all customers should receive their requirements of product p through at least one CD. Constraints (70) are considered to make sure the degree of the vertex corresponding to each customer equals 2 if that customer's demand for product p is positive. The inequalities (71) are cycle-elimination constraints (This means that only a single CD should exist on any route). Constraints (72) and (73) ensure that the routes are generated based on the ordered customers. To make sure that only one route is generated per each dummy CD, Constraints (74) and (75) are considered. Finally, (76), (77) and (78) define the ranges and allowable indices for the decision variables.

4.2 Linearizing the Model Formulation

As can be seen in Equations (61), (62) and (64), TC, HC and PC are nonlinear. TC and PC are quadratic while HC is the sum of a number of fractional terms. In Theorem 3, we demonstrate that we can linearize the quadratic parts by replacing each multiplication of two variables by a new variable.

Proposition 1. $z_{siji'j'p}^{gk}$ can be defined as equivalent to the product of $x_{sijp}^{gk}.x_{si'j'p}^{gk}$ that is when $i \in I \cup \{g_1, g_2\}, i < j, j \in I,\ i' \in I \cup \{g_1, g_2\}, i' < j', j' \in I,$ and the variable $z_{siji'j'p}^{gk} \in \{0,1\}$.

Proof. Since x_{sijp}^{gk} and $x_{si'j'p}^{gk}$ are binary variables, their multiplication together can have only two results, 0 and 1.

Variables	$\boldsymbol{x_{sijp}^{gk}}$	$\boldsymbol{x_{si'j'p}^{gk}}$	$\boldsymbol{z_{siji'j'p}^{gk} = x_{sijp}^{gk}.x_{si'j'p}^{gk}}$
Possible Values	0	0	0
	0	1	0
	1	0	0
	1	1	1

Table 2 – Values of z obtained by values of x

Clearly $z_{siji'j'p}^{gk} = 1$ only when both x_{sijp}^{gk} and $x_{si'j'p}^{gk}$ are equal to one; otherwise, $z_{siji'j'p}^{gk} = 0$. Note that when $i = i'$ and $j = j'$, we have:

$$z_{sijijp}^{gk} = x_{sijp}^{gk}.x_{sijp}^{gk} = x_{sijp}^{gk} \quad (79)$$

Therefore, we can consider z_{sijijp}^{gk} as the decision variable related to each edge (i,j) for product p on a particular route. Based on the definition of $z_{siji'j'p}^{gk}$ and Eq. (79), we provide a set of constraints which relate the variable arrays x to z, such that the logic of Table (2) holds:

$$z_{sijijp}^{gk} + z_{si'j'i'j'p}^{gk} \leq z_{siji'j'p}^{gk} + 1 \qquad \forall i,i' \in I \cup \{g_1, g_2\}; i < j; i' < j'; j,j' \in I\,; p \in P \quad (80)$$

Constraint (80) indicates if edges (i,j) and (i',j') are chosen (the left hand side equals 2), then $z_{siji'j'p}^{gk}$ must also equal 1. If either of these edges is not part of that route, then there is nothing that forces $z_{siji'j'p}^{gk}$ to be 1. Since this is a minimization problem and the coefficient of $z_{siji'j'p}^{gk}$ is positive in the linear parts of the objective function, and zero in the nonlinear part when $(i,j) \neq (i',j')$, $z_{siji'j'p}^{gk}$ will equal zero in this case. That is equivalent to the results shown in Table 2. Therefore, considering Constraint (80), Eq. (81) holds:

$$z_{siji'j'p}^{gk} = x_{sijp}^{gk}.x_{si'j'p}^{gk} \qquad \forall i,i' \in I \cup \{g_1, g_2\}; i < j; i' < j'; j,j' \in I\,; p \in P \quad (81)\ \square$$

Corollary 2. For all $i,i' \in I; i < j; i' < j'; j,j' \in I\,; p \in P$, if $i \neq i'$, then $z_{siji'jp}^{gk} = 0$. Similarly, if $j \neq j'$, then $z_{sijij'p}^{gk} = 0$.

Proof. The preceding follows from the assumption that indices of customers are ascending on a route. There are thus no pairs of edges available on a route such that their starting points are identical. □

4.3 Reformulation of Model Based on Variable *z*

Now, we reformulate the model based on the new set of variables defined to linearize the model.

Considering Equations (61) and (81), the linearized TC is

$$TC = \sum_{g} \sum_{k} \sum_{j \in I} \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_{j' \in I} \sum_{\substack{(i'=g_1) \\ \vee(i'=g_2) \\ \vee(i' \in I, i'<j')}} \sum_{p \in P} b_p c''_{i'j'} d_{jp} z^{gk}_{siji'j'p} \qquad (82)$$

Since HC is not quadratic, we use Eq. (79) to modify Eq. (62). HC thus becomes

$$HC = \sum_{g} \sum_{k} \frac{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p h_p d_{jp} z^{gk}_{sijijp}}{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p \bar{b}_p d_{jp} z^{gk}_{sijijp}} \qquad (83)$$

To reformulate PC, similar to TC, we replace $x^{gk}_{sijp}.x^{gk}_{si'j'p}$by $z^{gk}_{siji'j'p}$ in Eq. (64). Therefore, PC becomes linear.

$$PC = \sum_{g} \sum_{k} \sum_{p} \sum_{j} \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} d_{jp} h_p \sum_{\substack{(i'=g_1) \\ \vee(i' \in I, i'<j)}} \sum_{j'=1}^{j} t_{i'j'} z^{gk}_{siji'j'p} \qquad (84)$$

TC' and PC' are both linear, so we use only Eq. (79) to modify those two expressions. The results are

$$TC' = \sum_{g} \sum_{k} \sum_{j} \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_{p} \frac{c'_{s(g,p),g}}{2C'_T} b_p d_{jp} z^{gk}_{sijijp} \qquad (85)$$

and

$$PC' = \sum_{g} \sum_{k} \sum_{j} \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_{p} \frac{t_{s(g,p),g} h_p d_{jp} z^{gk}_{sijijp}}{2} \qquad (86)$$

We now turn attention to the constraints. Since they contain no quadratic term, we replace all $x_{sijp}^{gk}$with z_{sijijp}^{gk} based on Eq. (79). We also add the inequality (80) to the set of constraints, which now become

$$\sum_{g}\sum_{k}\sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} z_{sijijp}^{gk} \geq 1 \quad \forall j \in I, p \in P \qquad (87)$$

$$\forall j, p : d_{jp} > 0, \sum_{\substack{(i=g_1)\\ \vee(i=g_2)\\ \vee(i\in I)}} z_{sijijp}^{gk} = 2 \quad \text{for one route } (g, k) \qquad (88)$$

$$M\sum_{j\in I}\left(z_{sg_1jg_1jp}^{gk} + z_{sg_2jg_2jp}^{gk}\right) - \sum_{i'\in I}\sum_{j\in I} z_{si'ji'jp}^{gk} \geq 0 \quad \forall g \in G, p \in P, k \in K \qquad (89)$$

$$z_{sijijp}^{gk} + z_{si'j'i'j'p}^{gk} \leq z_{siji'j'p}^{gk} + 1 \qquad \forall i, i' \in I \cup \{g_1, g_2\}; i < j; i' < j'; j, j' \in I\, ; p \in P \qquad (90)$$

$$\forall i, g \in G, i \neq g : \; z_{sijijp}^{gk} = 0 \qquad (91)$$

$$z_{siji'j'p}^{gk} \in \{0,1\}, \qquad \forall i, i' \in I \cup \{g_1, g_2\}; i < j; i' < j'; j, j' \in I\, ; p \in P \qquad (92)$$

4.4 Solution Method

Although the quadratic equations of the objective function have been linearized, HC is still nonlinear: it is a set of fractions in which both numerator and denominator are linear. Therefore, the model is not still solvable. The approach of dummy CDs that was previously explained in the problem definition is very helpful to solve the model. It allows us to decompose the objective function over g and k. This means that we can consider the cost of each route (g, k) separately. Objective function decomposition has two advantages. We can calculate the cost of all possible routes separately, and hence define a set-covering problem to choose a number of these routes such that the demands of all customers are satisfied and the total cost is minimized. The other advantage of decomposition is we are then able to linearize HC. We elaborate on these two approaches further.

4.4.1 Objective Function Decomposition

All five parts of the objective function are decomposable over g and k. The result of that decomposition is

$$TC_{gk} = \sum_{j \in I} \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_{j' \in I} \sum_{\substack{(i'=g_1) \\ \vee(i'=g_2) \\ \vee(i' \in I, i'<j')}} \sum_{p \in P} b_p c''_{i'j'} d_{jp} z^{gk}_{siji'j'p} \qquad (93)$$

$$HC_{gk} = \frac{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p h_p d_{jp} z^{gk}_{sijijp}}{\sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p \bar{b}_p d_{jp} z^{gk}_{sijijp}} \qquad (94)$$

$$PC_{gk} = \sum_p \sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} d_{jp} h_p \sum_{\substack{(i'=g_1) \\ \vee(i' \in I, i'<j)}} \sum_{j'=1}^{j} t_{i'j'} z^{gk}_{siji'j'p} \qquad (95)$$

$$TC'_{gk} = \sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p \frac{c'_{s(g,p),g}}{2C'_T} b_p d_{jp} z^{gk}_{sijijp} \qquad (96)$$

$$PC'_{gk} = \sum_j \sum_{\substack{(i=g_1) \\ \vee(i \in I, i<j)}} \sum_p \frac{t_{s(g,p),g} h_p d_{jp} z^{gk}_{sijijp}}{2} \qquad (97)$$

This means that the total objective function is decomposed into $|G| \times |K|$ sub-problems. The constraints will be redefined for each decomposed problem separately.

4.4.2 Linearizing HC

After decomposition, we can linearize HC by setting the whole fraction equal to the variable R_{gk}.

$$\frac{\sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p h_p d_{jp} z_{sijijp}^{gk}}{\sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p \bar{b}_p d_{jp} z_{sijijp}^{gk}} = R_{gk} \qquad (98)$$

Assuming the denominator is greater than zero (if the denominator is equal to zero, then based on the structure of the fraction, the numerator is also zero. We define $\frac{0}{0}$ as 0.),

$$\sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p h_p d_{jp} z_{sijijp}^{gk} = R_{gk} \sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p \bar{b}_p d_{jp} z_{sijijp}^{gk} \qquad (99)$$

Let us define variable y_{sijijp}^{gk} as:

$$y_{sijijp}^{gk} = R_{gk} . z_{sijijp}^{gk} \qquad (100)$$

Therefore, Eq. (99) changes to:

$$\sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p h_p d_{jp} z_{sijijp}^{gk} = \sum_j \sum_{\substack{(i=g_1)\\ \vee(i\in I, i<j)}} \sum_p \bar{b}_p d_{jp} y_{sijijp}^{gk} \qquad (101)$$

Now, we need to linearize Eq. (100). Constraints (102)-(105) are defined, so that Eq. (100) holds:

$$Mz_{sijijp}^{gk} - y_{sijijp}^{gk} + R_{gk} \leq M \qquad i \in I \cup \{g_1, g_2\}, j \in I, i < j, p \in P \quad (102)$$

$$-Mz_{sijijp}^{gk} + y_{sijijp}^{gk} \leq 0 \qquad i \in I \cup \{g_1, g_2\}, j \in I, i < j, p \in P \quad (103)$$

$$y_{sijijp}^{gk} - R_{gk} \leq 0 \qquad i \in I \cup \{g_1, g_2\}, j \in I, i < j, p \in P \quad (104)$$

$$y_{sijijp}^{gk} \geq 0, R_{gk} \geq 0 \qquad (105)$$

4.4.3 Set-Covering Model

We now provide a set-covering model in which the goal is to choose a number of routes among all possible ones, such that the total cost is minimized, and all the customers' demands are satisfied. Consider the cost associated with each route, which is a subset of all customers with demanded products and a single CD. First we define set $IP = \{(i,p) | i \in I, p \in P: d_{ip} > 0\}$. This is the set of all combinations of customers and products, such that those customers have positive demands for those products. Now, let R be a non-empty subset of IP, and $\mathcal{R}$ be the set of all non-empty subsets of IP. Therefore, we have $R \in \mathcal{R}$. Based on Equations (93) to (97), the cost of route (g, k) is given by

$$C_{R,g,k} = \sum_{j;\forall p:(j,p)\in R} \sum_{p;\forall i':(i',p)\in R} d_{jp} b_p \left(c''_{g(|I^*|)} + \sum_{((i);\forall p:((i),p)\in R)\vee((i)=(0)\ for\ CD\ g)} c''_{(i)(i+1)} \right)$$

$$+ \frac{\sum_{(i;\forall p:(i,p)\in R)} \sum_{p;\forall i:(i,p)\in R} h_p d_{ip}}{2\sum_{(i;\forall p:(i,p)\in R)} \sum_{p;\forall i:(i,p)\in R} b_p d_{ip}}$$

$$+ \sum_{(j);\forall p:((j),p)\in R} \sum_{p:((j),p)\in R} d_{(j)p} h_p \sum_{((i);\forall p:((i),p)\in R)\vee((i)=(0)\ for\ CD\ g),(i)<(j)} t_{(i)(i+1)}$$

$$+ \frac{\sum_{(i;\forall p:(i,p)\in R)} \sum_{p;\forall i:(i,p)\in R} c'_{s(g,p),g} b_p d_{ip}}{C'_T}$$

$$+ \sum_{i;\forall p:(i,p)\in R} \sum_{p;\forall i:(i,p)\in R} t_{s(g,p),g} h_p d_{ip} \qquad (106)$$

Note that (i) and (j) both denote the ordered sequence of indices of the members of R (i.e., customers in R), based solely on i. For example, if Customer 7 is the second customer on the route, the ordered sequence of that customer is (2). A given CD is considered as node (0). Also, I^* is the set of customers on route (g,k).

Let C_R be the minimum of all $C_{R,g,k}$. C_R is thus the cost of the route assigned to CD g which has the minimum cost among all CDs that can potentially be assigned to route R.

We know that all dummy CDs of a particular CD are in fact the same as the CD itself in terms of their distance, travel time and transportation cost between other nodes. Consider two of the sub-problems $\mathbb{C}(g,k_1)$ and $\mathbb{C}(g,k_2)$ after decomposition. These two sub-problems are identical because the two dummy CDs are identical. Therefore, we can ignore index k and decompose the objective function only over g, keeping in mind that more than one route is allowed to be assigned to each CD. Hence, we have $|G|$ sub-problems:

$$\text{Min } \mathbb{C}(g) = TC_g + HC_g + PC_g + TC'_g + PC'_g \qquad (107)$$

Therefore, for each CD, we find the total cost for route R. Then, we take the minimum of all these values and call it C_R. For all R, C_R is calculated. Hence, the set-covering model is as follows:

$$Z_R = \begin{cases} 1 \text{ if set } R \text{ is in the solution} \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

$$Minimize \sum_{R \in \mathcal{R}} C_R Z_R \qquad (108)$$

$$St. \sum_{R \in \mathcal{R}:(i,p) \in R} Z_R \geq 1 \text{ for each } (i,p) \in IP \qquad (109)$$

$$Z_R \in \{0,1\}, for\ each\ R \in \mathcal{R} \qquad (110)$$

However, finding all $R \in \mathcal{R}$ is almost impossible, especially when the instance becomes relatively large. That is why we start solving the set-covering problem with a limited number of columns, i.e. with $\mathcal{R}' \subset \mathcal{R}$. Then, after LP relaxation of the set-covering problem, we use the dual values of the LP-relaxed problem, and generate columns for the set-covering problem using a pricing problem, in each iteration of the column-generation algorithm. In the next section, we elaborate more on that algorithm.

The linear programming relaxation of the set-covering model, considering $\mathcal{R}' \subset \mathcal{R}$ instead of $\mathcal{R}$, is

$$Minimize \sum_{R \in \mathcal{R}'} C_R Z_R \qquad (111)$$

$$St. \sum_{R \in \mathcal{R}':(i,p) \in R} Z_R \geq 1\ for\ each\ (i,p) \in IP \qquad (112)$$

$$0 \le Z_R \le 1, for\ each\ R \in \mathcal{R}' \qquad (113)$$

4.4.4 Column-Generation Algorithm

We have a set-covering model, introduced in the previous section. Consider the dual variables of the LP relaxation of the set-covering model (π_{jp}). In fact, we assign a dual variable to each of the constraints in the set-covering model. These dual variables help us find out if there is still any R with nonnegative reduced cost. If yes, we solve the pricing problem using the values of the dual variables, and the optimal solution of the pricing problem is added to the set-covering model as a new column. If not, that means the optimal solution of the set-covering model is found, and the column-generation algorithm stops.

Now, let us define the pricing problem. Each time, the pricing problem must generate the route with a negative reduced cost for us. Therefore, first of all, the pricing problem should be written in a way that it produces only one route at each iteration. Second, the objective function of the pricing problem should be to minimize the reduced cost, so that the route with the smallest reduced cost can be found. The dual variables are in fact the profit of serving customers (j, p), so the total saving for route (g, k) is

$$\text{Total saving for } (g, k) = \Pi(g, k) = \Pi(g) = \sum_{\substack{(i=g_1) \\ i_{\vee(i \in I, i<j)}}} \sum_j \sum_p \pi_{jp} . z_{sijijp}^{gk} \qquad (114)$$

Consider the reduced cost function as $\bar{\mathbb{C}}(g) = \mathbb{C}(g) - \Pi(g)$. The algorithm terminates when all the reduced costs are positive. Therefore, using the reduced cost function, the pricing problem will be expressed as

$$Min\ \mathbb{C}(g) - \sum_{i \in I \cup \{g_1\}, i<j} \sum_j \sum_p \pi_{jp} . z^{gk}_{sijijp}$$

$$= \sum_{j \in I} \sum_{\substack{(i=g_1) \\ \vee (i \in I, i<j)}} \sum_{j' \in I} \sum_{\substack{(i'=g_1) \\ \vee (i'=g_2) \\ \vee (i' \in I, i'<j')}} \sum_{p \in P} b_p c''_{i'j'} d_{jp} z^{gk}_{siji'j'p} + R_{gk}$$

$$+ \sum_p \sum_j \sum_{\substack{(i=g_1) \\ \vee (i \in I, i<j)}} d_{jp} h_p \sum_{\substack{(i'=g_1) \\ \vee (i' \in I, i'<j)}} \sum_{j'=1}^{j} t_{i'j'} z^{gk}_{siji'j'p}$$

$$+ \sum_j \sum_{\substack{(i=g_1) \\ \vee (i \in I, i<j)}} \sum_p \frac{c'_{s(g,p),g}}{2C'_T} b_p d_{jp} z^{gk}_{sijijp} + \sum_j \sum_{\substack{(i=g_1) \\ \vee (i \in I, i<j)}} \sum_p \frac{t_{s(g,p),g} h_p d_{jp} z^{gk}_{sijijp}}{2}$$

$$- \sum_{\substack{(i=g_1) \\ \vee (i \in I, i<j)}} \sum_j \sum_p \pi_{jp} . z^{gk}_{sijijp} \qquad (115)$$

In addition, as mentioned before, the pricing problem must produce only one route for each sub-problem. Therefore, solution to the pricing problem does not necessarily satisfy the demands of all customers. As a result, we need to rewrite the constraints corresponding to designing the routes, such that each time only one route is generated without covering all the customers. Here, we present one set of constraints which is helpful to solve the model with a single product. Also, we provide two groups of constraints for a multi-product model. One group is built from only the existing set of variables. However, for the second group of constraints, an additional set of variables is introduced.

4.4.4.1 Constraints of the pricing problem for a single-product model

Apart from constraints (90) to (92) and (101) to (105), the following constraints should be added to the single-product model:

$$\sum_{\substack{(i<j)\\ \vee(i=g_1)}} z_{sijijp}^{gk} = \sum_{\substack{i>j\\ \vee(i=g_2)}} z_{sjijip}^{gk} \qquad j \in I \qquad (116)$$

$$\sum_{j} z_{sg_1jg_1jp}^{gk} + \sum_{j} z_{sg_2jg_2jp}^{gk} = 2 \qquad (117)$$

Constraint (116) indicates that the sum of edges entering j should be equal to the sum of edges leaving j. It includes that the degree of customer j can equal zero (demand is not necessarily satisfied). Constraint (117) forces the sub-problem to provide at least one route, and in addition, makes sure that the degree of the CD is 2.

4.4.4.2 Constraints of the pricing problem for a multi-product model

Except for constraint (117), all constraints defined for the single-product model are necessary for a multi-product model, too. However, those constraints are not sufficient. We now provide two additional sets of constraints. Either can be used in the pricing problem to design multi-product routes.

- *Constraints without any additional variables:*

For every $i, i' \in I \cup \{g_1, g_2\}; i < j; i' < j'; j, j' \in I\ ; p \in P; p' \in P$, we have:

$$z_{sijijp}^{gk} + z_{si'j'i'j'p'}^{gk} \leq z_{siji'j'p}^{gk} + 1 \qquad (118)$$

$$z_{sijijp}^{gk} + z_{si'j'i'j'p'}^{gk} \leq z_{siji'j'p'}^{gk} + 1 \qquad (119)$$

In constraints (118) and (119), if p is delivered on edge (i, j), and p' is delivered on edge (i', j'), $z_{siji'j'p}^{gk}$ and $z_{siji'j'p'}^{gk}$ are forced to be equal to 1. This means that if there is an edge (i, j) on the route that product p is going to traverse, and similarly, there is an edge (i', j') that on which product p' is travelling, p and p'must also traverse edges (i', j') and (i, j), respectively. The reason is that we have only a single route, hence one vehicle. Each product shipped on the route must be considered on every edge.

- *Constraints with additional variables:*

For any customer j we have:

$$\sum_p \sum_i z^{gk}_{sg_1ig_1ip} + \sum_p \sum_i z^{gk}_{sg_2ig_2ip} \leq \sum_p \sum_i z^{gk}_{sijijp} + Mv_j \qquad (120)$$

$$\sum_p \sum_i z^{gk}_{sijijp} \leq M(1 - v_j) \qquad (121)$$

$$v_j \in \{0,1\}, j \in I \qquad (122)$$

Suppose $v_j = 1$ for customer j. Then, the left-hand side of Constraint (121) equals zero. This means the degree of node j equals zero. Now suppose $v_j = 0$. Then, the degree of CD g (left-hand side of Constraint (120)) should not be greater than the degree of customer j. Based on Constraint (116), the degree of each node is an even number. The left-hand side of Constraint (120) cannot be less than the right-hand side because for each product, if any edge variable is positive, the route corresponding to that product must also be connected to the CD. This follows based on the assumption made at the beginning (and Constraint (116) is taking care of it): no customer can be connected to two nodes which have greater indices than the customer itself. Thus, at one point, the last customer is forced to be connected to the CD. Therefore, the left-hand side of Constraint (120) cannot be less than the right-hand side. That constraint must be satisfied as an equality.

The first set of constraints for the multi-product problem has fewer variables; whereas the latter has a smaller number of constraints. Their efficiencies become important when the size of the problem instance grows. For large instances, there are so many constraints in the first approach that the second becomes much more efficient.

In the next chapter, we provide some numerical results for the single-product model, and discuss the results by changing the ranges of randomly-produced parameters.

Chapter 5

Computational Results

In this chapter, the results of running several instances of the single-product model are presented. The Column Generation algorithm is programmed in Matlab R2009b, and Gurobi 5.6.3 is used as the optimization software. All experiments are done on a system with 16 GB RAM and 3.40 GHz CPU. CPU times are in seconds.

We generate all parameters randomly. Ranges of the parameters are somewhat similar to those of Berman and Wang (2006). Let $U[a, b]$ be the Uniform distribution on the interval $[a, b]$. Cartesian coordinates of customers are created from $U[1,1000]$. Then, coordinates of CDs are produced from $U[1000,1500]$, and suppliers are located using $U[1500,2000]$. Euclidean distances are calculated between each supplier and CD, CD and customer, and all pairs of customers. The transportation cost between nodes is linearly proportional to the corresponding distance.

The transportation time between any two nodes is equal to the value of the corresponding transportation cost divided by 4000. Customer demands are generated from $U[10{,}100]$. The capacity of each vehicle is 1000 units. h_p and b_p are drawn from $U[1{,}10]$.

In Table 3, the results of the model with two suppliers are shown. The number of CDs varies from 1 to 4; the number of customers in the system is between 2 and 15. In this table, the transportation cost between each supplier and CD, and also between each CD and any customer equals the corresponding distance. However, the transportation cost between each pair of customers is 0.01 times the respective distance. We elaborate on this later.

For every combination of (I, K, S) in Table 3, an initial 10 replications are run, with the average of those replications shown in each row of that table. In addition, results are calculated for each set of parameters for the model in which only direct shipment is allowed. The results with vehicle routing are compared to the model with direct shipment only, by calculating the percentage improvement of the former relative to the latter. Table 3 also presents the CPU times and the total number of edges the algorithm needs to explore to find the optimal routes. As can be seen in the last column, the cost savings usually increase as the number of customers becomes larger. Fig. 5 shows the percentage improvement in cost for different numbers of customers.

Table 4 exhibits the results for instances with the same sizes as in Table 3, except that the number of suppliers is increased to 4. Note that the number of suppliers does not affect the CPU time since Theorem 2 helps us to choose the sources of each CD in advance. By increasing the number of suppliers, their assignment to CDs can be more flexibly done. Hence, the total shipping cost from suppliers to CDs might decrease. However, the additional suppliers do not slow down the algorithm.

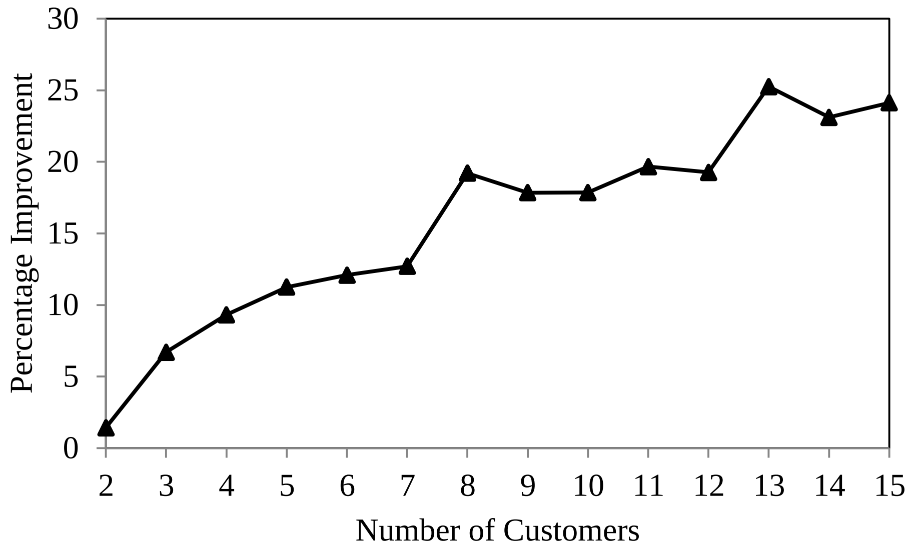

Figure 5 – Improvement of routing over direct shipment when $|G| = 3$ and $|S| = 2$

Row	$\lvert G \rvert$	$\lvert I \rvert$	Avg. no. of routes	No. of total edges	Avg. CPU Time	Avg. percentage improvement of routing over direct shipment
1	1	2	1.7	6	0.10	4.21
2		3	1.9	11	0.19	10.90
3		4	2.1	18	0.27	10.37
4		5	2.5	27	0.75	10.66
5		6	2.65	38	2.34	11.81
6		7	2.41	51	7.14	18.02
7		8	2.75	66	11.18	15.03
8		9	3.25	83	20.03	16.05
9		10	2.77	102	39.31	19.11
10		11	2.95	123	65.59	20.43
11		12	2.9	146	133.2	23.00
12		13	2.92	171	284.1	22.08
13		14	3	198	533.5	20.00
14		15	2.85	227	1024	23.97
15	2	2	1.5	10	0.14	6.64
16		3	1.9	16	0.34	3.41
17		4	1.66	24	0.87	10.94
18		5	2.7	34	1.87	7.03
19		6	2.4	46	4.76	10.09
20		7	3	60	10.32	16.22
21		8	3.28	76	19.45	10.09
22		9	2.6	94	46.41	20.55
23		10	3.6	114	51.44	16.22
24		11	3.37	136	103.2	20.41
25		12	3.15	160	254.1	18.94
26		13	3.4	186	575.5	24.21
27		14	3.6	214	716.9	20.56
28		15	3.55	244	1542	25.85

Table 3 – Average of results of instances with up to 15 customers and 4 CDs, considering 2 suppliers for 10 replications

(Continued on next page)

Row	$\|G\|$	$\|I\|$	Avg. no. of routes	No. of total edges	Avg. CPU Time	Avg. percentage improvement of routing over direct shipment
29	3	2	1.5	14	0.18	1.43
30		3	1.8	21	0.44	6.70
31		4	1.8	30	1.14	9.32
32		5	2.15	41	2.41	11.25
33		6	2.9	54	6.85	12.09
34		7	2.95	69	15.07	12.70
35		8	2.85	86	30.25	19.21
36		9	3.15	105	49.51	17.85
37		10	3.07	126	94.18	17.86
38		11	2.93	149	189.7	19.67
39		12	3.01	174	347.5	19.26
40		13	3.05	201	772.0	25.26
41		14	3.6	230	1162	23.11
42		15	3.35	261	2464	24.13
43	4	2	1.5	18	0.22	6.08
44		3	1.7	26	0.59	5.33
45		4	1.6	36	1.23	13.06
46		5	2.75	48	4.07	8.75
47		6	2.8	62	8.29	11.13
48		7	2.25	78	21.72	19.34
49		8	2.6	96	42.01	21.03
50		9	3	116	66.59	17.39
51		10	2.8	138	123.27	21.15
52		11	3.5	162	198.72	14.45
53		12	3.35	188	457.22	22.40
54		13	3.2	216	990.59	23.56
55		14	3.8	246	1632	23.92
56		15	3.75	278	2243	25.07

Table 3– Average of results of instances with up to 15 customers and 4 CDs, considering 2 suppliers for 10 replications

(Continued from previous page)

Row	$\lvert G \rvert$	$\lvert I \rvert$	Avg. no. of routes	No. of total edges	Avg. CPU Time	Avg. percentage improvement of routing over direct shipment
1	1	2	1.6	8	0.26	0.80
2		3	1.7	13	0.62	0.59
3		4	2	20	1.28	0.74
4		5	1.95	29	2.89	1.03
5		6	2.5	40	4.45	1.01
6		7	2.9	53	11.45	0.99
7		8	2.55	68	22.76	1.27
8		9	2.46	85	25.43	20.31
9		10	3.35	104	40.44	19.24
10		11	3.1	125	70.34	22.55
11		12	3.05	148	108.17	26.66
12		13	3.20	173	268.26	26.41
13		14	3.15	200	467.13	24.48
14		15	3.12	229	1013.91	27.03
15	2	2	1.6	14	0.42	1.00
16		3	2	20	1.02	0.23
17		4	1.8	28	2.64	1.19
18		5	2.15	38	4.66	1.02
19		6	2.35	50	10.10	1.05
20		7	2.7	64	25.66	1.48
21		8	3.05	80	37.26	1.22
22		9	2.5	98	42.57	20.49
23		10	2.9	118	59.94	17.99
24		11	3.75	140	109.22	18.23
25		12	3.26	164	270.13	20.91
26		13	3.45	190	491.38	23.40
27		14	3.3	218	888.00	26.43
28		15	3.26	248	1616.94	26.35

Table 4 – Average of results of instances with up to 15 customers and 4 CDs, considering 2 suppliers for 10 replications

(Continued on next page)

Row	$\lvert G \rvert$	$\lvert I \rvert$	Avg. no. of routes	No. of total edges	Avg. CPU Time	Avg. percentage improvement of routing over direct shipment
29	3	2	1.5	20	0.45	2.46
30		3	1.2	27	1.35	1.23
31		4	1.85	36	3.89	0.84
32		5	2.5	47	8.02	1.32
33		6	2.5	60	16.73	1.13
34		7	2.2	75	33.25	1.61
35		8	3.22	92	59.04	1.22
36		9	3.2	111	47.28	18.35
37		10	3.2	132	88.80	23.00
38		11	3.6	155	139.30	18.89
39		12	2.86	180	373.04	25.96
40		13	2.95	207	876.21	24.88
41		14	3.07	236	1259.06	28.00
42		15	4.13	267	1512.85	25.72
43	4	2	1.7	26	0.60	0.33
44		3	1.6	34	1.79	0.77
45		4	1.8	44	5.96	1.46
46		5	1.9	56	12.00	1.32
47		6	2.15	70	20.84	1.19
48		7	2.65	86	40.09	1.07
49		8	2.26	104	73.47	1.93
50		9	2.65	124	94.38	23.96
51		10	3.09	146	133.55	21.56
52		11	2.99	170	244.63	22.75
53		12	3.3	196	406.89	23.46
54		13	3.25	224	974.44	26.38
55		14	3.78	254	1489.53	24.68
56		15	2.91	286	2964.76	27.20

Table 4– Average of results of instances with up to 15 customers and 4 CDs, considering 2 suppliers for 10 replications

(Continued from previous page)

We stated at the start of this chapter that the transportation cost between every two customers is a multiple of the corresponding distance. Table 5 shows the results of running the algorithm for different values of that multiple. When the multiple is relatively high, meaning that the transportation cost per unit distance of edges connecting customers exceeds the unit transportation cost of other edges, direct shipment is preferable to routing. However, routing becomes more advantageous for a lower multiple.

This can be interpreted as follows. Direct shipment to clusters of customers that are close to CDs, costs less than routing. Routing is chosen as the shipping strategy for the customers that are further from CDs. The reason is the important role of pipeline inventory cost. This cost is considered on all edges that are to be traversed until the shipment is delivered to its destination.

If the transportation time (which is a multiple of transportation cost) is high, then direct shipment seems to be more beneficial. Fig. 6 (for 10 customers and 5 CDs) shows how the number of routes increases with an increase in the transportation costs of edges connecting pairs of customers. In this sensitivity analysis, no other parameters were varied.

Row	Multiple of c_{ij}	No. of routes	CPU Time
1	2	10	32.49
2	1.6	10	33.55
3	1.2	10	35.97
4	1	10	38.06
5	0.8	10	42.41
6	0.6	7	83.66
7	0.4	5	121.3
8	0.2	4	129.1
9	0.19	4	129.4
10	0.15	4	201.0
11	0.11	4	170.8
12	0.07	3	161.7
13	0.03	2	234.8
14	0	2	113.9

Table 5 – Variation of number of routes with multiple of c_{ij}

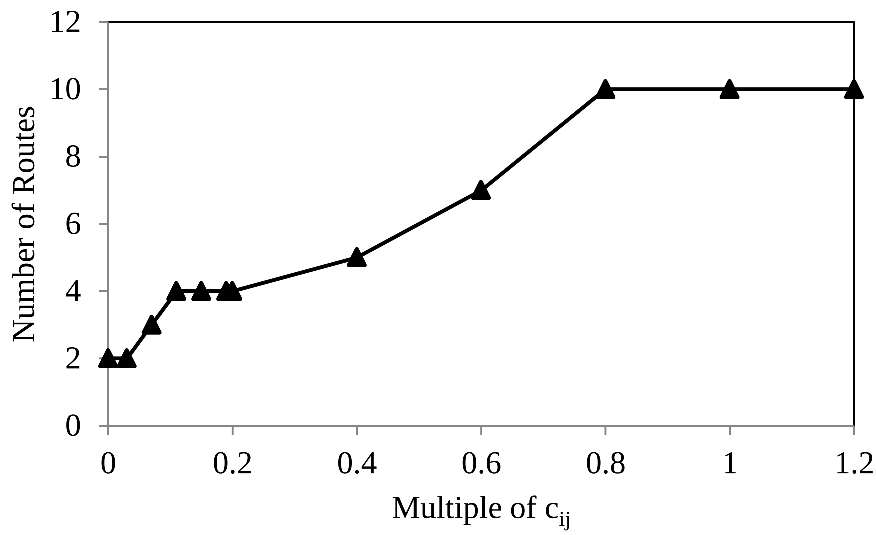

Figure 6 – Variation of number of routes with multiple of c_{ij}

In Table 6, we provide the results of changing the ratio of $\frac{b_p}{h_p}$. The number of routes does not change significantly. Keeping h_p constant, the number of routes in the optimal solution decreases only when the unit volume of product becomes too small. This can be interpreted as, when that ratio is relatively small, which means h_p is larger than b_p, the objective function tries to keep lower levels of inventory at the customers. Therefore, the quantities delivered to customers in each visit should become smaller. Hence, larger numbers of customers are visited on each route, and the number of routes decreases. Fig. 7 also shows the results of Table 6.

Row	$\boldsymbol{\frac{b_p}{h_p}}$	**No. of routes**	**CPU Time**
1	10	7	70.12
2	7	7	74.41
3	4	7	96.66
4	1	7	83.88
5	0.8	7	80.38
6	0.6	7	95.89
7	0.4	6	78.14
8	0.2	6	74.59
9	0	5	63.44

Table 6 – Variation of number of routes with $\frac{b_p}{h_p}$

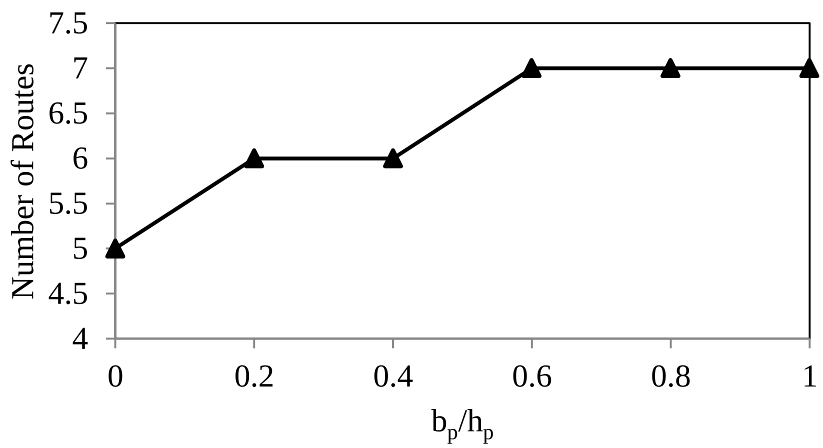

Figure 7 – Variation of number of routes with $\frac{b_p}{h_p}$

Table 7 demonstrates the findings when the algorithm is run for larger instances. We are able to solve the problem up to 25 customers. Incidentally, it turns out that, although the set-covering model is an LP relaxation, 90% of the results are integer. Each instances of Table 7 are run only once, and they were all integer. In addition, instances with up to 9 customers and 4 CDs resulted in integer solutions. Also, for 80% of the instances with 10 to 15 customers and up to 4 CDs were integer.

Row	$\lvert \boldsymbol{I} \rvert$	$\lvert \boldsymbol{G} \rvert$	$\lvert \boldsymbol{S} \rvert$	**No. of routes**	**CPU Time**	**No. of edges**
1	15	5	1	3	3632	290
2	20	5	1	4	74767	485
3	25	1	1	4	184433	626

Table 7 – Results of larger instances

Chapter 6

Conclusions and Future Work

In this thesis, the design of minimum-cost routes with optimal shipment frequencies was considered, for the problem in which suppliers satisfy customer demands for several product types through a set of CDs.

In the objective function, transportation cost and pipeline inventory cost of the whole supply chain, plus inventory carrying cost at the customers, were taken into account. Our goal was to minimize the total cost. As expected, in a good solution, reasonable tradeoffs are found between those inventory levels and the costs of shipping (transportation + pipeline inventories).

To the best of our knowledge, pipeline inventory cost has not been addressed in the literature on IRP (the Inventory Routing Problem). In addition, the idea of shipment frequencies introduced by Berman and Wang (2006) for a supply chain with direct shipment strategy was not applied to the supply chains with vehicle routing.

We presented a node-based formulation for the problem that we had defined. We demonstrated that each CD can be treated as a set of “dummy” CDs, each of which can be assigned to at most one route. This helped to formulate the model as a nonlinear mixed integer programming problem, in which the cost of each route is considered separately.

The model was solved using a column generation algorithm. This was after demonstrating some analytical results that enabled us to rewrite the model as an edge-based formulation. We linearized the nonlinear objective function by using those results and the decomposition of the objective function over each route. Based on the preceding result, we presented a pricing problem that produced a single route at each iteration of the column-generation algorithm.

In addition, we redefined the constraints after writing the objective function of the pricing problem, because those constraints needed to be defined such that only one single route could be built connecting a sub-set of customers to a given CD. A set-covering model was presented to choose the optimal (least-cost) routes generated by the pricing problem.

We formulated our problem for both single-product and multi-product scenarios. Results for a single product model were presented in Chapter 5. We observed that the proposed algorithm was able to provide an integer exact solution for 90% of the instances considered. Results for all instances with a number of customers smaller than 10 were integer. Also, 80% of the results for instances with 10 to 15 customers turned out to be integer.

Table 7 presented the larger instances all with 15 or more customers. Each instance had its own set of (random) cost parameters; the model was run just for that single case. The corresponding results were all integer. The largest instance that we could solve to optimality had

626 edges. We were able to solve the problems with up to 300 edges within only one hour of CPU time.

We demonstrated that routing is preferred to direct shipment when customers are relatively far from CDs because of the high pipeline inventory cost of direct shipment. In addition there are fewer routes for a smaller ratio $\frac{b_p}{h_p}$. This is because, as the objective function tries to decrease the inventory carrying cost at the customers, the number of customers on each route increases. The quantity delivered to each customer at each visit thus decreases.

For future work, we can consider removing the assumption of connecting the customers in an ascending order. This will permit more general routing policies. A greedy heuristic could be provided to sort the customers, such that solution of the column-generation algorithm would be close to optimal. One way to sort the customers before applying the column-generation algorithm is to solve a Travelling Salesman Problem (TSP) for all the customers, and continue with the resulting sequence. This might not give the exact optimal solution since the corresponding sequence might not be optimal. However, it can give a good insight on the ordering of customers. In addition, a new index can be added to decision variables. That index would correspond to edges, and would show the place of each edge on a particular route. In this case, the algorithm could result in an exact optimal solution, assuming that instances of the size we wish to solve are within the algorithm's capability. Naturally, optimality would have to be proven.

Based on Fig. 5, cost savings mostly increase for increasing size of the problem instances. However, that increase is not monotonic. This could be due to randomness of the parameters of the problem. We took the average of 10 replications for each instance-size. Running more replications and introducing a proper confidence interval will be helpful to have the graph represent the more-correct cost savings, which we believe will be closer to monotonic.

In our computational results, we compare two scenarios. These have direct shipments from CDs to customers, and employ routes originating only from CDs, respectively. As future work, we could also include shipments in which routes originate at the suppliers, and visit customers without going through CDs.

We initialized the set-covering model with a set of columns regarding direct shipment to all customers from random CDs. An algorithm could be introduced to find a better initial feasible solution for the set-covering model. Such a solution generator would help to find the optimal solution with fewer column generation iterations, and make the whole optimization process faster.

To solve those instances that result in non-integer solutions for the set-covering model, a branch-and-price algorithm would be helpful. Alternatively, a proper approximation algorithm, such as randomized rounding proposed by Raghavan and Tompson (1987), and improved by Slavik (1997) and Srinivasan (1995) could be developed.

This model can be extended to a multi-period problem, where customers' inventories in each period are taken into account. Capacitated CDs, availabilities of the suppliers, and stochastic demands at customers can also be considered. Locations of CDs could also be interesting to be considered as decision variables. This is because, as we observed, those CD locations play an important role in choosing a proper shipment strategy.

References

Abouee-Mehrizi H., Berman O., Baharnemati M.R. (2014) Designing Production-Inventory-Transportation Systems with Capacitated CDs. *Transportation Science*, 48(1):121-135

Andersson H., Hoff A., Christiansen M., Hasle G., Lokketangen A. (2010) Industrial Aspects and Literature Survey: Combined Inventory Management and Routing. *Computers & Operations Research* 37(9): 1515-1536

Archetti C., Bertazzi L., Laporte G., Speranza M.G. (2007) A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. *Transportation Science* 41(3):382-391

Baldacci R., Mingozzi A., Roberti R., Calvo R.W. (2013) An Exact Algorithm for the Two-Echelon Capacitated Vehicle Routing Problem. *Operations Research*, 61(2): 298-314

Belle J.V., Valckenaers P. Cattrysse D. (2012) Cross-Docking: State of the Art. *Omega: Special Issue on Forecasting in Management Science* 40(6): 827-846

Berman O., Wang Q. (2006) Inbound Logistic Planning: Minimizing Transportation and Inventory Cost. *Transportation Science* 40(3): 287-299

Buijs P., Vis I.F.A., Carlo H.J. (2014) Synchronization in Cross-Docking Networks: A Research Classification and Framework. *European Journal of Operations Research*, In Press

Clarke G., Wright J.W. (1964) Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12(4): 568-581

Coelho L.C., Cordeau J.F., Laporte G. (2014) Thirty Years of Inventory Routing. *Transportation Science* 48(1):1-19

Coelho L.C., Laporte G. (2013) The Exact Solution of Several Classes of Inventory-Routing Problems. *Computers & Operations Research* 40: 558–565

Cordeau J.F., Gendreau M, Laporte, G., Potvin, J., Semet, F. (2002) A Guide to Vehicle Routing Heuristics. *Journal of the Operational Research Society*, 53(5): 512-522

Cordeau J.F., Laporte G., Savelsbergh M.W.P., Vigo D. (2007) Vehicle Routing. In Barnhart C., Laporte G., Editors, *Transportation, Handbooks in Operations Research and Management Science* 14: 367-428. Elsevier, Amsterdam

Cordeau J.F., Laporte G. (2004). Tabu Search Heuristics for the Vehicle Routing Problem. In: Rego C., Alidaee B., Editors, *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search* 145-163, Kluwer Academic, Boston

Dantzig G.B., Ramser J.H. (1959) The Truck Dispatching Problem. *Management Science*, 6(1): 80-91

Gümüş M., Bookbinder J.H. (2004) Cross-Docking and its Implications in Location Distribution Systems. *Journal of Business Logistics*, 25(2): 199–228

Laporte G., Nobert Y. (1987), Exact Algorithms for the Vehicle Routing Problem, In: Martello S., Laporte G., Minoux M. and Ribeiro C., Editors, *Surveys in Combinatorial Optimization* 147-184, North-Holland, Amsterdam

Mingozzi A., Roberti R., Toth P. (2013) An Exact Algorithm for the Multitrip Vehicle Routing Problem. *INFORMS Journal on Computing* 25(2):193-207

Muter I., Cordeau J.F., Laporte G. (2014) A Branch-and-Price Algorithm for the Multidepot Vehicle Routing Problem with Interdepot Routes. *Transportation Science*, forthcoming

Padberg M., Rinaldi G. (1991) A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems. *SIAM Review* 33(1): 60-100

Raghavan P., Tompson C.D. (1987) Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica,* 7(4): 365-374

Santos F.A., Mateus G.R. (2011) A Branch-and-Price Algorithm for a Vehicle Routing Problem with Cross-Docking. *Electronic Notes in Discrete Mathematics,* 37: 249-254

Slavík P. (1997) A Tight Analysis of the Greedy Algorithm for Set Cover. *Journal of Algorithms,* 25(2): 237-254

Srinivasan A. (1995) Improved approximations of packing and covering problems. *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, 268-276

Stephan K., Boysen N. (2011) Cross-Docking *Journal of Management Control* 22(1): 129-137

Wen M., Larsen J., Clausen J., Cordeau J.F. (2009) Vehicle Routing with Cross-Docking. *The Journal of the Operational Research Society*, 60(12): 1708-1718