Location Problems in Supply Chain Design: Concave Costs, Probabilistic Service Levels, and Omnichannel Distribution

Aditya Malik

A Thesis
In
The John Molson School of Business

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Business Administration) at
Concordia University
Montreal, Quebec, Canada

July 2021

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Aditya Malik

Entitled: Location Problems in Supply Chain Design: Concave Costs, Probabilistic Service Levels, and Omnichannel Distribution

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Business Administration)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Saif Ullah	Chair
Dr. Dmitry Krass	External Examiner
Dr. Okan Arslan	External to Program
Dr. Satyaveer Chauhan	Examiner
Dr. Brigitte Jaumard	Examiner
Dr. Navneet Vidyarthi	Thesis Co-Supervisor
Dr. Ivan Contreras	Thesis Co-Supervisor

Approved by

Dr. Cedric Lesage, Graduate Program Director

July 20, 2021 Dr. Anne-Marie Croteau, Dean of The John Molson School of Business

Abstract

Location Problems in Supply Chain Design: Concave Costs, Probabilistic Service Levels, and Omnichannel Distribution

Aditya Malik, Ph.D.

Concordia University, 2021

Location of facilities such as plants, distribution centers in a supply chain plays critical role in efficient management of logistics activities. Real-life supply chains are generally large in size with multiple echelons, prone to disruptions and uncertainties, and constantly evolving to meet customer demands in a fast and reliable way. Therefore, it is quite challenging to identify these locations while balancing the trade-off between costs and service levels. In this thesis, we investigate three supply chain design problems addressing various issues that complicate the location of facilities in a supply chain.

The first paper investigates a multilevel capacitated facility location problem. Such problems commonly arise in large scale production-distribution supply chains with plants at one echelon, and distribution centers / warehouse at another, and there is hierarchy of flow between facilities and to the end customers such as retail stores. The operating costs at facilities and transportation costs on arcs are assumed to be concave. The concave functions model economies of scale in operations (such as production, handling, transportation) performed at large scale and emission of green house gases from transportation activities. The mathematical model for our problem is nonlinear (concave) for which we present two formulations. The first formulation is a prevalent mixed-integer nonlinear program, and second is a purely nonlinear programming problem. Extensive computations are performed to measure the efficiency of two formulations, and managerial insights are provided to understand the behavior of the model under different scenarios of concavities.

The second work focuses on e-commerce supply chains that have a common objective of providing fast and reliable deliveries of customers' orders. The order delivery time primarily depends on the time taken to process the order at the facilities and travel time from facilities

to customers. These two times are uncertain in practice, therefore, to capture the combined effect of both uncertainties, we introduce a mathematical model with a requirement that all customer orders should be delivered within a committed time with some probabilistic guarantee. The problem is formulated as a dynamic (multiperiod) capacitated facility location problem with modular capacities. The probabilistic service level constraints make the problem nonconvex. We present two linear binary programming reformulations, and develop an exact branch-and-cut algorithm utilizing the reformulations to solve large size instances. We also include sensitivity analysis to study the change in network configuration under various modeling parameters.

An increase in online sales every year is driving many brick-and-mortar retailers to follow an omni-channel retailing approach that would integrate their online sales channel with store sales. Omnichannel retailing requires a considerable change in current practices. For instance, a retailer generally decides if there is a need of new distribution facilities, which stores should be used as fulfillment centers as well, where to keep safety stocks, from where to serve online demand, among others. To study these aspect, in the third paper, we propose a novel mathematical model for the design of omnichannel distribution network along with allocation of safety stock to the facilities. The original problem is nonlinear which can be reformulated as conic quadratic mixed integer programming problem. The problem is solved using a branch-and-cut solution algorithm. Further, we present several managerial insights related to fulfillment and safety stock decisions using a small example.

To my family

The larger the island of knowledge, the longer the shoreline of wonder - Ralph W. Sockman

Acknowledgements

'The pursuit of PhD is an enduring daring adventure'- Lailah Gifty Akita

These past five years of PhD have been an emotional roller coaster ride for me, and I would like to thank everyone involved for their unrelenting support and faith in me.

First and foremost, I would like to express my sincere gratitude to my advisors Dr. Navneet Vidyarthi and Dr. Ivan Contreras. Dr. Vidyarthi's continuous words of encouragement and guidance always kept me on the right path, and Dr. Contreras's ingenious ideas gave a fresh perspective to my dissertation. Their patience and synergies have always motivated me to perform to the best of my capabilities. Their subject matter expertise, commitment to work and positive attitude is greatly instrumental in the production of this thesis, and in shaping my aptitude as a researcher.

I embarked on this journey of optimization working as a Research Associate for Dr. Jayaswal (Associate Professor, Indian Institute of Management, Ahmedabad) and Dr. Saurabh Chandra (Associate Professor, Indian Institute of Management, Indore). I deeply thank them for the opportunity they gave me and the skills I acquired under their supervision. The experience I gained here provided me a strong foundation and a head start for my PhD research.

I am forever grateful to my parents and family for their constant support and words of encouragement whenever I used to feel low. They are the reason why this dream became a reality. I have acquired the virtue of sincerity and commitment from my mother, and my father has always taught me to be ethical towards my work. Undeniably, this thesis would not have been in the shape it is today without the support of my loving wife, Arpita. She is my editor, proof-reader, and the first person I would approach to discuss any technical aspect of my projects. Arpita, I must acknowledge that your charm and positive attitude helped me maintain my sanity. My special thanks to my sister Aastha and my friend Jasvinder. The long, non-academic and fun-filled conversations I had with you guys considerably reduced my stress.

Sayed and Carlos, my lab-mates and my friends away from home, I would like to thank you both for all the insightful discussions we had. Other than my curiosity for science, this research would not have been possible without the financial assistance provided by the John Molson School of Business, and my advisors.

Finally, I want to thank all those people who showed interest and contributed towards this research in one or the other way, but their names could not be included here. Forever grateful!!

Contribution of Authors

This dissertation is presented as a manuscript-based thesis. It contains three articles that are either under revision in different journals or in preparation for submission to a journal. The articles were submitted in the following chronological order. The first article titled "Multilevel Capacitated Discrete Location with Concave Costs" has received major revision from the journal *Transportation Science*. The second manuscript titled "Stochastic Facility Location with Probabilistic Service Level Constraints on Delivery Times" is ready for submission to the journal *Operations Research*. Finally, the third manuscript titled "Demand Allocation, Inventory Positioning and Distribution Network Design in Omnichannel Retailing" is planned to be submitted in August 2021 to the journal *Production and Operations Management*.

All three manuscripts are co-authored with Dr. Ivan Contreras and Dr. Navneet Vidyarthi, as supervisors. The author of this thesis acted as the principal researcher with the corresponding duties such as development of formulations and algorithms as well as programming of solution methods, mathematical proofs, analysis of computational results, and writing the first drafts of the articles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A supply chain is a network of entities such as, suppliers, manufacturers, warehouses, transporters and retailers, that work in consonance to satisfy the demands for goods and services of end customers. To stay profitable in today's fiercely competitive economy, firms are always looking for ways to make their supply chain more efficient. For many businesses, supply chain costs vary as 10-20% of their revenues (OliverWyman [97]). Therefore, a lot of firms focus on minimizing supply chain costs to increase profitability. One of the ways to reduce this cost is through strategic placement of facilities and transportation routes in the supply chain networks (Melo et al. [95]). Most of the literature assume that production, warehousing and transportation costs in the supply chain are linear. However, in consideration of factors such as economies of scale, and the fact that environmental costs from transportation activities are known to be concave (Elhedhli and Merrick [40]), we model variable costs in the supply chain as concave. Thus, we study a facility location model to optimize the total cost of configuring a *large scale production-distribution supply chain with concave costs*.

A production-distribution system is an example of a classical supply chain in which demand is represented by sales at large brick-and-mortar (B&M) retail stores, and customers travel to these B&M stores to satisfy their demand. However, the advent of internet has brought forth an e-commerce retail model, wherein businesses directly deal with individual customers through online retail stores (e.g., websites, mobile applications). The products are directly delivered to customers' homes in e-commerce business practices. For online customers the quick, low cost (preferably free) and reliable delivery of the orders are some important considerations while shopping online. For instance, a survey by Shipbob[1] reports that 74% of customers are more inclined towards retailers providing free shipping, and 38% customers would not like to deal with the retailer if the orders were delayed in the past. Therefore, to meet such expectations, location of facilities is critical in an e-commerce supply chain. These locations not only affect the delivery time commitments but also the overall supply chain costs. With such considerations in mind, we also present a problem that designs *an e-commerce supply chain* while minimizing the total costs and providing guarantees on delivery times.

The increasing popularity of online shopping has pushed many B&M retailers to include online sales channel with in-store sales. This is commonly known as 'Omnichannel Sales' strategy. The integration of offline and online sales channels provides frictionless shopping experience because customers can browse products online and buy in store, and vice-versa. To improve shopping experience further, the shoppers are also provided with multiple options to serve their demand, such as, home delivery, in-store, curb side or designated pick-up point options. This strategy is beneficial for retailers too. They can leverage their well-established network of B&M stores to provide multiple fulfillment options and to improve their last mile delivery that can lead to lower delivery costs and higher customer service. Such opportunities open several questions for the retailers to successfully implement omnichannel retailing. For instance, B&M retailers must decide which stores to use in fulfilling online demand, whether they need additional distribution facilities to reach online customers and what would the impact be on costs, how much inventory should be allocated for each sales channel, and at which locations, etc. These questions motivate us to investigate a problem for designing *distribution supply chain for an omnichannel retailer*.

[1]https://www.shipbob.com/blog/ecommerce-shipping/

Therefore, based on the problems studied, the contributions of this thesis can be classified as follows.

- Problem modelling.
  - To address a practical scenario of concave costs in facility location problems.
  - To incorporate service level requirements for online order delivery times in e-commerce supply chain
  - To provide a new model for omnichannel distribution that integrates fulfillment planning for in-store and online demand with home delivery option.
- Algorithmic development.
  - To investigate mixed integer nonlinear and purely nonlinear programming models to solve capaciated facility location problems with concave costs.
  - To provide linear reformulations and develop branch-and-cut algorithms to solve a dynamic capacitated facility location problem with stochastic demand and uncertain order processing and travel times.
  - To exploit underlying properties of mixed integer nonlinear model for Omnichannel distribution network design problem to provide a conic quadratic mixed integer programming reformulation.
- Managerial insights
  - Analysis of impact of varying concave costs parameters on the location of manufacturing plants and distribution centers and allocation of customers to distribution centers
  - Evidence of potential benefits of studying dynamic capacity planning problem from the perspective of service level commitments.

– Showcasing the benefits of studying integrated planning (using distribution centers and brick-and-mortar stores) to fulfill in-store and online demand with home delivery option in an omnichannel distribution supply chain.

The rest of this thesis is organized as follows. Chapter 2 presents two formulations for a multilevel capacitated location problem with concave costs. The first formulation is a prevalent mixed-integer nonlinear programming problem, whereas second one is purely nonlinear programming problem. Extensive computations are done to compare the performance of two formulations under different cost and capacity scenarios. In Chapter 3, we discuss a dynamic (multiperiod) capacitated facility location problem with modular capacities and service level restrictions on order delivery time. The problem is formulated as a binary nonconvex program for which two linear reformulations are presented, and a branch-and-cut based exact solution algorithm is developed. We compare the computational performance of three versions of the branch-and-cut algorithm. Chapter 4 presents an Omnichannel distribution network design problem with integrated planning for in-store and online demand using a common set of facilities. We exploit properties of the problem to present a conic quadratic mixed integer programming reformulation. A branch-and-cut algorithm is developed for the reformulated problem. Extensive computations are performed by varying model parameters and costs. Finally, conclusions are presented in Chapter 5.

Chapter 2

Multilevel Capacitated Discrete Location with Concave Costs

Abstract

In this paper, we study a general class of multilevel capacitated discrete location problems with concave costs. The concavity arises from the economies of scale in production, inventory or handling at the facilities and from consolidation of flows for transportation and transshipment on the links connecting the facilities. Given the discrete nature of the problem, it is naturally formulated as a mixed-integer nonlinear program that uses binary variables for locational decisions and continuous variables for routing flows. We present an alternative formulation that only uses continuous variables, resulting in a purely nonlinear program with a concave objective function. We present an exact branch-and-bound algorithm to optimally solve large-scale instances of the considered problems. We present algorithmic enhancements such as alternative branching strategies and variable fixing, to improve the convergence of the algorithm. Extensive computational experiments are performed to evaluate the strength of the two formulations and the performance of the exact algorithms. Results obtained on large-scale instances with up to 2,250 customers and 150 potential facilities, and two levels

under different costs and capacity scenarios confirm the effectiveness of the purely nonlinear formulation. We also present a sensitivity analysis on an instance considering the 3,109 counties in the contiguous USA, to understand the impact of varying economies of scale in operations at the facilities and in transportation links connecting the facilities on the location and allocation decisions.

2.1 Introduction

Discrete location problems (DLPs) are central problems in location science with applications ranging from supply chain management (Melo et al. [94]), public policy (Salman and Yücel [111]), health care (Ahmadi-Javid et al. [7]), and telecommunications (Fortz [49]), among many others (Drezner and Hamacher [36], Laporte et al. [80]). Generally speaking, DLPs seek to determine the location of facilities from a potential discrete set and to efficiently allocate customers to open facilities to satisfy customer demands, while optimizing a given objective function. Depending on the application and considered objective (costs, service quality, profits), a wide range of DLPs have been studied. For instance, typical models with a cost-based objective are: *fixed-charge facility location* (Kuehn and Hamburger [79], Fernández and Landete [45]), which can be *uncapacitated* (Fischetti et al. [47]) or *capacitated* (Gadegaard et al. [51]), *p-median problems* (Hakimi [61], Daskin and Maass [33]), *uncapacitated p-location problems* (Ortiz-Astorquiza et al. [100]); whereas models optimizing service quality are: *p-center* (Hakimi [61], Kramer et al. [77]), *set covering* (Toregas et al. [119], García and Marín [55]), and *maximum covering problem* (Church and ReVelle [26], Máximo et al. [92]).

This paper focuses on an important class of DLPs arising in the design of hierarchical systems. *Multilevel facility location problems* (MFLPs) deal with the location of interacting facilities at different levels of a hierarchical system. Applications of MFLPs arise in a wide variety of contexts such as production-distribution systems, health care systems, telecommunications systems, urban and air transportation systems, and cargo and postal delivery systems. We refer to Şahin and Süral [108], Farahani et al. [43], Ortiz-Astorquiza et al. [99], and Contreras and Ortiz-Astorquiza [29] for detailed surveys on MFLPs.

We study a general class of MFLPs denoted as *multilevel capacitated facility location problems with concave costs* (MCFLP-Cs), which can be defined as follows. Let I be the set of customers, V be the potential set of facilities partitioned into m levels, and the facilities at every level 1 to m are assumed to be capacitated. We consider a fixed setup cost for opening a facility and a variable operating cost at each facility modeled via a concave function of the amount of flow passing through the facility. Transportation costs are also assumed to be modeled with concave functions of the amount of flow routed though each link. We assume a *single-flow* pattern in which flows start from facilities at the highest level m and pass through all levels until they arrive at their demand points at the customer level. Moreover, we consider the possibility of transshipment flows. That is, a facility at level $\ell \in \{1, 2, .., m-1\}$ can send flows to facilities located at the same level ℓ. All customers receive shipments only from facilities opened at the first level. We assume a multiple allocation strategy in which customers can receive demand from more than one facility at level 1. The MCFLP-C consists of selecting a set of facilities to open at each level, such that the incoming flow on each facility satisfies its capacity limitations, while ensuring that all demand is met. The objective is to minimize the total fixed and variable costs to open and operate a set of facilities, and the variable transportation cost for all used links.

Potential applications of the MCFLP-C arise in the design and reconfiguration of production distribution systems in which a two-level hierarchical structure is used. Let V_1 and V_2 be the sets of potential locations among which production facilities and warehouses need to be opened. Production, warehousing and distribution costs can be modeled as concave functions of the quantities produced, stored and distributed, respectively. These concave functions provide flexibility to the model, i.e., they can be used to represent various situations such as economies of scale in production, inventory and safety stock, and transportation, as well as environmental costs associated with greenhouse gas emissions, supplier selection with quantity discount, and technology acquisition (see for instance, Elhedhli and Merrick [40], Saif and Elhedhli [109], Soland [115], Vidyarthi et al. [123]). For the ease of exposition and to alleviate the mathematical notation, we will focus on the particular case of $m = 2$, i.e., the *two-level capacitated facility location problem with concave costs* (2CFLP-Cs), to illustrate the proposed models and solutions algorithms. However, the models and algorithms presented in this work can be readily extended to the more general case of $m > 2$.

The main contributions of this paper are as follows. First, we introduce a general class of MFLPs in which variable operating costs at facilities and transportation costs on arcs are concave functions of the flows. Second, we propose two mathematical programming formulations for the 2CFLP-Cs. The first one is a prevalent mixed-integer nonlinear program which models location decisions and fixed costs with binary variables, and flows with continuous variables. The second formulation is a purely nonlinear program which includes only continuous flow variables on facilities and arcs. At every facility, a fixed charge function models facility's fixed and variable costs such that if the aggregated flow through that facility is zero the total cost is zero, otherwise for flows greater than zero the total cost is the sum of the fixed setup cost and variable operational cost which is concave. Third, we develop exact branch-and-bound algorithms to solve large-scale instances using each of the two formulations. These algorithms contain some algorithmic refinements such as an alternative branching strategy and a variable fixing test at the root node. We perform extensive computational experiments to evaluate the strength of the proposed formulations and the performance of the exact algorithms on problem instances with different cost and capacity scenarios. Results obtained on large-scale instances with up to 2,250 customers and 150 potential facilities confirm the effectiveness of our solution approaches, in particular on the one based on the purely nonlinear program. We also present sensitivity analysis on an instance considering the 3,109 counties in the contiguous USA, to understand the impact of economies of scale in facility operations and transportation on the location and allocation decisions.

The remainder of the paper is organized as follows. Section 2.2 presents a literature review on related DLPs and MFLPs in which nonlinear objective functions are considered. Section 2.3 provides a formal definition of the 2CFLP-C and two mathematical programming formulations. Section 2.4 describes the exact solution algorithms and algorithmic enhancements. The results of extensive computational experiments are reported in Section 2.5. Conclusions follow in Section 2.6.

2.2 Literature Review

We next provide a succinct review of the most relevant MFLPs to our work. For $m = 1$, the problem reduces to a one-level DLP (Efroymson and Ray [38]). For $m = 2$, Kaufman et al. [75] present a two-level uncapacitated location problem with sets of potential warehouse and plant locations at first and second level, respectively. The customer demands are satisfied from plants through warehouses, while minimizing the sum of fixed setup costs of plants and warehouses, and linear variable costs which include production, warehousing, and transportation costs. Pirkul and Jayaraman [102] consider a two-level capacitated DLP with locational decisions at both levels. Aardal et al. [1] present an approximation algorithm for a more general case of uncapacitated MFLPs with $m \geq 2$ levels. Ortiz-Astorquiza et al. [98] present different formulations for MFLPs considering facility location decisions and cardinality constraints at each level. To the best of our knowledge, Ortiz-Astorquiza et al. [100] study the most general class of uncapacitated MFLPs, in which not only cardinality constraints on the number of open facilities at each level are considered but also setup costs for activating the links of the network. In all these MFLPs, the objective functions are assumed to be linear functions of location, allocation, and operational decisions.

DLPs with nonlinear objective functions have also received significant attention. The nonlinear objective functions may represent different scenarios such as congestion cost, diseconomies/economies of scale, general setup costs, among others. Elhedhli [39] and Vidyarthi and Jayaswal [122] study capacitated DLPs in which congestion costs at facilities are modeled with a convex function. They provide an exact solution algorithm by approximating the convex function with a set of linear epigraph constraints (and variables) which are sequentially added in a cutting plane algorithm. For more discussion on DLPs with congestion costs, we refer to Berman and Krass [17]. Harkness and ReVelle [62] assume diseconomies of scale in production cost, where unit production cost increases with increase in production levels. Such production cost functions represent scenarios where capacity of a facility can be increased (e.g., by using extra workers, overtime or additional work shifts) to increase the output, however, with each increase the per unit production cost also increases. The authors model these costs as convex functions approximated with linear piecewise segments. Under a more general setting, Lu et al. [85] concentrate on a problem with nonconvex terms in the objective function. The authors argue that in the long-run, production costs follow economies of scale (i.e., decrease in per unit cost with each additional output) to a certain level of outputs, and after that these costs observe diseconomies of scale. Such features in their work are captured with an inverse S-shaped function, which is first concave to a point and then convex afterwards. Fischetti et al. [46] study a congested multiple allocation DLP in which congestion costs are modeled with non-separable convex quadratic functions of aggregated load at facilities. To obtain a tighter perspective reformulation, quadratic terms are replaced with linear terms at the expense of adding second-order conic constraints. The proposed reformulation is then solved with a generalized Benders decomposition using a branch-and-cut algorithm.

The study of DLPs with concave terms in the objective function started with the work of Feldman et al. [44], in which costs to locate and operate warehouses are included in separable concave functions representing economies of scale, whereas transportation costs are assumed to be linear. They consider a one-level uncapacitated DLP, and develop a heuristic approach to find near optimal solutions. Zangwill [128] models certain special cases of uncapacitated concave DLPs as a single commodity minimum concave cost network flow problem. For these cases, this work proposes an efficient algorithm based on dynamic programming and on the property that an optimal solution of a concave problem can be found in an extreme point of the feasible region. Soland [115] considers a one-level capacitated DLP in which transportation costs are modeled with separable concave functions, and develops a branch-and-bound algorithm to solve the problem. Lower and upper bounds at every node of the tree are obtained by solving a linear transportation problem, in which linear approximations of the concave functions are considered. Kelly and Khumawala [76] focus on mixed-integer programming (MIP) formulation to model one-level DLPs with concave warehousing costs. Their solution approach is based on linear under- and over-estimates of concave functions, and it iteratively solves linear transportation problems to find optimal solutions. Kubo and Kasugai [78] consider the same model as in Soland [115]. They argue that the branch-and-bound algorithm of Soland [115] is mainly effective in problems with a relatively small number of concave functions, otherwise the enumeration algorithm may require a considerable amount of memory and computational time to converge to an optimal solution. The authors present instead an algorithm based on Lagrangian relaxation to provide tighter lower and upper bounds. The algorithm is particularly effective in single-level problems as the Lagrangian subproblems can be further decomposed into smaller independent subproblems, which are easy to tackle. Their upper bounding procedure considers locational decisions obtained from the Lagrangian subproblems to heuristically generate feasible solutions. Aboolian et al. [4] also study a single uncapacitated facility location problem, and develop a Search-and-Cut algorithm in which lower bound is continually improved and upper bound is obtained heuristically.

Cohen and Moon [28] study a multiproduct supply chain problem with a set of potential plant locations. The decisions include assignment of product lines to plants, and shipment volumes from suppliers to plants, and plants to warehouses (acting as demand points). The plants have fixed capacities for each product, and the production cost at each plant-product combination is a piecewise concave function while the transportation costs are linear. Lin et al. [83] address a distribution network problem with uncapacitated facilities and location decisions at two levels. The distribution network consists of plants (fixed locations), potential sets of consolidation and distribution centers, and retailers. They assume linear facility variable costs, concave transportation costs, and solve the concave problem heuristically. Baumgartner et al. [12] focus on an uncapacitated DLP with economies of scale in production at plants, in storage at tanks and in transportation. The design decisions in their model include location of storage tanks and selection of transportation routes. The various concave costs are interpreted as piecewise segments, where the first segment represents a flat rate (fixed charge) up to a certain volume, and afterwards each segment represents per unit cost charged for additional volume. They solve the resulting MIP with a standard branch-and-bound algorithm and develop a heuristic algorithm for providing feasible solutions to large-scale instances. Elhedhli and Merrick [40] consider a sustainable supply chain network design problem with a known set of uncapacitated plants, a potential set of capacitated warehouses, and a set of customers. They assume that transportation activities in the supply chain lead to emissions of green house gases. The amount of emissions, and hence the emission costs are a concave function of the weight of the shipments on the arcs. The problem seeks to optimize traditional supply chain costs and nonlinear emission costs. They use a Lagrangian relaxation to decompose the problem into two subproblems, and the nonlinear concave functions are tackled in one of the subproblem that is further decomposed into linear knapsack problems.

Concave functions find applications in other DLPs as well. Hajiaghayi et al. [60] deal with general setup cost functions. These functions arise in applications where multiple types of facilities are to be co-located, for instance, the problem of placing servers on the Internet. Their objective function comprises of site setup costs (fixed) and facility setup costs, which are concave functions of the number of customers assigned or the number of facilities co-located at each site. In the context of production-inventory-distribution systems, Vidyarthi et al. [123], Daskin et al. [34], Shen et al. [113], and Shen and Daskin [112] use a concave expression to model inventory levels and corresponding inventory costs at facilities. Saif and Elhedhli [109] consider a facility location problem in which costs to acquire the technology are concave.

2.3 Problem Definition and Formulations

The 2CFLP-C is defined as follows. Let $G = (V, E)$ be a directed graph with vertex set $V = V_1 \cup V_2 \cup I$, where V_1 and V_2 are the set of potential facilities at level 1 (intermediate facilities) and level 2 (origin facilities), respectively, and I is the set of customers. We define the directed arc set $E = E_1 \cup E_2 \cup E_t$, where $E_1 = \{\{j, i\} : j \in V_1, i \in I\}$ is the set of arcs connecting the intermediate facilities (level 1) to customers, $E_2 = \{\{k, j\} : k \in V_2, j \in V_1\}$ is the set of arcs representing the flows from the origin facilities (level 2) to the intermediate facilities (level 1), and $E_t = \{\{j, j'\} : j \in V_1, j' \in V_1, j \neq j'\}$ is the set of transshipment arcs interconnecting the intermediate facilities. Let D_i be the customer's demand at node $i \in I$. For $r = 1, 2$, let f_{rj} denote the nonnegative fixed cost associated with locating/opening facility at node $j \in V_r$ and b_{rj} be its capacity. Following the literature (Vidyarthi et al. [123]) of the facility location problem, we further assume that the maximum capacity value at each node is fixed and known in advance.

We define binary location variables z_{1j} and z_{2k} equal to one if and only if facility is located at node $j \in V_1$ and $k \in V_2$, respectively. Also, v_j and u_k are continuous flow variables at node $j \in V_1$ and $k \in V_2$, respectively, and $W_j(v_j)$ and $P_k(u_k)$ are their respective operating cost functions. The functions $W_j(v_j)$ and $P_k(u_k)$ modeling operating costs at the first-level and second-level facilities, respectively, are assumed to be univariate and separable concave functions. Moreover, for each $e \in E$ we define and classify set of continuous flow variables on the arcs as follows: $x_1 = \{x_{1ji} : \{j, i\} \in E_1\}$, $x_2 = \{x_{2kj} : \{k, j\} \in E_2\}$ and $y = \{y_{jj'} : \{j, j'\} \in E_t\}$. Similarly, we define $C_1(x_1)$, $C_2(x_2)$ and $C(y)$ as concave cost functions on arc sets E_1, E_2 and E_t, respectively. The 2CFLP-Cs consists of selecting sets of facilities to open at each level, determining the flows through the facilities at the two levels, and assigning each customer to a set of open facilities, while minimizing the sum of fixed location costs and variable operating and transportation costs. With the above notations, we first present a mixed-integer nonlinear programming (MINLP) formulation for the 2CFLP-C as follows:

$$\textbf{(MINLP)} \quad \text{minimize} \quad \sum_{j \in V_1} \left(f_{1j} z_{1j} + W_j(v_j) + \sum_{\substack{j' \in V_1 \\ : j' \neq j}} C_{jj'}(y_{jj'}) + \sum_{i \in I} C_{1ji}(x_{1ji}) \right) + \sum_{k \in V_2} \left(f_{2k} z_{2k} + P_k(u_k) + \sum_{j \in V_1} C_{2kj}(x_{2kj}) \right) \tag{2.1}$$

$$\text{subject to} \quad \sum_{j \in V_1} x_{2kj} = u_k \leq b_{2k} z_{2k} \;\; \forall\, k \in V_2 \tag{2.2}$$

$$\sum_{k \in V_2} x_{2kj} + \sum_{\substack{j' \in V_1 \\ : j' \neq j}} y_{j'j} = v_j \leq b_{1j} z_{1j} \;\; \forall\, j \in V_1 \tag{2.3}$$

$$\sum_{k \in V_2} x_{2kj} + \sum_{\substack{j' \in V_1 \\ : j' \neq j}} y_{j'j} = \sum_{\substack{j' \in V_1 \\ : j' \neq j}} y_{jj'} + \sum_{i \in I} x_{1ji} \;\; \forall \;\; j \in V_1 \tag{2.4}$$

$$\sum_{j \in V_1} x_{1ji} = D_i \;\; \forall \;\; i \in I \tag{2.5}$$

$$\mathbf{z} \in \{0, 1\}, \mathbf{u} \geq 0, \mathbf{v} \geq 0, \mathbf{x}_1 \geq 0, \mathbf{x}_2 \geq 0, \mathbf{y} \geq 0. \tag{2.6}$$

The objective function consists of two sets of costs. The first set comprises of the fixed costs of opening and variable costs of operating the first-level facilities, transshipment costs between the first-level facilities, and flow costs from the first-level facilities to the customers. The second set consists of the fixed costs of opening and variable costs of operating the second-level facilities, and the flow costs from the second- to first-level facilities. Constraint sets (2.2) and (2.3) are linking and capacity restrictions for facilities at the second and first levels, respectively. Constraints (2.4) are flow balance constraints at the intermediate facilities. Constraints (2.5) ensure that the customers' demands are met. Constraint (2.6) are standard non-negativity and integrality restrictions on the decision variables.

Observe that in the above formulation, fixed costs at both levels are modeled using the binary variables z_{1j} and z_{2k} in the objective function. We next present a purely continuous nonlinear formulation of the 2CFLP-C that does not require the use of binary variables. This nonlinear programming (NLP) formulation is based on the following concave functions:

$$\Theta_j(v_j) = \begin{cases} 0 & \text{if } v_j = 0 \\ f_{1j} + W_j(v_j) & \text{if } v_j > 0, \end{cases} \qquad \Omega_k(u_k) = \begin{cases} 0 & \text{if } u_k = 0 \\ f_{2k} + P_k(u_k) & \text{if } u_k > 0. \end{cases} \tag{2.7}$$

Given that the fixed costs are embedded into the *modified* operational functions $\Theta_j(v_j)$ and $\Omega_k(u_k)$, there is no need to consider binary variables to represent them (as it is often done when modeling DLPs). Combining $\Theta_j(v_j)$ and $\Omega_k(u_k)$ with the continuous variables used in MINLP, the 2CFLP-C can be reformulated as:

$$\textbf{(NLP)} \quad \text{minimize} \quad \sum_{j \in V_1} \left(\Theta_j(v_j) + \sum_{\substack{j' \in V_1 \\ :j' \neq j}} C_{jj'}(y_{jj'}) + \sum_{i \in I} C_{1ji}(x_{1ji}) \right) +$$

$$\sum_{k \in V_2} \left(\Omega_k(u_k) + \sum_{j \in V_1} C_{2kj}(x_{2kj}) \right) \tag{2.8}$$

$$\text{subject to} \quad (2.4) - (2.5)$$

$$\sum_{j \in V_1} x_{1kj} = u_k \leq b_{2k} \;\; \forall\, k \in V_2 \tag{2.9}$$

$$\sum_{k \in V_2} x_{2kj} + \sum_{\substack{j' \in V_1 \\ :j' \neq j}} y_{j'j} = v_j \leq b_{1j} \;\; \forall\, j \in V_1 \tag{2.10}$$

$$\mathbf{u} \geq 0, \mathbf{v} \geq 0, \mathbf{x}_1 \geq 0, \mathbf{x}_2 \geq 0, \mathbf{y} \geq 0 \tag{2.11}$$

The objective function consists of the concave functions $\Theta_j(v_j)$ and $\Omega_j(v_j)$ that capture the fixed opening and variable operating costs of the first-level and second-level facilities, respectively. Constraint sets (2.9) and (2.10) are capacity constraints at the facilities. Constraints

(2.11) are standard non-negativity restrictions on continuous variables.

2.4 Exact Solution Algorithms

We next present two exact algorithms for the 2CFLP-C. They are based on the branch-and-bound (BB) algorithms introduced in Falk and Soland [41] and Soland [115] for continuous separable concave minimization problems. This BB algorithm is an iterative procedure that solves a convex relaxation of the original problem at every node of the enumeration tree. The optimal solution value of the relaxed problem provides a valid lower bound on the original problem. At each node, an upper bound is obtained either heuristically or simply by evaluating the original nonlinear objective function using the solution of the current node. If the desired optimality tolerance has not been achieved, a node is selected from a list of unexplored nodes and two child nodes are created by branching (i.e., partitioning) on the domain of a continuous decision variable.

The tightest convex relaxation of a concave univariate function is the line segment joining the end points of the concave function on the domain of the variable. This is the relaxation that we follow in this work. Other approaches specific to concave minimization problems are broadly based on cutting planes, e.g., Hoffman [64], Tuy et al. [120], and on piecewise inner-approximations of concave functions (Baumgartner et al. [12]). Manousiouthakis et al. [87] present a method to linearize concave power law function resulting in an optimization problem with linear and convex constraints. Ryoo and Sahinidis [106] introduce a branch-and-reduce algorithm which enforces strong domain reduction rules in addition to standard domain reduction techniques commonly used in BB algorithms. This algorithm is available in *BARON*, a state-of-the-art global optimization solver for nonconvex optimization problems. In Section 2.5, we compare BARON with our proposed BB algorithms.

In what follows, we provide a succinct overview of a generic BB algorithm used to solve continuous separable concave minimization problems. For more details on this algorithm, we refer readers to Lawler and Wood [81], Falk and Soland [41], and Soland [115]. For a survey on different solution algorithms for convex and nonconvex MINLPs, we refer to Burer and Letchford [20], Lee and Leyffer [82], and D'Ambrosio and Lodi [32]. We then describe how this algorithm can be used to solve MINLP and NLP. We also present a preprocessing phase to efficiently fix some location decisions before branching on the continuous flow variables. Finally, we describe two approaches to select the considered partitioning point when branching on the flow variables.

2.4.1 A BB Algorithm for Separable Concave Minimization Problems

Consider the following concave minimization problem:

$$
\begin{aligned}
\textbf{(P)} \quad \text{minimize} \quad & \sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} \quad & x \in D \\
& L_i \leq x_i \leq U_i \qquad\qquad \forall i = 1, .., n,
\end{aligned}
$$

where we assume D is a polyhedral set, and each $f_i(x_i)$ is univariate and concave over the set $G_i = \{x_i | L_i \leq x_i \leq U_i\}$, where $L_i \geq 0$ and U_i are finite. We define $G = \bigcup_{i=1}^{n} G_i$. P is a concave minimization problem whose optimal solution lies at some extreme point of the polytope $D \bigcap G$. Further, let $N^0, N^1, ...$ be the nodes of the enumeration tree. We denote N^0 as the root node of the enumeration tree with the concave minimization problem at N^0 equals to **P**. The problem at any node N^a is defined over the feasible region $D \bigcap G^a$, where $G^a \subset G$, i.e.,

$$
G_i^a = \{x_i | L_i \leq L_i^a \leq x_i \leq U_i^a \leq U_i\} \qquad\qquad \forall i = 1, .., n.
$$

We first describe the bounding step of the algorithm. The concave problem at node N^a

is relaxed using ϕ_i^a as the underestimator of $f_i(x_i)$. The function $\phi_i^a(x_i)$ is the equation associated with the line segment joining the points $(L_i^a, f_i(L_i^a))$ and $(U_i^a, f_i(U_i^a))$. Given that $f_i(x_i)$ is concave, for $x_i \in [L_i^a, U_i^a]$, we have $\phi_i^a(x_i) \leq f_i(x_i)$, and for $x \in G^a$, we have $\sum_{i=1}^n \phi_i^a(x_i) \leq \sum_{i=1}^n f_i(x_i)$ (see, Figure 2.2a). Therefore, the lower bound $LB(N^a)$ at node N^a is obtained by solving the following optimization problem:

$$
\begin{aligned}
(\mathbf{P_L^a}) \quad LB(N^a) = \text{minimize} \quad & \sum_{i=1}^{n} \phi_i^a(x_i) \\
\text{subject to} \quad & x \in D \\
& L_i^a \leq x_i \leq U_i^a \qquad\qquad \forall i = 1, .., n.
\end{aligned}
$$

Let $\hat{x}^a$ be the optimal solution of $\mathrm{P_L^a}$. The value of upper bound, $UB^a(N^a)$ at node N^a is equal to $\sum_{i=1}^n f_i(\hat{x}_i^a)$. From now on, we refer to $\phi_i(x_i)$ as the *linear underestimation function* (LUF) of $f_i(x_i)$, and $\mathrm{P_L^a}$ as the *linear underestimation problem* (LUP) at node N^a.

At iteration r, the branching step partitions a selected node N^s into two child nodes N^{2r+1} (left node) and N^{2r+2} (right node), with rectangles $G^{2r+1}, G^{2r+2} \subset G^s$, such that $G^s = G^{2r+1} \bigcup G^{2r+2}$. The branching step further consist of three sub-steps: (1) node selection (best bound), (2) variable selection (worst approximation), and (3) branching point selection to create rectangles. Note that we create two new rectangles by branching on single dimension i.e. the variable selected for branching. Let $\pi = \{N^p, N^q, ...\}$ be the list of available nodes at iteration r^{th}, i.e., the nodes from which no branching has been performed yet. We use a *best bound strategy* to select the node for branching, therefore $N^s = \pi_n$, where

$$
n \in \arg\min \{LB(\pi_t) : t = 1, .., |\pi|\}.
$$

Given that N^s is the node with lowest lower bound value in the set π, the best lower bound (BestLB) of P is equal to $LB(N^s)$. The index of the variable for branching is selected as

$$j \in \arg\max \{f_i(\hat{x}_i^s) - \phi(\hat{x}_i^s) : i = 1, .., n\},$$

i.e., a concave function with the worst underestimation in solution $\hat{x}^s$ of $\mathrm{P}_{\mathrm{L}}^{\mathrm{s}}$ at node N^s. The solution value of the variable x_j lies in the interval $[L_j^s, U_j^s]$. We divide this interval into two intervals: $[L_j^s, S_j]$ and $[S_j, U_j^s]$ and create their associated nodes N^{2r+1} and N^{2r+2}, respectively. The performance of the algorithm is further dependent on how point S_j is selected. Note that the concave subproblems at nodes N^{2r+1} and N^{2r+2} differ from each other and from the subproblem at node N^s only in the interval limits of variable x_j, while the limits for all other variables $x_i \ \forall i = \{1, .., n\} \setminus \{j\}$ are same in these subproblems. The overall BB algorithm is depicted in Algorithm 3 in the Appendix.

2.4.2 Using MINLP as Bounding Procedure

We next describe how Algorithm 1 can be adapted to solve 2CFLP-Cs using MINLP. For each node N^a, we define $p_k^a(u_k)$ and $w_j^a(v_j)$ as LUFs for the univariate concave functions $P_k(u_k)$ and $W_j(v_j)$, respectively. Similarly, $c_{1ji}^a(x_{1ji})$, $c_{2kj}^a(x_{2kj})$ and $c_{jj'}^a(y_{jj'})$ are LUFs for $C_{1ji}^a(x_{1ji})$, $C_{2kj}^a(x_{2kj})$, and $C_{jj'}^a(y_{jj'})$, respectively. Therefore, the LUP for MINLP at node N^a is:

$$(\mathbf{MP_L^a}) \quad \text{minimize} \quad \sum_{j \in V_1} \left(f_{1j} z_{1j} + w_j^a(v_j) + \sum_{\substack{j' \in V_1 \\ : j' \neq j}} c_{jj'}^a(y_{jj'}) + \sum_{i \in I} c_{1ji}^a(x_{1ji}) \right) + \sum_{k \in V_2} \left(f_{2k} z_{2k} + p_k^a(u_k) + \sum_{j \in V_1} c_{2kj}^a(x_{2kj}) \right) \tag{2.12}$$

$$\text{subject to} \quad (2.2) - (2.6)$$

$$L_{2k}^a \leq u_k \leq U_{2k}^a \ \ \forall\, k \in V_2 \tag{2.13}$$

$$L_{1j}^a \leq v_j \leq U_{1j}^a \ \ \forall\, j \in V_1 \tag{2.14}$$

$$L_{2kj}^a \leq x_{2kj} \leq U_{2kj}^a \ \ \forall\, k \in V_2,\ j \in V_1 \tag{2.15}$$

$$L^a_{jj'} \leq y_{jj'} \leq U^a_{jj'} \;\; \forall\, j \in V_1,\; j' \in V_1 : j' \neq j \tag{2.16}$$

$$L^a_{1ji} \leq x_{1ji} \leq U^a_{1ji} \;\; \forall\, j \in V_1,\; i \in I. \tag{2.17}$$

The objective function (2.12) is linear, and constraints (2.13) – (2.17) impose lower and upper limits on continuous variables at node N^a. At root node, lower limits of all variables in $\mathrm{MP}^0_{\mathrm{L}}$ are set to 0, and the upper limits are given in Table 2.1.

Table 2.1: Upper limits at root node

Data Parameter	Value
$U^0_{2k} \;\; \forall k \in V_2$	$\min\{b_{2k}, \sum_{i\in I} D_i\}$
$U^0_{1j} \;\; \forall j \in V_1$	$\min\{b_{1j}, \sum_{i\in I} D_i\}$
$U^0_{2kj} \;\; \forall k \in V_2, j \in V_1$	$\min\{b_{2k}, b_{1j}, \sum_{1\in I} D_i\}$
$U^0_{jj'} \;\; \forall j \in V_1, j' \in V_1$	$\min\{b_{1j}, b_{1j'}, \sum_{1\in I} D_i\}$
$U^0_{1ji} \;\; \forall j \in V_1, i \in I$	$\min\{b_{1j}, D_i\}$

Note that $\mathrm{MP}^{\mathbf{a}}_{\mathbf{L}}$ is an MILP in which integrality conditions on the z variables are kept. Therefore, similar to BB algorithms based on Lagrangean relaxations (Contreras et al. [e.g., 30]), we use an MILP as a bounding procedure at each node of the enumeration tree. The advantage is that $\mathrm{MP}^{\mathbf{a}}_{\mathbf{L}}$ provide stronger bounds as compared to a continuous linear relaxation whereas the disadvantage is clearly the increased computational time to solve $\mathrm{MP}^{\mathbf{a}}_{\mathbf{L}}$. In Section 2.5, we assess the computational benefits of using MILPs as bounding procedures as compared to continuous relaxations.

2.4.3 Using NLP as Bounding Procedure

We now describe how Algorithm 1 can be adapted to solve 2CFLP-Cs using NLP. For each node N^a, we define $\theta^a_j(v_j)\, \forall j \in V_1$ and $\omega^a_k(u_k)\, \forall k \in V_2$ as LUFs for concave operating cost functions $\Theta_j(v_j)$ and $\Omega_k(u_k)$, respectively, whereas, the LUFs for transportation costs are the same as in MINLP. The LUP for NLP at node N^a is:

$$
\begin{aligned}
(\mathbf{NP^a_L}) \quad \text{minimize} \quad & \sum_{j \in V_1} \left(\theta^a_j(v_j) + \sum_{\substack{j' \in V_1: \\ j' \neq j}} c^a_{jj'}(y_{jj'}) + \sum_{i \in I} c^a_{1ji}(x_{1ji}) \right) + \\
& \sum_{k \in V_2} \left(\omega^a_k(u_k) + \sum_{j \in V_1} c^a_{2kj}(x_{2kj}) \right) && (2.18) \\
\text{subject to} \quad & (2.9) - (2.11) \\
& L^r_{2k} \leq u_k \leq U^r_{2k} \;\; \forall\, k \in V_2 && (2.19) \\
& L^r_{1j} \leq v_j \leq U^r_{1j} \;\; \forall\, j \in V_1 && (2.20) \\
& L^a_{2kj} \leq x_{2kj} \leq U^a_{2kj} \;\; \forall\, k \in V_2,\; j \in V_1 && (2.21) \\
& L^a_{jj'} \leq y_{jj'} \leq U^a_{jj'} \;\; \forall\, j \in V_1,\; j' \in V_1 : j' \neq j && (2.22) \\
& L^a_{1ji} \leq x_{1ji} \leq U^a_{1j1} \;\; \forall\, j \in V_1,\; i \in I. && (2.23)
\end{aligned}
$$

Objective function (2.18) is linear, and constraints (2.19)–(2.23) define lower and upper limits (reduced domain) on variables corresponding to univariate concave functions. At root node N^0, all lower limits are set to zero, while upper limits are derived from data as shown in Table 2.1.

Note that $\mathrm{NP^a_L}$ is a (continuous) two-level transportation problem with no binary variables for the location decisions. Therefore, we can state $\mathrm{NP^a_L}$ as a *minimum cost network flow problem* (MCNFLP). This allows us to use highly efficient network solvers which are much faster when compared to a general purpose linear programming (LP) commercial solvers. In what follows, we provide the details of the transformation of $\mathrm{NP^a_L}$ into an MCNFLP.

Let node sets V_2, V_1 and I of graph G be the supply nodes, intermediate nodes and sink nodes, respectively. To transform $\mathrm{NP^a_L}$ into an MCNFLP, we augment G with sets of dummy supply nodes V_2', dummy intermediate nodes V_1', and a dummy sink node i'. Let $d(k)$ and $d'(k')$ be the mapping functions for supply nodes such that $\forall k \in V_2, k' = d(k)$, and $\forall k' \in V_2', k = d'(k')$. Similarly, let $e(j)$ and $e'(j')$ be the mapping functions for intermediate nodes such that $\forall j \in V_1, j' = e(j)$, and $\forall j' \in V_1', k = e'(j')$. Supply and demand are balanced by setting the demand at dummy sink node i' as $\sum_{k \in V_2} b_{2k} - \sum_{i \in I} D_i$. The augmented graph G^N for the network flow representation is depicted in Figure 2.1.

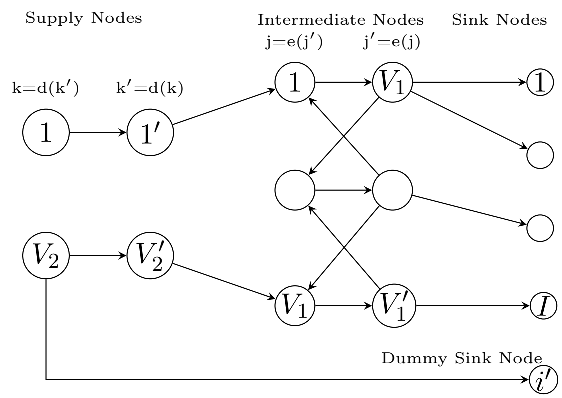


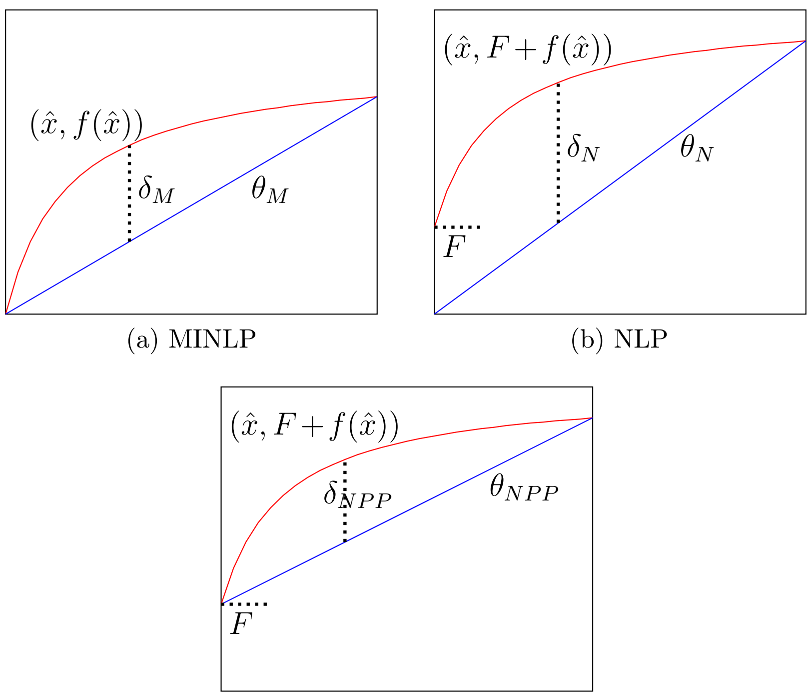
Figure 2.1: Network flow representation of $[LU_{NLP}](Graph : G^N)$

Dummy node i' receives all its inflow directly from supply nodes set V_2 with zero link operating cost. A dummy supply node $k' \in V_2'$ is linked only to its original supply node $k = d'(k')$. Flow and capacity on an arc $\{k, k'\} \forall k \in V_2, k' = d(k)$ represent operational quantity decision u_k and upper limit restriction on u_k at facility $k \in V_2$, respectively. An intermediate node $j \in V_1$ receives inflow from dummy supply nodes set V_2' and dummy intermediate nodes set $V_1' \setminus \{j'\}$, where $j' = e(j)$. Also, dummy intermediate node $j' \in V_1'$ has only one inflow arc from original intermediate node $j = e'(j')$, and the operational quantity decision v_j and upper limit restriction on v_j at facility $j \in V_1$ are represented through flow on arc $\{j, e(j)\}$. Also, a sink node $i \in I$ receives all its inflow from dummy intermediate nodes set V_1'. We define flow variables on arcs as: (i) $u'_{kk'} \, \forall k \in V_2, k' = d(k)$, (ii) $v'_{jj'} \, \forall j \in V_1, j' = e(j)$, (iii) $x'_{2k'j} \, \forall k' \in V_2', j \in V_1$, (iv) $y'_{j'j} \, \forall j' \in V_1', j \in V_1 \setminus \{e'(j')\}$, (v) $x'_{1j'i} \, \forall j' \in V_1', i \in I$, and (vi) $x'_{ki'} \, \forall k \in V_2$. The $\mathrm{NP}_{\mathrm{L}}^{\mathrm{a}}$ can be equivalently stated as the following MCNFLP:

$$
\begin{aligned}
(\mathbf{FP^a_L}) \quad \text{minimize} \quad & \sum_{\substack{j \in V_1 \\ j'=e(j)}} \left(\theta^a_j(v'_{jj'}) + \sum_{\substack{j'' \in V_1 \\ j'' \neq j}} c^a_{j'j''}(y'_{j'j''}) + \sum_{i \in I} c^a_{1j'i}(x'_{1j'i}) \right) + \\
& \sum_{\substack{k \in V_2 \\ k'=d(k)}} \left(\omega^a_k(u'_{kk'}) + \sum_{j \in V_1} c^a_{2k'j}(x'_{2k'j}) \right) && (2.24) \\
\text{subject to} \quad & u'_{kk'} + x'_{ki'} = b_{2k} \ \ \forall\, k \in V_2,\ k' = d(k) && (2.25) \\
& u'_{kk'} = \sum_{j \in V_1} x'_{2k'j} \ \ \forall\, k' \in V'_2,\ k = d'(k') && (2.26) \\
& \sum_{k' \in V'_2} x'_{2k'j} + \sum_{\substack{j'' \in V'_2: \\ j'' \neq j'}} y'_{j''j} = v'_{jj'} \ \ \forall\, j \in V_1,\ j' = e(j) && (2.27) \\
& v'_{jj'} = \sum_{\substack{j'' \in V_1: \\ j'' \neq j}} y'_{j'j''} + \sum_{i \in I} x'_{1j'i} \ \ \forall\, j' \in V'_1,\ j = e'(j') && (2.28) \\
& \sum_{j' \in V'_1} x'_{1j'i} = D_i \ \ \forall\, i \in I && (2.29) \\
& \sum_{k \in V_2} x'_{ki'} = D_{i'} && (2.30) \\
& L^a_{kk'} \leq u'_{kk'} \leq U^a_{kk'} \ \ \forall\, k \in V_2, k' = d(k) && (2.31) \\
& L^a_{jj'} \leq v'_{jj'} \leq U^a_{jj'} \ \ \forall\, j \in V_1, j' = e(j) && (2.32) \\
& L^a_{2k'j} \leq x'_{2k'j} \leq U^a_{2k'j} \ \ \forall\, k' \in V'_2,\ j \in V_1 && (2.33) \\
& L^a_{j'j} \leq y'_{j'j} \leq U^a_{j'j} \ \ \forall\, j' \in V'_1,\ j \in V_1 : j \neq e'(j') && (2.34) \\
& L^a_{1j'i} \leq x'_{1j'i} \leq U^a_{1j'i} \ \ \forall\, j' \in V'_1,\ i \in I && (2.35) \\
& \boldsymbol{u', v', x'_2, x'_1, x', y'} \geq 0. && (2.36)
\end{aligned}
$$

The costs terms in objective function (2.24) are in the same order as in (2.18). Constraints (2.25)–(2.30) are flow conservation at node sets V_2, V'_2, V_1, V'_1, I and i', respectively. Limits on variables associated with concave functions are imposed in (2.31)–(2.35). Finally, (2.36) impose non-negativity restriction on flow variables.

The main advantage of MINLP over NLP, when used in Algorithm 1, is the strength of its associated lower bounds at each node of the enumeration tree. From Figures 2.2a and 2.2b, we note that at any point $\hat{x}$, the error deviation $\delta_N = F + f(\hat{x}) - \theta_N \hat{x}$ in NLP is greater than the error deviation $\delta_M = f(\hat{x}) - \theta_M \hat{x}$ in MINLP. This difference arises given that the fixed cost of opening a facility is included in the LUF of NLP but not in the LUF of MINLP. The main disadvantage of MINLP over NLP is that to get these stronger bounds we need to solve an MILP at each node, which is significantly more time consuming as compared to solving the associated MCNFLPs of NLP with the *network simplex algorithm* of CPLEX. In Section 2.5, we perform an extensive comparison of both approaches to determine under which configuration of input parameters one bounding procedure may dominate the other.



(a) MINLP (b) NLP

(c) NLP after preprocessing

Figure 2.2: Underestimation of formulation at node N^a

2.4.4 Preprocessing Phase

Observe that $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ at a node N^a corresponds to a discrete two-level capacitated facility location problem which is challenging to solve for large-size instances. One way to reduce the computational time of $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ is by fixing as many binary location variables as possible before enumeration begins. Fixing facilities also reduces the computation time of $\mathrm{FP}_{\mathrm{L}}^{\mathrm{a}}$. In this case, when a facility is fixed to zero, we can eliminate all the inflow/outflow from that facility, whereas if a facility is fixed to one, the cost functions in equation (2.7) at that facility reduces to $\Theta_j(v_j) = f_{1j} + W_j(v_j)$ and $\Omega_k(u_k) = f_{2k} + P_k(u_k)$. That is, the constant terms in the LUFs ($\theta_j^a(v_j)$ and $\omega_k^a(u_k)$) always include the fixed costs of facilities, and the slope is calculated on the concave operating cost only. On a fixed interval, the slope value at such a facility is lesser than that when a facility is not fixed (see for instance, Figures 2.2b and 2.2c). Fixing facility location decisions to one tend to improve the linear underestimation at each node, thus providing stronger lower bounds.

The preprocessing phase begins with solving $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ with no variables fixed, to give a valid upper bound $UB(N^0)$ and location decision vector $\hat{z}$. Next, fix a facility decision variable z $= 1$ (or 0) if $\hat{z} = 0$ (or 1) and reoptimize $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$. If the lower bound $LB(N^0)$ obtained after reoptimizing is greater than $UB(N^0)$, then we permanently fix $z = 0$ (or 1) in subsequent problems of the preprocessing phase. Let $z_{1j}^F\, \forall j \in V_1$ and $z_{2k}^F\, \forall k \in V_2$ be the decision vectors for fixing facilities at first and second level, respectively. The preprocessing phase is depicted in Algorithm 1.

2.4.5 Partitioning Point Selection

At r^{th} iteration of Algorithm 1, the node selected for branching is partitioned into two child nodes that differ from their parent node and from each other in terms of the lower and upper limits to be set on the variable selected for branching. We have empirically tested two strategies to select the branching point S_j. The first strategy is the one described in Section 2.4.1, in which $S_j = \hat{x}_j$, where j is the index of variable x that gives maximum deviation, and

Algorithm 1: Preprocessing Phase

Solve $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ to give $UB(N^0), \hat{z}_{2k}\,\forall\, k \in V_2$ *and* $\hat{z}_{1j} \forall\, j \in V_1$.
for $k = 1$ to $|V_2|$ **do**
    Temporarily set $z_{2k} = 1\,(0)$ if $\hat{z}_{2k} = 0\,(1)$.
    Reoptimize $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ by considering $z_{2k'}^F$ $\forall k' = \{1, ..., k-1\}$ that are already fixed.
    if $(LB(N^0) > UB(N^0))$ **then** update $z_{2k}^F = 0\,(1)$
end for
for $j = 1$ to $|V_1|$ **do**
    Temporarily set $z_{1j} = 1\,(0)$ if $\hat{z}_{1j} = 0\,(1)$.
    Reoptimize $\mathrm{MP}_{\mathrm{L}}^{\mathrm{a}}$ by considering z_{2k}^F and $z_{1j'}^F$ $\forall j' = \{1, ..., j-1\}$ that are already fixed.
    if$(LB(N^0) > UB(N^0))$ **then** update $z_{1j}^F = 0\,(1)$
end for

$\hat{x}_j$ is solution value of variable x_j in the LUP at parent node. We denote it as the *solution point* (SP) strategy. The advantage of this branching strategy is that when the solution of the two child problems are close to that of the parent problem, it has a positive impact in improving the lower bound quickly. We consider a second branching strategy in which we select a point such that the sum of the area between a concave function and its respective linear underestimation of the resulting child problems is minimum (Liu et al. [84]). This is denoted as the *minimum area point* (MAP) strategy (see, Figure 2.3).

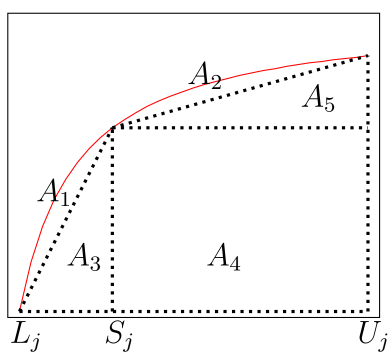


Figure 2.3: Minimum area point strategy

This point can be obtained by solving following continuous optimization problem:

$$\min_{S_j} \{A_1 + A_2 : L_j \leq S_j \leq U_j\} \equiv \max_{S_j} \{A_3 + A_4 + A_5 : L_j \leq S_j \leq U_j\}.$$

In Section 2.5, we perform an extensive comparison of both partitioning strategies to determine under which configuration of input parameters one strategy may dominate the other.

2.5 Computational Experiments

We present the results of our extensive computational experiments comparing the traditional MINLP formulation and the purely continuous NLP formulation for the 2CFLP-C. We analyze the performance of the two branching strategies, preprocessing, and the BB algorithm for the two formulations over problem instances under different cost structures and capacity scenarios. All algorithms were coded in C++ and executed on an Intel Xeon E5-2687W v3 processor at 3.10 GHz in a Linux environment. The algorithms were implemented using CPLEX 12.6 Concert Technology with its default settings using one thread. In what follows, we present the summary of the results of our experiments, whereas the detailed tables of the results appear in the Appendix. We also present a sensitivity analysis using real location data from the 3,109 counties of the contiguous United States to analyze the solution output of our model under different concave costs.

2.5.1 Benchmark Instances

The first set of test problems are generated using the scheme proposed in Vidyarthi et al. [123]. The coordinates for facilities and customers are generated uniformly as U[10,100]. Customer demand D_i is randomly generated with U[50,300]. The capacities of the potential facilities at the first level (b_{1j}) and second level (b_{2k}) are first generated uniformly on U[10,160] and then scaled such that $\frac{\sum_{j\in V_1} b_{1j}}{\sum_{i\in I} D_i} = \frac{\sum_{k\in V_2} b_{2k}}{\sum_{i\in I} D_i} = \kappa$, where, $\kappa \geq 0$ is a scaling parameter varied to represent different capacity scenarios (e.g. tight capacity and excess capacity). Fixed costs of the first and second level facilities are set to $f_{1j} = \alpha_1 \times (U[0,250] + U[10,100] \times (b_{1j})^{0.5})$ and $f_{2k} = \alpha_2 \times (U[0,250] + U[10,100] \times (b_{2k})^{0.5})$, respectively, where

$\alpha_1 \geq 0$ and $\alpha_2 \geq 0$ are scaling parameters. We model various concave costs with power law function $f(x) = Ax^{\lambda}$, where A is a non-negative constant, and exponent $\lambda \in (0, 1)$ ensures that $f(x)$ is concave over it's domain. We set $\lambda = 0.5$ in our computational experiments unless otherwise stated. Thus, transportation costs on the links are captured using the following concave functions: $C_{1ji}(x_{1ji}) = \beta \times (0.2 \times d_{1ji})x_{1ji}^{\lambda}$, $C_{2kj}(x_{2kj}) = \beta \times (0.2 \times d_{2kj})x_{2kj}^{\lambda}$ and $C_{jj'}(y_{jj'}) = \beta \times (0.2 \times d_{jj'})y_{jj'}^{\lambda}$, where, d_{1ji}, d_{2kj}, and $d_{jj'}$ are euclidean distances between nodes: $i \in I$, $j, j' \in V_1$, $k \in V_2$, and $\beta \geq 0$ is a scaling parameter. Variable costs at the first and second level facilities are $W_j(v_j) = \gamma_1 \times U[10, 50] \times v_j^{\lambda}$ and $P_k(u_k) = \gamma_2 \times U[10, 50] \times u_k^{\lambda}$, respectively, where $\gamma_1 \geq 0$ and $\gamma_2 \geq 0$ are scaling parameters. The scaling parameters ($\beta, \alpha_1, \alpha_2, \gamma_1, \gamma_2$, and κ) are varied to generate instances with balanced cost, dominant fixed cost, dominant variable cost, tight capacity, and excess capacity scenarios. Balanced cost scenario refers to instances in which the facility fixed cost (FFC), facility variable cost (FVC), and transportation cost (TrC) are fairly evenly distributed. In dominant fixed cost scenario, the FFC accounts for more than 50% of the total cost (FFC+FVC+TrC). In dominant variable cost scenario, the (FVC+TrC) accounts for more than 50% of the total cost. In tight capacity scenario, the sum of the capacities of the potential facilities at each level is three times the total demand. In excess capacity scenario, the sum of the capacities of the potential facilities at each level is seven times the total demand. We vary the scaling parameters to generate instances under different scenarios as follows:

- Balanced Cost (BC): $\beta = 1$, $\alpha_1 = 1$, $\alpha_2 = 1$, $\gamma_1 = 1$, $\gamma_2 = 1$, and $\kappa = 5$
- Tight Capacity (TC): $\beta = 1$, $\alpha_1 = 1$, $\alpha_2 = 1$, $\gamma_1 = 1$, $\gamma_2 = 1$, and $\kappa = 3$
- Excess Capacity (EC): $\beta = 1$, $\alpha_1 = 1$, $\alpha_2 = 1$, $\gamma_1 = 1$, $\gamma_2 = 1$, and $\kappa = 7$
- Dominant Fixed Cost (DFC): $\beta = 1$, $\alpha_1 \sim U[4{,}6]$, $\alpha_2 \sim U[4{,}6]$, $\gamma_1 = 1$, $\gamma_2 = 1$, and $\kappa = 5$
- Dominant Variable Cost (DVC): $\beta \sim U[4{,}6]$, $\alpha_1 = 1$, $\alpha_2 = 1$, $\gamma_1 \sim U[4{,}6]$, $\gamma_2 \sim U[4{,}6]$ and $\kappa = 5$

Table 2.2 provides the details of the test instances such as the number of potential facilities at the first and second levels as well as the number of customers for every problem instance. These instances are categorised into five sets. For example, Set I comprises of fours instances in which the number of potential facilities at first level and second level are 40 and 10, respectively, whereas the number of customers vary from 200 in instance 1, 300 in instance 2, 400 in instance 3 to 500 in instance 4. The largest problem instance #20 (Set V) comprises of 100 potential facilities at the first level, 50 potential facilities at the second level and 2,250 customers. Thus, our set of test problems comprises of a total of 100 instances as a result of the combination of five scenarios (BC, TC, EC, DFC, and DVC) and 20 instances (Set I to V).

Table 2.2: Details of Test Instances

		No. of Potential Facilities		
Set	Instance	Level 1	Level 2	No. of Customers
I	1, 2, 3, 4	40	10	200, 300, 400, 500
II	5, 6, 7, 8	55	20	600, 700, 800, 900
III	9, 10, 11, 12	70	30	1000, 1100, 1200, 1300
IV	13, 14, 15, 16	85	40	1400, 1500, 1625, 1750
V	17, 18, 19, 20	100	50	1875, 2000, 2125, 2250

2.5.2 Computational Performance

In our preliminary computational experiments, we used BARON, a state-of-the-art solver for nonconvex optimization problems to solve the MINLP formulation to global optimality. The obtained results indicated that BARON takes excessive time to solve even a small size problem to optimality. Furthermore, even for instances in Set I, BARON yields huge optimality gaps within a time limit of one day of computation time.

In the remainder of this section, we report the results of the computational experiments with the proposed exact solution algorithm. In the first set of experiments, we compare the performance of two branching strategies. We then analyze the impact of preprocessing on the MINLP and NLP formulations. Finally, we report the performance of the BB algorithm on the MINLP and NLP formulations under a variety of cost structures.

2.5.2.1 Performance of Branching Strategies

We compare the performance of the two branching strategies using both MINLP and NLP formulations. For this set of experiments, we select six medium-size instances (#9 to #14), and solve them using the proposed BB algorithm under various scenarios. The termination criteria is set to an optimality gap of 0.1% and a time limit of 86,400 secs (24 hours). A summary of the results is presented in Table 2.3, whereas the detailed results are reported in Tables A.1 and A.2 in the Appendix. The columns $\mathrm{SP_g}$ and $\mathrm{MAP_g}$ report the average optimality gap (%) obtained from the SP and MAP branching strategies, respectively, whereas columns $\mathrm{SP_t}$ and $\mathrm{MAP_t}$ report average CPU time in seconds. The reduction in the computational time is reported in the column %Reduction, which is computed as follows: $\%\text{Reduction} = \frac{(\mathrm{SP_t} - \mathrm{MAP_t}) \times 100}{\mathrm{SP_t}}$. For instances where the BB algorithm did not converge to an optimality gap of 0.1% within prescribed time limit, we write 'time' in the corresponding entry of the table. Next two columns lists number of instances out of six that are solved to the desired optimiality gap within time limit using each strategy. Finally, in the last column SP/MAP, we report the number of instance on which SP outperformed MAP and MAP outperformed SP. We use the notation n/m to indicate that out of $(n + m)$ instances, SP outperformed MAP on n instances and MAP outperformed SP on m instances.

From Table 2.3, we observe that the SP strategy performs slightly better than the MAP strategy on four (out of six) instances using the MINLP formulation as SP reports an average optimality gap of 0.15% (as compared to 0.16% by MAP) at termination. Moreover, using the NLP formulation, the SP strategy outperforms the MAP strategy in all six instances with an average optimality gap of 0.11% (compared to 0.14%) and a 36% reduction in average CPU time.

Table 2.3: Summary of Performance of Branching Strategies on MINLP and NLP Formulations

		Average Gap (%)		Average Time (s)			# Inst Opt		
Formulation	Scenario	SP_g	MAP_g	SP_t	MAP_t	%Reduction	SP	MAP	SP/MAP
MINLP	BC	0.15	0.16	time	time	-	0	0	4/2
	DFC	0.11	0.10	34,409	24,271	29	4	5	1/5
	DVC	0.20	0.19	79,476	time	-9	1	0	3/3
	EC	0.15	0.14	50,115	52,208	-4	3	3	2/4
	TC	0.40	0.39	time	time	-	0	0	2/4
NLP	BC	0.11	0.14	63,697	time	-36	2	0	6/0
	DFC	0.10	0.10	12,056	1,699	86	6	6	0/6
	DVC	0.13	0.14	61,433	time	-41	3	0	5/1
	EC	0.10	0.11	33,257	42,846	-29	5	4	5/1
	TC	0.30	0.29	time	time	-	0	0	4/2

Under the DFC scenario, using the MINLP formulation, we were able to solve four (out of six) instances with the SP strategy and five (out of six) instances with the MAP strategy to optimality. However, using the NLP formulation and the two branching strategies, the BB algorithm was able to solve all six instances to optimality with an average gap of 0.10%. The average optimality gap of the MINLP formulation under the SP and MAP strategies are 0.11% and 0.10% respectively. However, note that the average CPU times of the two strategies are significantly different under both formulations. The MAP strategy is computationally efficient compared to SP as it reduces the CPU time by 86% using the NLP formulation and 29% with the MINLP formulation.

Under the DVC scenario, we observe that using the MINLP formulation and the SP strategy, we were able to solve one (out of six) instance to optimality compared to none with the MAP strategy. Using the NLP formulation, we were able to solve two (out of six) instances to optimality with the SP strategy and none with the MAP strategy. The average gaps of the MINLP formulation under the SP and MAP strategies are 0.20% and 0.19%, respectively. The average gaps of the NLP formulation under the SP and MAP strategies are 0.13% and 0.14%, respectively. Hence, under the DVC scenario, using the NLP formulation, SP outperforms MAP on five (out of six) instances with a 41% reduction in average CPU time.

Under the EC scenario, using the NLP formulation, five (out of six) instances were solved to optimality with the SP strategy and four (out of six) instances with the MAP strategy. Furthermore, SP outperforms MAP on five (out of six) instances especially in the case of the NLP formulation with a 29% reduction in average CPU time. Finally, under the TC scenario, there is no clear distinction between the performance of the two strategies as none of the instances were solved to optimality within the time limit and the average optimality gaps at terminations are also similar.

From these results, we observe that the right branching strategy for an instance is dependent on the cost and capacity structures of the instance (i.e. the type of scenario). In the rest of the experiments, we use a mix of both strategies as follows. We first categorize every instance into a scenario. To do so, we use the upper bound information obtained at the root node. More specifically, we determine the percentage of the facility fixed costs in the upper bound value, and if that percentage is greater than 50% , the instance is then categorized as DFC, and hence we use MAP branching strategy otherwise we use the SP branching strategy.

2.5.2.2 Performance of Preprocessing

In the next set of experiments, we analyze the performance of preprocessing on the strength of MINLP and NLP formulations. For this, we solve the medium-size instances #9 to #14 using the MINLP and NLP formulations under every cost and capacity scenarios. We use the same termination criteria as in the previous experiments. The results are summarized in Table 2.4 and the detailed results are provided in Tables A.3 and A.4 in the Appendix. Columns $\mathrm{WoP_g}$ and $\mathrm{WoP_t}$ report the average optimality gap and CPU time without preprocessing, respectively. Similarly, columns $\mathrm{WP_g}$ and $\mathrm{WP_t}$ report the average optimality gap and CPU time with preprocessing, respectively. The column "%Reduction" reports the percentage reduction obtained in CPU time as a result of preprocessing and is computed as: %Reduction $\left(=\frac{(\mathrm{WoP_t}-\mathrm{WP_t})\times 100}{\mathrm{WoP_t}}\right)$. In the last column WoP/WP, we use n/m notation to indicate that out of $(n+m)$ instances, n instances performed better without preprocessing and m with preprocessing.

From Table 2.4, we observe that under the BC scenario, preprocessing reduces the average CPU time significantly using both the formulations. The reduction in average CPU time is 17% using the MINLP and 21% using the NLP formulations. Also, with preprocessing, the average optimality gap is slightly lower with the MINLP formulation, i.e, 0.14% compared to 0.15% without preprocessing. Under the DFC scenario, we observe that the average optimality gaps with preprocessing are similar to those without preprocessing using both the formulations. However, it is interesting to note that preprocessing reduces the average CPU time significantly using the MINLP formulation (64%), but there is an increase of 43% in average CPU time using the NLP formulation. Under the DVC scenarios, the preprocessing increases the computational time for both the formulations. Under the EC scenario, using the MINLP formulation, the preprocessing reduces the average optimality gap from 0.15% to 0.12% and the average CPU time by 6%. With the NLP formulation, the preprocessing reduces the average CPU time by 21%, however there is no impact on optimality gaps. We note that none of the six instances under the TC scenario were solved to optimality gap of 0.1% under the time limit using any of the formulations. However, with preprocessing, we do observe slight reduction in the optimality gap at termination in both the formulations. From these results, we observe that preprocessing is recommended as it reduces the optimality gap and/or the CPU time under BC, DFC, EC, and TC scenarios.

Additionally, in Table 2.5, we report the percentage of facilities fixed out of $(|V_1| + |V_2|)$ facilities at the root node. Note that our results indicate that preprocessing fixes an average of 93% of the facilities under the DFC scenario. Also, as a result of preprocessing, we were able to fix 66%, 64% and 48% of facilities under EC, BC and TC scenarios, respectively. However, we were able to fix only 6% of the facilities under the DVC scenario. Furthermore, in this scenario preprocessing is computationally inefficient in both the MINLP and NLP formulations. Hence, we recommend preprocessing in all these scenarios except DVC. To decide whether to perform preprocessing in our implementation, we first solve the root node problem. If the percentage of fixed cost in the upper bound value from the root node solution is less than 20% then we categorize the instance as DVC, and do not consider preprocessing for that instance.

Table 2.4: Summary of Performance of Preprocessing on MINLP and NLP Formulations

		Average Gap (%)		Average Time (s)			
Formulation	Scenario	WoP_g	WP_g	WoP_t	WP_t	%Reduction	WoP/WP
MINLP	BC	0.15	0.14	time	71,834	17	0/6
	DFC	0.10	0.10	24,271	8,847	64	1/5
	DVC	0.20	0.20	79,476	81,610	-3	3/3
	EC	0.15	0.12	50,115	47,203	6	0/6
	TC	0.40	0.38	time	time	-	0/6
NLP	BC	0.11	0.11	63,697	50,043	21	1/5
	DFC	0.10	0.10	1,699	2,428	-43	4/2
	DVC	0.13	0.14	61,433	71,883	-17	6/0
	EC	0.10	0.09	33,257	26,341	21	3/3
	TC	0.30	0.29	time	time	-	2/4

Table 2.5: Effect of Preprocessing on the Percentage of Facilities Fixed at the Root Node

	Percentage of Facilities Fixed under Scenario				
Instance	BC (%)	DFC (%)	DVC (%)	EC (%)	TC (%)
9	59	98	4	70	50
10	62	98	18	59	62
11	83	95	2	59	53
12	68	90	0	86	49
13	39	90	11	54	26
14	70	85	2	66	46
Average	64	93	6	66	48

2.5.2.3 Performance of BB Algorithm

In this set of computational experiment, we compare the performance of the BB algorithm using the MINLP and NLP formulations along with the efficient branching strategy and (with/without) preprocessing for various scenarios as identified from our previous computational results.

First, we establish a limit on the stopping criteria in terms of desired optimality gap to make the final comparison between the performance of the two formulations. We analyze the CPU time required to obtain solutions with different optimality gaps: 1%, 0.5%, 0.2% and 0.1%. For this, we choose MINLP formulation because from the previous results (to establish the criterion for branching strategy and preprocessing) we realized that in those subset of instances MINLP formulation is comparatively slower. The results for the medium-size instances #9 to #14 under various scenarios and efficient branching and preprocessing strategies are shown in Table 2.6.

From these results, it is evident that these instances converge to an optimality gap of 1% in less than an hour. However, as we attempt to close this optimality gap further, the CPU time increases many folds. For example, in the BC scenario, the average CPU time to reach 0.2% and 0.1% optimality gaps are 12 and 30 times higher, respectively, compared to the time to reach 1% optimality gap. Under the DVC scenario, the average CPU time to reach 0.2% and 0.1% optimality gaps are 23 and 37 times higher, respectively, compared to the time to reach 1% optimality gap. We observe similar multifold increase in CPU times in the other scenarios as well. Also, we note that, under TC scenario, no optimal solutions were found within a gap of 0.2% and 0.1% in time limit of 86,400 secs. Thus, in the remainder of the experiments, we set the termination criteria to an optimality gap of 0.2% and a time limit of 86,400 secs.

Table 2.6: CPU Times to Reach Different Optimality Gaps using MINLP Formulation

	Branching	With or Without	Average Time (s)			
Scenario	Strategy	Preprocessing	Gap = 1 %	Gap = 0.5%	Gap = 0.2 %	Gap = 0.1 %
BC	SP	WP	2,348	2,913	29,132	71,816
DFC	MAP	WP	1,352	1,353	1,585	8,847
DVC	SP	WoP	2,211	7,823	52,072	81,598
EC	SP	WP	2,837	4,721	19,816	47,199
TC	SP	WP	1,496	10,900	time	time
Average			2,049	5,542	37,801	59,712

The performance of the BB algorithms is summarized in Table 2.7, whereas the detailed results are provided in Tables A.5-A.9 in the Appendix. In Table 2.7, the first and the second columns give the scenarios and sets of instances respectively. The next four columns report the quality of the root node solution for both the formulations. Column MINLP_{rg} and NLP_{rg} are root node gaps, and MINLP_{rt} and NLP_{rt} are CPU times needed to obtain root node solution for the two formulations, respectively. We also report the performance of the BB algorithm on the two formulations based on the final optimality gap obtained at termination or the time taken to converge to optimality gap of 0.2%. The next four columns report the average optimality gaps (MINLP_g, NLP_g) and CPU times (MINLP_t, NLP_t) at termination. The column "%Red" refers to the percentage reduction in CPU time of the NLP formulation compared to the MINLP formulation. In the last column "MINLP/NLP", we use n/m notation to report the number of instances on which MINLP outperforms NLP (n) and NLP outperforms MINLP (m) out of ($n + m$) instances.

Results in Table 2.7 indicate that under all the scenarios, the average optimality gap at the root node for the MINLP formulation is lower than that of the NLP formulation. Note that the average optimally gap at the root node for the MINLP formulation is 0.86% whereas that of the NLP formulation is 2.40 %. However, the average CPU time to solve the root node of the problem is lower for the NLP formulation. While it takes on average of 21 sec to solve the problem using the MINLP formulation, it takes less than 1 sec using the NLP formulation.

The performance of the BB algorithm at termination indicates that, on average, the NLP formulation takes 30% less CPU time than the MINLP counterpart. The significant reduction in CPU times using NLP formulation are obtained in EC (71%) and DFC scenarios (59%) followed by BC scenario (40%), whereas in case of TC (20%) and DVC scenarios (16%) the reductions are comparatively moderate. We also observe that NLP formulation is particularly faster on large-size instances. For example, in the instances belonging to Set V, NLP reduces CPU times by 78%, 62% and 25% under the EC, DFC, and BC scenarios, respectively. Similar savings in CPU time are also observed in Set IV instances, where the reductions are 68%, 57%, 50% and 29% under the EC, BC, DFC and DVC scenarios, respectively.

The results in this table indicates that, out of 20 instances, the NLP outperforms MINLP formulation on 19 instances each under the BC and TC scenarios, and on 16 instances under the DFC scenario. Furthermore, the NLP formulation outperforms MINLP formulation on

Table 2.7: Summary Results of Performance of BB Algorithm using MINLP and NLP Formulations

		Root Node Performance				Performance at Termination					
		Avg. Gap (%)		Avg. Time(s)		Avg. Gap (%)		Avg. Time(s)			
Scenario	Set	MINLP_{rg}	NLP_{rg}	MINLP_{rt}	NLP_{rt}	MINLP_{g}	NLP_{g}	MINLP_{t}	NLP_{t}	%Red	MINLP/NLP
BC	I	1.21	2.93	< 1	< 1	0.18	0.20	90	78	13	1/3
	II	0.94	2.32	1	< 1	0.20	0.19	689	467	32	0/4
	III	0.90	2.53	6	< 1	0.20	0.20	7,564	3,774	50	0/4
	IV	0.75	2.11	45	< 1	0.23	0.20	76,646	32,960	57	0/4
	V	0.65	1.7	82	< 1	0.41	0.28	time	64,853	25	0/4
	Avg.	**0.89**	**2.32**	**27**	**<1**	**0.24**	**0.21**	**34,289**	**20,426**	**40**	**1/19**
DFC	I	0.55	2.83	< 1	< 1	0.19	0.19	22	23	-5	3/1
	II	0.46	1.03	< 1	< 1	0.19	0.19	236	226	4	0/4
	III	0.41	1.11	< 1	< 1	0.20	0.20	668	647	3	1/3
	IV	0.35	2.27	6	< 1	0.20	0.20	5,797	2,885	50	0/4
	V	0.29	1.4	28	< 1	0.20	0.20	38,411	14,625	62	0/4
	Avg.	**0.41**	**1.73**	**7**	**<1**	**0.20**	**0.20**	**9,027**	**3,681**	**59**	**4/16**
DVC	I	1.77	4.9	1	< 1	0.20	0.20	175	87	50	0/4
	II	1.53	4.52	7	< 1	0.20	0.19	2,104	2,241	-7	2/2
	III	1.31	3.23	10	< 1	0.19	0.19	34,887	26,218	25	1/3
	IV	0.84	2.49	59	< 1	0.31	0.22	time	61,100	29	0/4
	V	1.14	2.6	75	< 1	0.58	1.04	time	time	-	3/1
	Avg.	**1.32**	**3.55**	**30**	**<1**	**0.29**	**0.36**	**42,003**	**35,210**	**16**	**6/14**
EC	I	1.24	2.8	0	< 1	0.18	0.19	34	34	0	3/1
	II	1.31	4.37	1	< 1	0.19	0.19	386	396	-3	3/1
	III	1.08	3.08	9	< 1	0.20	0.18	4,339	3,577	18	1/3
	IV	0.68	2.09	21	< 1	0.20	0.20	24,354	7,813	68	1/3
	V	0.67	2.05	29	1	0.21	0.20	45,841	10,210	78	0/4
	Avg.	**0.99**	**2.88**	**12**	**<1**	**0.20**	**0.19**	**14,991**	**4,406**	**71**	**8/12**
TC	I	1.13	2.37	< 1	< 1	0.19	0.20	2,708	638	76	1/3
	II	0.52	1.73	4	< 1	0.20	0.20	71,761	24,236	66	0/4
	III	0.59	1.44	14	< 1	0.37	0.29	time	68,889	20	0/4
	IV	0.54	1.07	33	< 1	0.40	0.33	time	time	-	0/4
	V	0.61	0.96	88	1	0.52	0.42	time	time	-	0/4
	Avg.	**0.68**	**1.51**	**28**	**<1**	**0.34**	**0.29**	**66,749**	**53,314**	**20**	**1/19**
	Avg.	**0.86**	**2.40**	**21**	**<1**	**0.25**	**0.25**	**33,406**	**23,407**	**30**	**20/80**

14 and 12 instances (out of 20) under DVC and EC scenarios, respectively. In summary, the NLP outperforms MINLP formulation on 80 out of 100 instances comprised of varying cost and capacity structures. Thus, for large-size instances, using the NLP formulation in the BB algorithm seems to be a better choice compared to its MINLP counterpart.

2.5.3 Sensitivity Analysis

In this section, we perform a sensitivity analysis of the proposed model to changes in the input parameters, especially the economies of scale in operations at the facilities and transportation. We also provide insights by comparing the solutions (locations and allocation decisions) across various scenarios. For this analysis, we consider a large-size instance using real location data available on the demographic information of the 3,109 counties in the contiguous United States. For every county, we obtain its population, latitude/longitude information from the United States Census Bureau (*https://www2.census.gov/geo/docs/reference/cenpop2010/county*), and its average housing price from *https://www.census.gov/support/USACdataDownloads.html#HSG*. Each county represents an aggregate set of customers (demand points) and hence there are 3,109 demand points ($|I| = 3109$). We consider a two-level production-distribution system, where the first level represents distribution centers (DCs) and the second level represents plants. Thirty most populated counties are used as potential sites for DCs ($|V_1|$=30), and ten most populated counties are used as potential sites for plants ($|V_2|$=10). The demand at every customer node (D_i) is obtained by dividing the county's population by 1,000. The fixed cost of opening plants are set proportional to the average residential prices as: $f_{2k} = 0.5 \times$ (County's Average Residential Price), similarly the fixed cost of opening DCs are set proportional to the average residential prices as: $f_{1j} = 0.2 \times$ (County's Average Residential Price). Plant's capacity b_{2k} and DC's capacity b_{1j} are randomly generated as U[1.0,1.5] $\times f_{2k}$ and U[1.0,1.5] $\times f_{1j}$, respectively. These capacities are then scaled such that total plant capacities is 10 times the total demand, and also total DC capacities is 10 times the total demand. The transportation costs between two nodes are set as $C_{1ji}(x_{1ji}) = [0.2 \times d_{1ji}]x^{\lambda}_{1ji}$, $C_{2kj}(x_{2kj}) = [0.2 \times d_{2kj}]x^{\lambda}_{2kj}$ and $C_{jj'}(y_{jj'}) = [0.2 \times d_{jj'}]y^{\lambda}_{jj'}$, where, d_{1ji}, d_{2kj}, and $d_{jj'}$ are obtained by dividing the spherical distance between nodes by 100. Finally, we set production cost at facilities to $P_k(u_K) = U[5, 36] \times u^{\lambda}_k$, and handling cost at the DC to $W_j(v_j) = U[5, 36] \times v^{\lambda}_j$. The economies of scale in operations at facilities and/or in transportation are captured using concave cost functions that are modeled as power law functions $f(x) = A(x)^{\lambda}$, where $0 \leq \lambda \leq 1$ as described in Section 2.5.1.

We have conducted sensitivity analysis by varying λ under three scenarios. In the first scenario (referred as *EOS-F*), we assume that production cost $P_k(u_k)$ and handling cost $W_j(v_j)$ are concave function of the flow, while transportation costs are linear. In second scenario (referred as *EOS-T*), we assume concave transportation costs: $C_{2kj}(x_{2kj})$, $C_{jj'}(y_{jj'})$ and $C_{1ji}(x_{1ji})$ while facility variable costs are linear. And in third scenario (referred as *EOS-FT*), both facility variable costs and transportation costs are considered to be concave functions of flow. For every scenario, we vary λ from 0.2 to 1.0 at an interval of 0.1. The network configurations for these scenarios are depicted in Figures 2.4 and 2.5a - 2.5f. In Table 2.8, we provide the plants and DC locations, and the cost structure of the network for varying levels of economies of scale.

Figure 2.6a and 2.6b depicts the percentage discount in the per unit facility variable and transportation costs in the optimal value as we solve the problems with different values of economies of scale (λ). These percentage discounts are calculated with respect to the values of c_f and c_t in the basecase scenario ($\lambda = 1$). Figure 2.6c shows the effect of varying economies of scale on the average utilization ($\bar{\rho}$) of the facilities (plants and DCs) in the network. The effect of economies of scale on the average shipping distance from plants to customers in the network is shown in Figure 2.6d. In the remainder of the section, we compare the solution under different scenarios.

Basecase Scenario (No Economies of Scale in Operations and Transportation)

Figure 2.4 depicts the configuration of the two-level production-distribution network without any economies of scale in facilities operations and/or transportation (i.e. $\lambda = 1$ for all the facility variables costs at the plants and DCs and transportation costs between facilities). In this case, the optimal solution consists of 11 facilities, i.e. 5 plants and 6 DCs. The plant locations are Queens (NY), Cook (IL), San Bernadino (CA), Dallas (TX) and Miami-Dade

(FL) whereas the DC locations are Suffolk (NY), Cook (IL), San Bernadino (CA), Broward (FL), Tarrant (TX), and Dallas (TX). Counties where plants are located are colored in red and marked with red facility; whereas a county colored in black and marked with black facility represents a DC. We note that, under this scenario, the model opens DCs very close to plants, e.g. Queens (NY) and Suffolk (NY), Tarrant (TX) and Dallas (TX), Miami-Dade (FL) and Broward (FL) or in some cases in the same county, e.g. in Cook county (IL) or San Bernardino (CA). The co-location of plants and DCs minimizes the inbound transportation cost and is attributed to the absence of economies of scale in transportation between the two levels of facilities. The total cost of the configuration is 3,060,741, where the facility fixed cost is 1,052,260 (34%), the facility variable cost is 773,441 (25%), and the transportation cost is 1,235,040 (50%). The plants utilization range from 25% to 60% (with an average of 37%) and the DC utilization range from 41% to 100% with an average of 66%. The average outbound shipping distance (from DCs to customers) is 401 km/unit of demand.

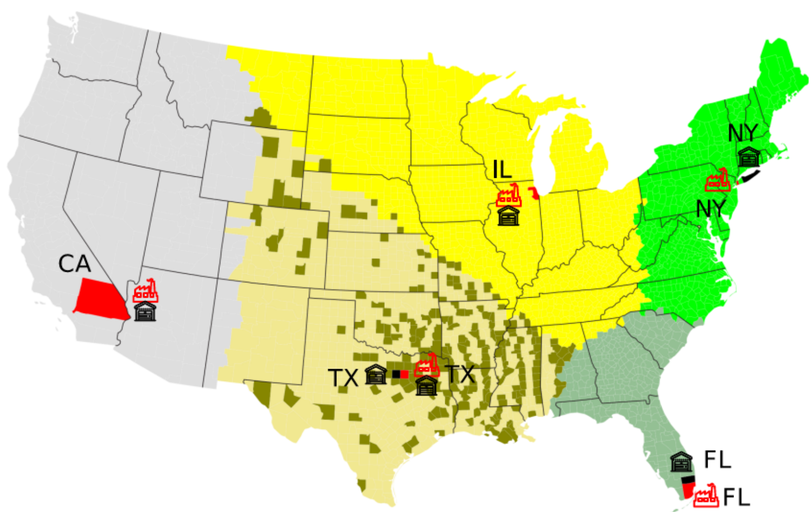


Figure 2.4: Solution without Economies of Scale ($\lambda = 1$)

Effect of Economies of Scale in Facility Operations:

We analyze the effect of economies of scale in facility operations, i.e. production at plants and handling at the DCs, on the configuration of the network. As a result of the economies of scale, the unit facility variable cost at facilities decreases. In the basecase scenario (where $\lambda = 1$), unit variable cost c_f is \$2.51/unit. When λ is set to 0.9, c_f (=\$0.88/unit) is discounted by 65%. When λ is set to 0.8, c_f=\$0.37/unit corresponds to a discount of 85%.

Figures 2.5a and 2.5b show the configuration of the network for $\lambda = 0.9$ and $\lambda = 0.8$ respectively. At $\lambda = 0.9$, the optimal solution locates 10 facilities, 5 plants: Queens (NY), Cook (IL), Clark (NV), Dallas (TX), Miami-Dade (FL) and 5 DCs: Suffolk (NY), Cook (IL), Clark (NV), Dallas (TX), Broward (FL). Compared to the basecase scenario, where the network has 6 DCs, we now have 5 DCs only as the DC in Tarrant (TX) is closed. Note that one of the plant and DC locations changes from San Bernardino (CA) to Clark (NV). The total cost of the configuration is 2,522,567, where the facility fixed cost is 984,100 (39%), the facility variable cost is 271,727 (11%), and the transportation cost is 1,232,170 (50%). The reduction in facility variable costs at plants and DCs compensates for the opening of higher capacity plants and DCs that are now farther from the customers, and hence an increase in transportation cost. As a result of the consolidation, the average outbound shipping distance increases to 412 km/unit of demand. The average DC utilization increases to 76% however the average plant utilization remains unchanged at 37%.

When λ is further decreased from 0.9 to 0.8, location of plants and their capacities remained unchanged, however, the DCs in Suffolk (NY) and Broward (FL) are now relocated to Bronx (NY) and Palm Beach (FL) respectively. As a result of the further reduction in facility variable costs, flows are consolidated at DCs, and hence the model prescribes higher capacity DCs at Bronx (NY) and Palm Beach (FL). The average DC utilization increases to 77% as a result of consolidation. However, the average plant utilization remained unchanged at 37%.

Effect of Economies of Scale in Transportation:

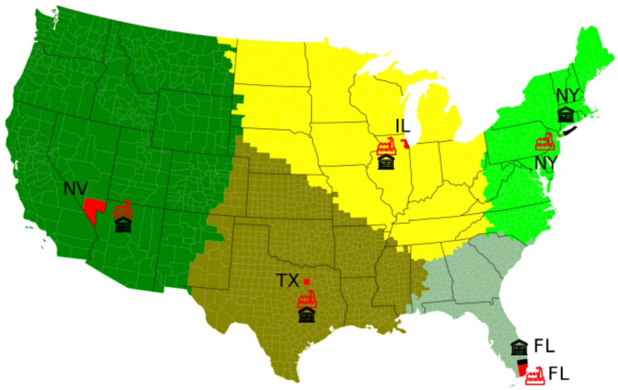
By decreasing λ on the transportation arcs, we incorporate economies of scale in transportation. For instance, when λ=0.8, the unit transportation cost is c_t=\$2.39, which corresponds to a discount of 25% compared to the base case scenario ($\lambda = 1$). Similarly, for λ=0.5, c_t=\$0.44 amounts to a discount of 89%.

Figures 2.5c and 2.5d show the production-distribution network for λ=0.8 and 0.5 respectively. At $\lambda = 0.8$, the model recommends a total of nine facilities, i.e. three plants and six DCs. The plant locations are Cook (IL), Dallas (TX), Miami-Dade (FL) and the DCs locations are Cook (IL), Broward (FL), Sacramento (CA), Philadelphia (PA), Wayne (MI), Tarrant (TX). With discounts in transportation costs, the model recommends locating facilities that are farther from customers as this decreases the total facility fixed cost from \$1,052,260 (in the baseline scenario) to \$588,660.

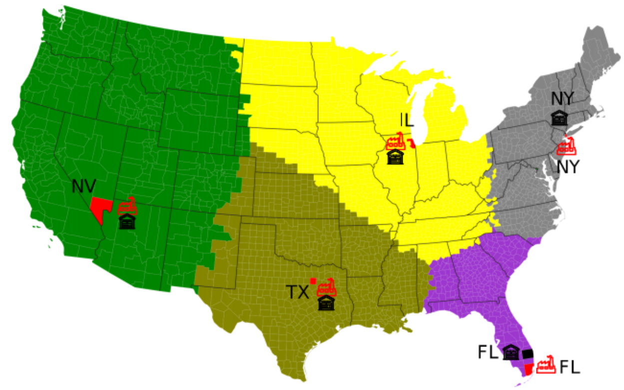
When λ is further decreased to 0.5, although the location and utilization of plants in the new network (Figure 2.5d) do not change, the solution recommends closing the DC in Wayne (MI), improving the average DC utilization to 92%. This results in an increase in the average inbound and outbound shipping distances to 493 kms and 625 kms, respectively.

Effect of Economies of Scale in Facility Operations and Transportation:

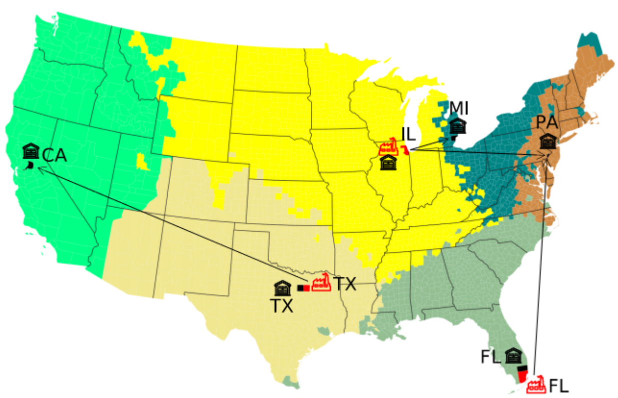
Finally, we analyze the effect of economies of scale in facility operations and transportation on the configuration of the network. We note that when economies of scale is introduced in both, the solution changes significantly. Figures 2.5e and 2.5f illustrate the network configuration for $\lambda = 0.8$ and 0.3, respectively. At $\lambda = 0.8$, the network comprises of four plants and six DCs. The four plant locations are Cook (IL), Dallas (TX), Wayne (MI), and Clark (NV) and the six DC locations are Cook (IL), Clark (NV), Philadelphia (PA), Dallas (TX), Cuyahoga (OH), and Hillsborough (FL). The cost of the network is \$ 1,260,069, where the fixed facility cost (627,880) accounts for 50% of the total cost. The transportation cost is 524,829 (41%) and the facility variable cost is 107,360 (9%). Consolidation of facility



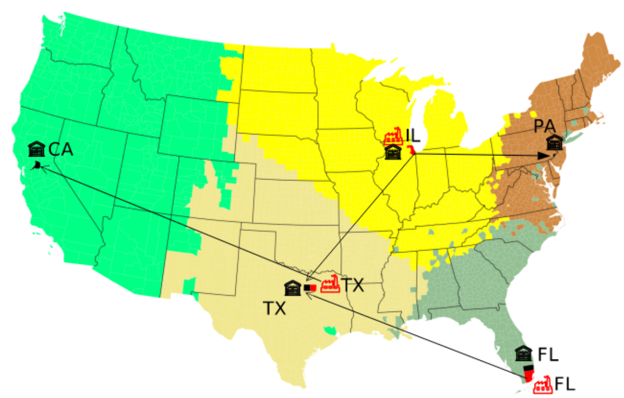
(a) *EOS-F*: $\lambda = 0.9$



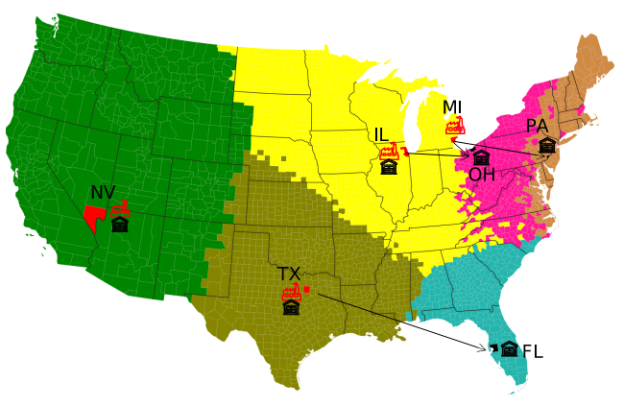
(b) *EOS-F*: $\lambda = 0.8$



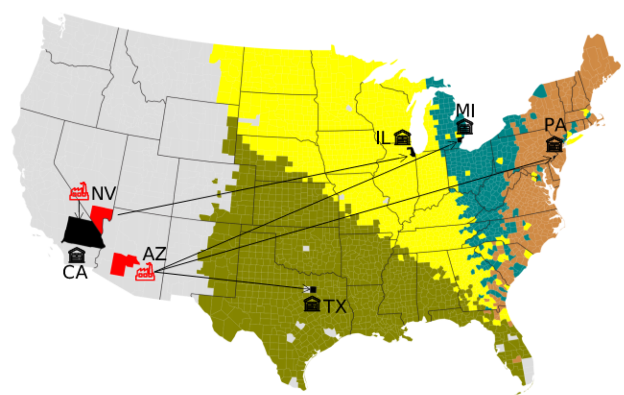
(c) *EOS-T*: $\lambda = 0.8$



(d) *EOS-T*: $\lambda = 0.5$



(e) *EOS-FT*: $\lambda = 0.8$



(f) *EOS-FT*: $\lambda = 0.3$

Figure 2.5: Solution with Economies of Scale only in Operations at Facility (*EOS-F*), only in Transportation (*EOS-T*), and both in Operations at Facility and Transportation (*EOS-FT*)

operations and inbound transportation leads to fewer plants (with large capacity) in the network. The average plant utilization is 71%. At $\lambda = 0.3$, the network has two plants and five DCs; plant locations are Clark (NV) and Maricopa (AZ), and the DC locations are San Bernardino (CA), Cook (IL), Philadelphia (PA), Dallas (TX), and Wayne (MI). The cost of the network is \$ 518,857, where the fixed facility cost (458,600) accounts for 88% of the total cost. Further consolidation of facility operations and transportation leads to fewer plants in the network, resulting in an average plant utilization of 90%.

Figure 2.6c shows that as a result of consolidation of flows at plants and DCs due to economies of scale on facility operations and transportation (EOS-FT), the average facility utilization is higher compared to that in other scenarios where the economies are scale are realized either on the facility operations (EOS-F) or the transportation (EOS-T). For instance, under the EOS-FT scenario, the average facility utilization at $\lambda = 0.6$ is 94%. Similarly, Figure 2.6d depicts that as a result of consolidation of flows at plants and DCs, the average shipping distance also increases. For example, at $\lambda = 0.6$, the average inbound and outbound shipping distances for the network are 787 kms and 1,650 kms (per unit of demand) respectively.

Table 2.8: Sensitivity Analysis: Facilities locations and average utilization, and cost structure for various λs

Economies of Scale	λ		Locations	Total Cost($)	FFC	FVC	TrC	Facility Util.
None	1	Plants:	Queens (NY), Cook (IL), San Bernadino (CA), Dallas (TX), Maimi-Dada (FL)	3,060,741	34%	25%	40%	41
		DCs	Suffolk (NY), Cook (IL), San Bernadino (CA), Dallas (TX), Broward (FL)					68
Operations	0.9	Plants:	Queens (NY), Cook (IL), Clark (NV), Dallas (TX), Miami-Dade (FL)	2,522,567	39%	11%	50%	39
		DCs	Suffolk (NY), Cook (IL), Clark (NV), Dallas (TX), Broward (FL)					80
Operations	0.8	Plants:	Queens (NY), Cook (IL), Clark (NV), Dallas (TX), Miami-Dade (FL)	2,321,305	42%	5%	38%	39
		DCs	Bronx (NY), Cook(IL), Clark (NV), Dallas (TX), Palm Beach (FL)					80
Transportation	0.8	Plants:	Cook (IL), Dallas (TX), Miami-Dade (FL)	1,908,767	31%	31%	38%	87
		DCs	Cook (IL), Broward (FL), Sacramento (CA), Philadelphia (PA), Wayne (MI), Tarrant (TX)					89
Transportation	0.5	Plants:	Cook (IL), Dallas (TX), Miami-Dade (FL)	1,261,084	45%	45%	11%	87
		DCs	Cook (IL), Sacramento (CA), Broward (FL), Philadelphia (PA), Tarrant (TX)					95
Both	0.8	Plants:	Cook (IL), Dallas (TX), Wayne (MI), Clark (NV)	1,260,069	50%	9%	42%	72
		DCs	Cook (IL), Clark (NV), Philadelphia (PA), Dallas (TX), Cuyahoga (OH), Hillsborough (FL)					92
Both	0.3	Plants:	Clark (NV), Maricopa (AZ)	518,857	88%	0%	12%	90
		DCs	San Bernardino (CA), Cook (IL), Philadelphia (PA), Dallas (TX), Wayne (MI)					99

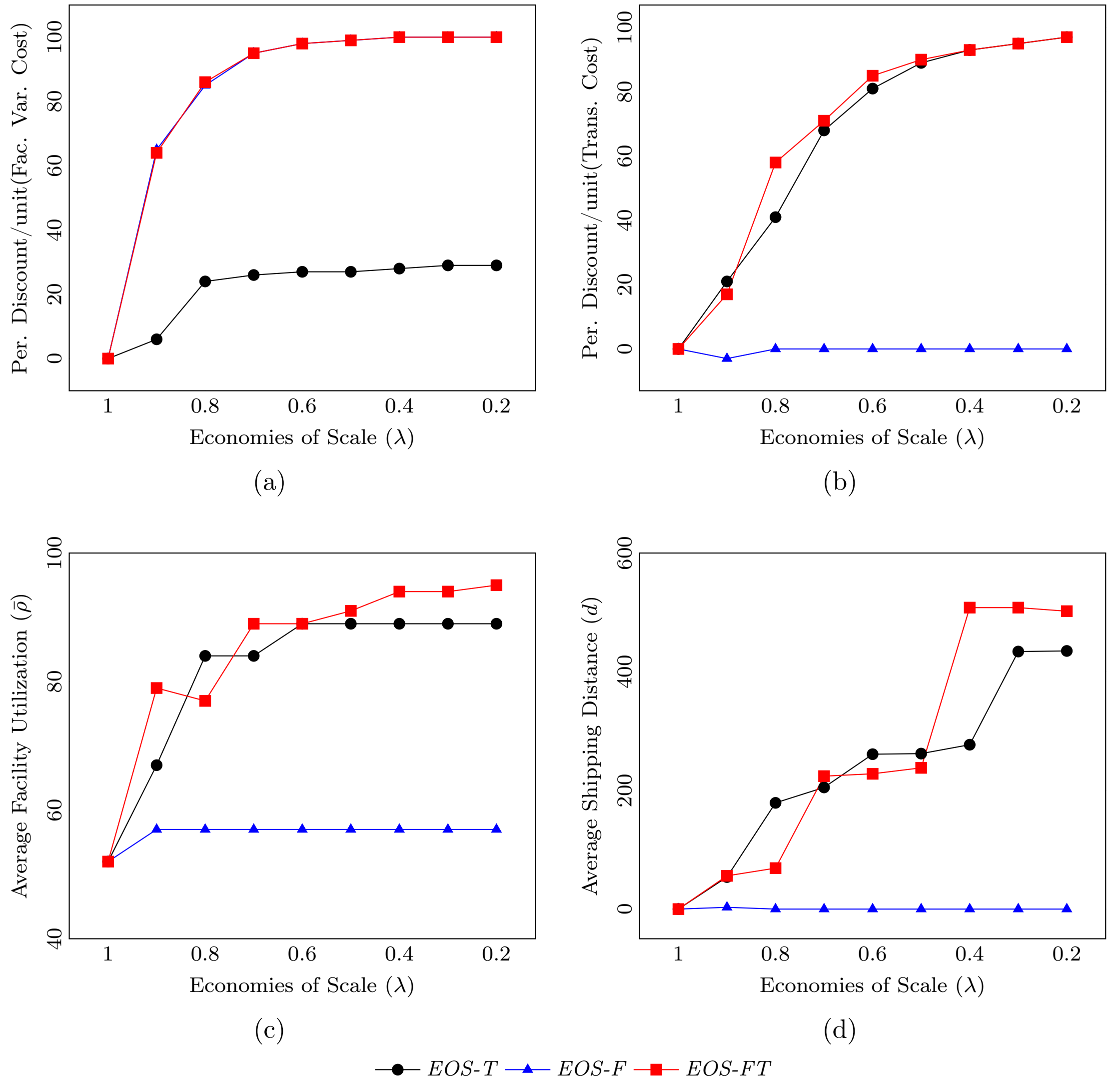


Figure 2.6: Effect of Varying Economies of Scale on Average Facility Utilization, Shipping Distance, and Percentages Discounts in Facility Variables Cost and Transportation Cost

2.6 Conclusions

We studied a general class of multilevel capacitated facility location problems with concave costs, where the concavity arises due to economies of scale in production and handling at the facilities and/or transportation between levels of facilities. We present an alternative mathematical formulation that uses continuous decision variables in which facility location decisions are embedded in a fixed charged function. The resulting nonlinear programming model is compared with prevalent mixed-integer programming formulation that uses binary variables for location decisions and continuous variables for flows. A branch-and-bound algorithm improvised with enhancements such as alternative branching strategies and variable fixing at the root node via preprocessing was presented to solve the problem efficiently. Instances with up to 2,250 customers and 150 potential facilities, and two levels of hierarchy were solved to an optimality gap near to 0.2% under varying costs and capacity structures. The results of extensive computational experiments confirm that the continuous nonlinear programming formulation outperforms the traditional mixed-integer nonlinear formulation on 80 out of 100 instances, especially on medium to large-size instances. A 30% reduction of average CPU time was obtained with the nonlinear formulation compared to the mixed-integer nonlinear formulation. We also analyse the impact of varying economies of scale in the operations at the facilities and/or transportation on the facility location and customer allocation decisions by conducting a sensitivity analysis on instances based on the 3,109 counties in the contiguous United States.

Chapter 3

Stochastic Facility Location with Probabilistic Service Level Constraints on Delivery Times

Abstract

We study a general class of stochastic dynamic capacitated facility location problems arising in the design and reconfiguration of e-commerce supply chains. It considers probabilistic service level constraints to ensure the stochastic delivery times of customers' orders, defined as the sum of waiting time (including service time) at the processing facility plus shipping time to reach the customer location, is within a prescribed time limit with a probability greater or equal to a threshold value. We provide alternative polyhedral representations of these highly nonlinear and non-convex probabilistic constraints, and develop an exact branch-and-cut algorithm for solving the resulting reformulations. The proposed algorithm is enhanced with several algorithmic refinements to accelerate its convergence. We perform extensive computational experiments based on real location data from the United States to evaluate the performance of our algorithm and to provide insights on the optimal structure

of solution networks. Results obtained on large-scale instances with up to 2,500 customers and 225 potential facilities under different service level scenarios confirm the effectiveness of the proposed algorithm.

3.1 Introduction

Supply chain network design involve many interrelated decisions regarding the number, locations, and capacities of different types of facilities, the assignment of retailers to facilities, and the routing of products through the supply chain, among others. Traditionally, such decisions are primarily centered around minimizing the total design and operational costs. With increasing popularity of electronic retailing (e-tailing), the time taken to fulfill customer demand (i.e., delivery time) has also become a prominent factor for businesses when designing their supply chain. In fact, in e-commerce industries response time and shipping charges are important attributes of good customer service and repeat business. Through a global survey, Shopify reports that 59% of customers value free shipping, and 37% are not happy with retailers when orders are delayed (Shopify [114]). The online retailer Amazon meets such expectations through its 'Amazon Prime Program' in which customer orders are delivered within two days (DigitalCommerce360 [35]). Before the COVID-19 pandemic started, Amazon had 118 million Prime members in the US, which has increased to 147 million by the end of first quarter of 2021. To fulfill delivery commitments for growing Prime Memberships, Amazon plans to add 50 new fulfillment facilities in the US in 2021 (Freightos [50]).

The recent growth of omnichannel retailing further emphasizes the importance of strategic network design and capacity planing in e-commerce supply chains to meet delivery time commitments. A rapidly growing number of retailers have started to use warehouses for replenishing stores as well as fulfilling online demand (Hübner et al. [67]). This creates the need to open warehouses closer to densely populated areas to reduce delivery times to their retail stores and online customers. Another application where fast delivery is a necessity is in fresh produce supply chains. As the quality of the food produce rapidly diminishes over time, quick delivery not only improves customer experiences but also reduces waste (Gokarn and Kuthambalayan [56]). Response time is critical in industries that follow make-to-order and assemble-to-order (ATO) strategies (Vidyarthi et al. [124], Aboolian et al. [5]). Although ATO eliminates finished goods inventory and thus, reduces cost and obsolescence, the assembly of finished goods starts only after an order is received thereby making ATO supply chains prone to order delays, high levels of congestion and long response times.

Among several factors, order processing and shipping to customers are two crucial factors that impact the service time needed to deliver orders to customers. The order processing takes place at the facilities and shipping is done by using the company's own fleet of vehicles, a third-party logistics provider, or a mix of both. The order processing comprises of picking items from the shelves, sorting, packing and labeling for delivery. In practice, order processing times at facilities are uncertain due to stochastic demand, variability in orders, and limited order-picking capacity. The order is then picked up by a shipper for the delivery, where shipment time from a facility to a customer is also uncertain due to several factors such as dynamic traffic and weather conditions. Therefore, committing to a short delivery time to customers poses important challenges. Although, it is possible to design a supply chain without taking into account both sources of uncertainty, the obtained network configurations may be sub-optimal from a customer service perspective.

In this paper, we study a general class of *stochastic dynamic capacitated facility location problems* (DCFLP) arising in the design and reconfiguration of e-commerce supply chains. We consider a situation in which customers generate stochastic streams of demand modeled via a Poisson process with dynamic but known arrival rates. Customer orders are allocated by a centralized management system to predetermined facilities for processing and shipment to customers. We consider immobile facilities, each modeled as a single server infinite buffer queue with limited capacities and stochastic order processing times, thereby causing congestion and delays in response times. We consider service level commitments on delivery times defined as the time taken to satisfy a customer demand. Delivery time starts from the moment an order is placed until it is received by the customer. Travel times from facilities to customers are assumed to be uncertain, and we model them as random variables with known probability distribution functions. Therefore, delivery time is the sum of time spent at the facility to process the order and shipment time to customer location. Given that both processing and travel times are uncertain parameters, delivery time is uncertain as well. We incorporate probabilistic service level constraints to guarantee each customer order is delivered within a prescribed time limit, with a probability greater or equal to a threshold value.

The DCFLP considers a dynamic setting with a finite planning horizon in which arrival rates of customer demands change from one period to another and as a consequence, capacity installed at facilities may be modified as needed. A finite set of capacity levels for each potential facility in each period is considered. The change in capacity level is relevant when a retailer can increase processing capacity by hiring part time workers to process the orders during the periods with high demand. The number of opened facilities in each period and their capacity levels are not known a priori, but a fixed set-up cost for each facility and each size is considered. Depending on demand fluctuation between periods, an existing facility may increase, decrease, or even shut down operations from one period to the next. Also, a new facility may be opened in any period and its capacity updated in subsequent periods. The DCFLP seeks to minimize the sum of the facilities fixed costs and variable transportation costs over a finite time planning horizon, while ensuring all demand is satisfied in each period. DCFLPs can also model cases in which there already exists an open set of facilities operating at various capacity levels at the beginning of the planning horizon. However, a limitation of our model is that it abstracts away from few aspects of supply chain planning such as inventory control and positioning, and economies of scale in facility operating and transportation costs, among others.

The main contribution of this work is threefold.

1. We introduce DCFLPs, a general class of stochastic location problems applicable to the design and reconfiguration of e-commerce supply chains. DCFLPs consider a multi-period setting to model the stochastic and dynamic nature of customer demands, stochastic travel times, and service level constraints on delivery times.

2. The problem can be naturally stated as an integer nonlinear program (INLP). We propose two integer linear programming (ILP) formulations containing either an exponential number of cover-like inequality constraints or a compact number of residual service capacity constraints. These constraints provide a polyhedral representation of the probabilistic service level constraints.

3. We develop an exact branch-and-cut algorithm to solve large-scale instances of the DCFLP. To the best of our knowledge, this is the first paper to present an exact algorithm to address probabilistic service level constraints that account for the stochastic shipping time from the facility to the customer's location as well as waiting time (including service times) at the facility. Our algorithm contains several algorithmic refinements to accelerate its convergence. These include: i) exact and heuristic separation routines that efficiently generate cover inequalities at integer and fractional points, ii) a preprocessing phase to fix variables, iii) a matheuristic to provide initial feasible solutions, iv) an in-tree local search heuristic, and v) additional valid inequalities that strengthen the formulation and help improve the overall performance of the enumeration algorithm.

We report the results of extensive computational experiments based on real location data from the USA to evaluate the efficiency and limitations of the proposed solution algorithm. Results obtained on large-scale instances under different service level scenarios confirm the effectiveness of our exact algorithm. We also provide detailed sensitivity analyzes to assess the optimal structure of solutions under different input parameter settings.

The remainder of the paper is organized as follows. In Section 3.2, we present relevant literature to our work, followed by a formal definition of the problem in Section 3.3. Section 3.4 describes two ILP formulations for the DCFLP. Section 3.5 describes the details of our branch-and-cut algorithm, and several enhancements to improve its convergence. The results of extensive computational experiments and sensitivity analyzes are presented in Section 3.6. Conclusions follow in Section 3.7.

3.2 Literature Review

E-commerce companies must consider several aspects while designing and reconfiguring their supply chain networks. In what follows, we present the literature relevant to the factors considered in this paper: *i)* stochastic demand and travel times, *ii)* dynamic location, and *ii)* congestion and probabilistic service level constraints. We refer to Cordeau et al. [31] for an extensive review of other important factors relevant to the design of supply chains not considered below.

Discrete location problems (DLP) are fundamental to the design of supply chains. DLPs with immobile facilities, stochastic demands and congestion capture the trade-off between fixed set-up costs for installing/modifying facilities, variable transportation costs, and customer expectations on service time. This trade-off has been studied from different perspectives. Some works explicitly include probabilistic constraints ensuring, with a desired probability, that customers are served within a threshold time limit. The total cost increases for systems with higher probabilities and/or lower threshold wait times. Others studies implicitly model waiting time as a cost in the objective function. These problems seek a balance between other logistics and wait time costs to improve customer experience. Whereas optimizing wait time cost improves the overall system-wide service level, it does not guarantee a minimum service level. For a comprehensive review on this challenging class of location problems, we refer to Berman and Krass [17] and for earlier works, we refer to Berman and

Krass [16] and Boffey et al. [18].

Marianov and Serra [88], Marianov and Serra [89], and Marianov and Serra [90] consider probabilistic constraints to restrict wait time/queue length at a facility to a predetermined value. They model a multi-server queuing system at each facility with service request as a random Poisson process, and service time as exponentially distributed random variable. The assumptions allow to state the probabilistic constraints in a tractable linear form. The resulting mixed integer linear programs are solved using Lagrangian relaxation and heuristics. Although uncertainty in waiting time is addressed in the form of constraints in these works, they do not explicitly study the impact of travel times uncertainties between customers and facilities on the order delivery time for every customer.

Travel times are also critical in location-allocation decisions to ensure minimum service guarantee. In a study on vehicle inspection stations and health clinics, Grossman and Brandeau [59] report that users' decision to patronize a facility depends on both travel time and waiting time at facilities. However, few studies have investigated the effect of jointly considering waiting times and stochastic travel times. Wang et al. [126] study the location of immobile facilities with the objective of minimizing system wide customers' waiting time costs and travel costs. A generalized version of this problem with a fixed number of servers to locate among open facilities is studied in Berman and Drezner [15]. This work is then further extended in Aboolian et al. [2] by including fixed cost to open facilities and variable cost of servers, in addition to travel and waiting costs. To locate hospitals and medical centers, Aboolian et al. [3] work with similar problem settings. It is assumed that facilities are fixed, customers choose nearest facility, and demand and service times are stochastic. However, the considered objective is to minimize the maximum travel time and waiting time at the facility. Baron et al. [11] consider a continuous facility location problem in which demand is assumed to be spatially distributed, and arrival processes and service times follow a general distribution. To reduce congestion and ensure adequate service, they consider either deterministic maximum coverage distance constraints or probabilistic service level constraints to limit waiting time at a facility to a maximum value. However, our model is different from these works in the way we consider probabilistic constraints that jointly capture uncertainties from both waiting and travel times to ensure minimum service level with a desired probability. The work of Aboolian et al. [5] is most closely related to our problem. The response time in their problem is also the sum of waiting time at facilities and travel time from facilities to customers. However, a major difference between their and ours is that we consider travel time to be uncertain, whereas they assume deterministic travel times. We also generalize their model by considering planning over multiple time periods, and capacity at facilities can be adjusted according to the demand in that period.

Most stochastic DLPs assume that facilities once opened would be operational in many future periods and their capacity levels would not change. However, such assumptions may not be suitable in some applications. For instance, in e-commerce supply chains, the current network may become sub-optimal (or capacities may not be sufficient) to handle a decrease (or increase) in demand. In the case of mergers and acquisitions, it may be optimal to close existing facilities and/or open new ones in the future. Dynamic variants of capacitated DLPs can be classified into two forms: capacity expansion via addition of extra capacity in a period, and choice of capacity type at a facility in each period. Jena et al. [74] study a rather general class of dynamic DLPs with modular capacities in which at any time of the planning horizon, their model allows (re)opening, (re)closing of facilities, and increasing/decreasing their capacity levels. These problems generalize a large set of models in the dynamic DLPs literature. Our proposed DCFLP is a further generalization of the deterministic model presented in Jena et al. [74] in which stochasticity in customers' demand and travel times are considered, in combination with probabilistic service level constraints on delivery times. We refer to Farahani and Hekmatfar [42] and Saldanha-da Gama and Nickel [110] (and references therein) for detailed discussion on dynamic DLPs.

3.3 Problem Definition and Formulation

Let V_1 be the set of potential facility locations, I be the set of customer nodes, $T = \{1, 2, .., |T|\}$ be the set of time periods in the planning horizon, and $L = \{0, 1, 2, .., |L|\}$ be the set of potential capacity levels for each facility. Let c_{ijlt} denote the unit cost to serve demand of node $i \in I$ from facility $j \in V_1$ opened at capacity level $l \in L$ in period $t \in T$. The parameter c_{ijlt} is composed of two components: (i) transportation cost from facility j to customer i, and (ii) production cost at level l at facility j in period t. Moreover, $f_{jl_1l_2t}$ represents the fixed cost of changing the capacity level at facility $j \in V_1$ from $l_1 \in L$ to $l_2 \in L$ at the beginning of period $t \in T$, and it also includes the cost of operating the facility j at level l_2 during entire period t. Depending on the values of l_1 and l_2, $f_{jl_1l_2t}$ (where, f_{j00t} $= 0$) captures various scenarios related to capacity levels and facility decisions at j at the beginning of each period t. For instance, for $l_1 = 0$ and $l_2 \in L \setminus \{0\}$, $f_{jl_1l_2t}$ represents the sum of opening and operating costs of a new facility opened at level l_2 in period t. However, when $l_1 \in L \setminus \{0\}$ and $l_2 = 0$, $f_{jl_1l_2t}$ corresponds to the closing cost of facility j. In other situations where l_1 and l_2 are nonzeroes, for $l_1 < l_2$, $f_{jl_1l_2t}$ represents the sum of capacity expansion and operating costs of facility j; whereas for $l_1 > l_2$, it represents the capacity reduction and operating costs of facility j; and for $l_1 = l_2$, $f_{jl_1l_2t}$, it corresponds to the operational cost.

We consider an inelastic demand where the demand rate for each customer $i \in I$ and period $t \in T$ follows a Poisson process with constant mean rate λ_{it} (Aboolian et al. [5], Berman and Krass [17]). A period can be interpreted as a long duration for which planning needs to be done. For example, $t \in T$ can be a month, which is a resonable assumption because many companies forecast monthly demand (Nahmias and Olsen [96]), and we assume that λ_{it} is the daily average demand from customer i during month t. Similarly, the service rate is also defined as average number of daily orders that can be processed at facility $j \in V_1$ during entire period $t \in T$. We assume independent inter-arrival times of the demand processes generated by customers. Moreover, each potential facility is considered to be single-server with limited capacity (Baron et al. [11], Vidyarthi and Jayaswal [122]). If facility $j \in V_1$ is opened at period $t \in T$, we must decide its service capacity Ω_{jt}, which can be interpreted as its average processing rate. Similar to [122], we assume the capacity Ω_{jt} can be selected from a predetermined finite set of values, i.e., $\Omega_{jt} \in \left\{\mu_{jt}^1, \cdots, \mu_{jt}^{|L|}\right\}$, where μ_{jt}^l, $l \in L$ corresponds to the discrete set of available choices for service levels for facility j during any period t. We consider stochastic *service (or order-processing) times* at each opened facility and a first-come-first-serve (FCFS) service discipline. Also, service times are assumed to be independent and identically distributed random variables that follow an exponential distribution. Since ecommerce supply chain is similar to make-to-order supply chain, as in both of them order processing starts after orders are received. Therefore, we can model each facility in our problem as a queuing system. And, similar to the paper by Aboolian et al. [5], we also consider M/M/1 queuing system at every open facility in each period $t \in T$. Under steady-state conditions, the waiting time W_{jt} in this system (including service time) follows exponential distribution.

The DCFLP consists of selecting the location and service capacity of a set of facilities and the assignment of each customer to exactly one opened facility (i.e., single sourcing) for each period of the planning horizon. The objective is to minimize the total setup, operational and servicing cost. More importantly, the DCFLP considers independent *probabilistic service level constraints* for each customer $i \in I$ and period $t \in T$ to guarantee that the (stochastic) delivery time DT_{it}, defined as the sum of waiting time at its assigned facility plus the shipping time from such facility to node i, does not exceed a threshold value τ_{it} with a minimum probability of θ_{it}.

We define the binary variable $z_{jl_1l_2t}$ equal to one if and only if facility $j \in V_1$ changes its capacity level from $l_1 \in L$ to $l_2 \in L$ at the beginning of period $t \in T$, and operates at l_2 during period t. We also define the binary assignment variable x_{ijlt} equal to one if and only if the customer node $i \in I$ is allocated to facility at node $j \in V_1$ operating at capacity level $l \in L$ in period $t \in T$.

Each opened facility observes aggregate demand from a subset of nodes assigned to it.

Due to the superposition principle of Poisson processes, the demand arrival rate at facility $j \in V_1$ in period $t \in T$ is also a Poisson process with mean rate $\Lambda_{jt}(x) = \sum_{i \in I} \sum_{l \in L} \lambda_{it} x_{ijlt}$. Given that capacity levels at each facility are represented using a discrete set L, and we select only one capacity level at each facility, the service rate of facility j in period t can be stated as, $\Omega_{jt}(z) = \sum_{l_1 \in L} \sum_{l \in L} \mu^{l}_{jt} z_{jl_1lt}$.

Under system stability conditions, i.e., $\Lambda_{jt}(x) \leq \Omega_{jt}(z)$, $j \in V_1$, $t \in T$, the steady-state system waiting time $W_{jt}(x,z)$ follows an exponential distribution. We recall that the exponential distribution is a special case of the Gamma distribution with shape parameter $\alpha_{1jt} = 1$ and scale parameter $\beta_{1jt} = \frac{1}{\Omega_{jt}(z) - \Lambda_{jt}(x)}$. Similar to Polus [103], for each shipping arc $(i,j) \in I \times V_1$ we assume the *shipping time* ST_{ij} to be a random variable that follows a gamma distribution. For each $(i,j) \in I \times V_1$, let $E[ST_{ij}]$ be the mean travel time (in days), and CV_{ij} be the coefficient of variation of travel time. The shape and scale parameters for ST_{ij} are $\alpha_{2ij} = \frac{1}{CV_{ij}^2}$ and $\beta_{2ij} = \frac{E[ST_{ij}]}{\alpha_{ij}}$, respectively. For each $i \in I$ and $t \in T$, the *delivery time* is given by $DT_{it}(x,z) = \sum_{j \in V_1} \sum_{l \in L} \left(W_{jt}(x,z) + ST_{ij}\right) x_{ijlt}$, and its associated probabilistic service level constraints can then be stated as:

$$\mathcal{P}_{it}(DT_{it}(x,z) \leq \tau_{it}) \geq \theta_{it}, \tag{3.1}$$

where $\mathcal{P}_{it}(\cdot)$ is the steady-state distribution of DT_{it}, τ_{it} is the specified threshold for the delivery time, and θ_{it} is the minimum acceptable probability of delivery times smaller or equal to τ_{it}. The parameter θ_{it} can be interpreted as the minimum acceptable proportion of customers requests originating at node i in period t for which their orders will be received within the promised delivery times. For each $i \in I$ and $t \in T$, the random variable DT_{it} is defined as the sum of two gamma random variables. The shipping activities begin only after the orders are prepared, therefore, the two sources of uncertainty are assumed to be independent. Therefore, the steady-state distribution $\mathcal{P}_{it}(\cdot)$ of DT_{it} is given as (Mathai [91]):

$$\mathcal{P}_{it}(DT_{it}\,(x,z) \leq \tau_{it}) = \left(\frac{\beta_1}{\beta_2}\right)^{\alpha_2} \sum_{k=0}^{\infty} \binom{\alpha_2+k-1}{k} \left(1-\frac{\beta_1}{\beta_2}\right)^{k} G\left(\frac{\tau}{\beta_1}; k+\gamma\right) DT_{it}, \qquad (3.2)$$

where $G\left(\frac{\tau_{it}}{\beta_1}; k+\gamma\right)$ is the *cumulative distribution function* (CDF) of a gamma random variable with shape parameter equal to $k+\gamma$, scale parameter equal to 1, $\gamma = (\alpha_{1jt} + \alpha_{2ij})$, and $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (\alpha_{1jt}, \beta_{1jt}, \alpha_{2ij}, \beta_{2ij})$, if $\beta_{1jt} \leq \beta_{2ij}$, and $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (\alpha_{2ij}, \beta_{2ij}, \alpha_{1jt}, \beta_{1jt})$, otherwise. For a detailed derivation of distribution (3.2), we refer to Mathai [91] and Hu et al. [65]. For a given solution (x, z), the steady-state distribution $\mathcal{P}_{it}(\cdot)$ of DT_{it} can be evaluated using the algorithm of Hu et al. [65]. Finally, we can state the DCFLP as the following INLP:

$$\text{(INLP)} \quad \text{minimize} \sum_{j\in V_1}\sum_{l_1\in L}\sum_{l_2\in L}\sum_{t\in T} f_{jl_1l_2t} z_{jl_1l_2t} + \sum_{i\in I}\sum_{j\in V_1}\sum_{l\in L}\sum_{t\in T} c_{ijlt}\lambda_{it}x_{ijlt} \qquad (3.3)$$

$$\text{subject to} \sum_{j\in V_1}\sum_{l\in L} x_{ijlt} = 1 \ \ \forall i \in I, t \in T \qquad (3.4)$$

$$\sum_{i\in I} \lambda_{it} x_{ijlt} \leq \mu^l_{jt} \sum_{l_1\in L} z_{jl_1lt} \ \ \forall j \in V_1, l \in L, t \in T \qquad (3.5)$$

$$\sum_{l_1\in L} z_{jl_1l(t-1)} = \sum_{l_2\in L} z_{jll_2t} \ \ \forall j \in V_1, l \in L, t \in T \setminus \{1\} \qquad (3.6)$$

$$\sum_{l_2\in L} z_{jl^jl_21} = 1 \ \ \forall j \in V_1 \qquad (3.7)$$

$$\mathcal{P}_{it}(DT_{it}\,(x,z) \leq \tau_{it}) \geq \theta_{it}, \ \ \forall i \in I, t \in T \qquad (3.8)$$

$$x_{ijlt} \in \{0,1\} \ \ \forall i \in I, j \in V_1, l \in L, t \in T \qquad (3.9)$$

$$z_{jl_1l_2t} \in \{0,1\} \ \ \forall j \in V_1, l_1 \in L, l_2 \in L, t \in T. \qquad (3.10)$$

The objective (3.3) minimizes the total cost of opening, closing, and changing the capacity levels of facilities, of operating the facilities, and of serving the customers. Constraints (3.4) are the demand satisfaction constraints which guarantee that all demand generated at each node i will be met at each period t. Constraints (3.5) are the disaggregated version of the system stability conditions, i.e., $\Lambda_{jt}(x) \leq \Omega_{jt}(z)$, $j \in V_1$, $t \in T$. Constraints (3.6) and (3.7) are flow balance constraints that model the change in capacity levels from one time period to another and ensure that, for each facility exactly one capacity level is selected at each period. Constraints (3.8) are the probabilistic service level constraints to guarantee that the probability of the delivery time within τ_{it} units of time is at least θ_{it}. Finally, constraints (3.9) and (3.10) are standard integrality restrictions.

In the following section, we introduce two sets of linear constraints which are equivalent to the probabilistic constraints (3.8). These inequalities can be used to derive valid reformulations of the DCFLP in which the feasible region (3.4)–(3.10) is characterized with a polyhedron. Baron et al. [11] and Berman and Krass [17] suggest the use of tractable approximations for probabilistic service level constraints when considering only wait times at facilities.

3.4 Polyhedral Representations of Probabilistic Service Level Constraints

We next present two alternative ILP reformulations for the DCFLP that exploit the single-sourcing assumption and the fact that steady-state probabilities $\mathcal{P}_{it}(DT_{it}(x,z) \leq \tau_{it})$ can be evaluated if the values of the decision variables (x, z) are known.

For each $j \in V_1$, $l \in L$, and $t \in T$, we denote as $\mathcal{C}_{jlt}$ the set of subsets of customers such that if assigned to facility j using capacity level l in period t, would lead to at least one violated probabilistic service level constraint, i.e.,

$$\mathcal{C}_{jlt} = \left\{ C \subseteq I : \exists i \in C, \mathcal{P}_{it}(\overline{W}_{jt}(C,l) + ST_{ij} \leq \tau_{it}) < \theta_{it}, \right\},$$

where $\overline{W}_{jt}(C,l)$ is the steady-state waiting time of an M/M/1 queuing system with demand arrival rate $\Lambda_{jt}(C) = \sum_{i \in C} \lambda_{it}$ and service rate $\Omega_{jt} = \mu^l_{jt}$. A simple but useful observation is that any infeasible solution (x, z) in which the subset of customers $C \in \mathcal{C}_{jlt}$ is assigned to facility j in period t can be removed from the feasible region by the following *cover inequality* (CI):

$$\sum_{i \in C} x_{ijlt} \leq (|C| - 1) \sum_{l_1 \in L} z_{jl_1lt}.$$

The following result shows that CIs are sufficient to provide a valid formulation for the DCFLP.

Proposition 1. *The following ILP1 is a valid formulation for the DCFLP:*

$$\begin{aligned}
(ILP1) \quad \textit{minimize} \quad & \sum_{j \in V_1} \sum_{l_1 \in L} \sum_{l_2 \in L} \sum_{t \in T} f_{jl_1l_2t} z_{jl_1l_2t} + \sum_{i \in I} \sum_{j \in V_1} \sum_{l \in L} \sum_{t \in T} c_{ijlt} \lambda_{it} x_{ijlt} \\
\textit{subject to} \quad & (3.4) - (3.7), (3.9), (3.10) \\
& \sum_{i \in C} x_{ijlt} \leq (|C| - 1) \sum_{l_1 \in L} z_{jl_1lt} \quad \forall j \in V_1, l \in L, t \in T, C \in \mathcal{C}_{jlt}. \quad (3.11)
\end{aligned}$$

Proof. See Section B.1 in the Appendix. □

The second ILP reformulation is based on the following observation. If facility $j \in V_1$ is opened at level $l \in L$ in period $t \in T$, i.e., $z_{jl_1lt} = 1$ for some l_1, and customer $i \in I$ is assigned to it, i.e., $x_{ijlt} = 1$, then the *residual service capacity* available to assign other customers $i' \in I \setminus \{i\}$ to j is $\mu^l_{jt} - \lambda_{it} \geq 0$. However, this residual service capacity does not take into account that the probabilistic constraint of customer i at period t must be satisfied. We can estimate the maximum demand Δ_{ijlt} from one or several customer nodes $i' \in I \setminus \{i\}$ that can be assigned to facility j at level l in period t (in addition to the demand λ_{it}) such that the probabilistic constraint remains feasible, i.e., $\mathcal{P}_{it}(W^*_{jt} + ST_{ij} \leq \tau_{it}) = \theta_{it}$, where W^*_{jt} is the waiting time with mean demand arrival rate $\Lambda_{jt} = \lambda_{it} + \Delta_{ijlt}$ and service rate $\Omega_{jt} = \mu^l_{jt}$. For each $i \in I$, $j \in V_1$, $l \in L$, and $t \in T$, we can use Δ_{ijlt} to derive the following valid inequality:

$$\sum_{i' \in I \setminus \{i\}} \lambda_{i't} x_{i'jlt} \leq \Delta_{ijlt},$$

where $0 \leq \Delta_{ijlt} \leq \mu_{jt}^{l} - \lambda_{it}$. The following result shows how these inequalities can be extended to provide a valid formulation for the DCFLP.

Proposition 2. *The following ILP2 is a valid formulation for the DCFLP:*

$$\begin{aligned}
(ILP2) \ \ \textit{minimize} \ \ & \sum_{j \in V_1} \sum_{l_1 \in L} \sum_{l_2 \in L} \sum_{t \in T} f_{jl_1l_2t} z_{jl_1l_2t} + \sum_{i \in I} \sum_{j \in V_1} \sum_{l \in L} \sum_{t \in T} c_{ijlt} \lambda_{it} x_{ijlt} \\
\textit{subject to} \ \ & (3.4) - (3.7), (3.9), (3.10) \\
& \sum_{i' \in I \setminus \{i\}} \lambda_{i't} x_{i'jlt} \leq \mu_{jt}^{l} \sum_{l_1 \in L} z_{jl_1lt} - \left(\mu_{jt}^{l} - \Delta_{ijlt}\right) x_{ijlt} \\
& \qquad \forall i \in I, j \in V_1, l \in L, t \in T, \mu_{jt}^{l} - \Delta_{ijlt} > \lambda_{it},
\end{aligned} \tag{3.12}$$

where Δ_{ijlt} is the maximum demand that can be assigned to facility j at level l in period t (in addition to the demand λ_{it}) such that $\mathcal{P}_{it}(W_{jt}^{*} + ST_{ij} \leq \tau_{it}) = \theta_{it}$ for customer i.

Proof. See Section B.2 in the Appendix. □

From now on, we denote constraints (3.12) as residual service capacity (RSC) constraints. Finally, note that none of these new inequalities (CI and RSC) dominate the other.

3.5 Exact Solution Algorithm

We now present an exact branch-and-cut algorithm that exploits both sets of CI and RSC constraints. To speed up its convergence, we propose the following refinement strategies: *(i)* we include some additional families of valid inequalities that help strengthen the linear programming (LP) relaxation bounds, *(ii)* we use heuristic separation routines at fractional solutions to efficiently derive violated CIs, *(iii)* we employ a pre-processing step to eliminate a subset of the assignment variables x_{ijlt}, *(iv)* we solve a perturbed version of the DCFLP to provide a good initial upper bound at the root node, and *(v)* we use a local search heuristic to improve the feasible integer solutions obtained at the nodes of the enumeration tree.

3.5.1 Valid Inequalities

A natural extension of the CIs introduced in Section 3.4 is to define an extended cover. To do so, we must consider the stochastic nature of demands and travel times when defining the extended cover. The following result on the steady-state distribution of waiting times is needed to identify feasible extended covers.

Lemma 3.5.1. *For each $i \in I$ and $t \in T$, $\mathcal{P}_{it}(DT_{it}(x,z) \leq \tau_{it})$ is monotonically nondecreasing with respect to the residual service capacity $(\Omega_{jt}(z) - \Lambda_{jt}(x))$ at any facility $j \in V_1$ in period $t \in T$.*

Proof. See Section B.3 in the Appendix. □

Using the above property, we next provide two families of *extended cover inequalities* (ECIs) which differ in the way extended covers are defined.

Proposition 3. *For each $j \in V_1$, $l \in L$ and $t \in T$, if $C \in \mathcal{C}_{jlt}$ and $E(C) = C \cup i$, where $i \in I \setminus C$, such that: (1) $\lambda_{it} \geq \max_{i' \in C} \lambda_{i't}$, and (2) for any $i' \in C : \mathcal{P}_{i't}(DT_{i't}(x,z) \leq \tau_{i't}) < \theta_{i't}$, $E[ST_{ij}] \geq E[ST_{i'j}]$ and $CV_{ij} \geq CV_{i'j}$, the ECI*

$$(ECI_1) \sum_{i \in E(C)} x_{ijlt} \leq (|C| - 1) \sum_{l_1 \in L} z_{jl_1lt}, \tag{3.13}$$

is valid for X, where X denotes the set of feasible solutions of the DCFLP. Furthermore, (ECI_1) dominates the cover inequality (3.11).

Proof. See Section B.4 in the Appendix. $\square$

Proposition 4. *For $j \in V_1$, $l \in L$ and $t \in T$, if*
$C \in \mathcal{C}_{jlt}$ and $E(C) = C \cup \{i \in I : \lambda_{it} \geq \max_{i' \in C} \lambda_{i't}\}$, the following ECI is valid for X:

$$(ECI_2) \qquad \sum_{i \in E(C)} \sum_{l'=1}^{l} x_{ijlt} \leq |C| - 1. \tag{3.14}$$

Proof. See Section B.5 in the Appendix. $\square$

The third set of valid inequalities are an adaptation of the inequalities given in Marianov and Serra [88] to linearize probabilistic service level constraints. If we ignore travel times in INLP and consider only waiting times at facilities, while assuming all τ and θ parameters to be equal (i.e., do not depend on i and t), the probabilistic constraints (3.8) reduce to

$$\mathcal{P}(W_{jt} \leq \tau) \geq \theta \qquad \forall j \in V_1, t \in T, \tag{3.15}$$

where $\mathcal{P}(W_{jt} \leq \tau_{it})$ is the steady-state distribution of the waiting time W_{jt}. Thus, for any $l > 0$, the following inequalities are valid (Marianov and Serra [88]):

$$\sum_{i \in I} \lambda_{it} x_{ijlt} \leq \mu_{jt}^{l} + \frac{1}{\tau} \ln(1 - \theta) \qquad \forall j \in V_1, l \in L \setminus \{0\}, t \in T. \tag{3.16}$$

Given that in INLP we consider that service probability (θ_{it}) and delivery time (τ_{it}) depend on i and t, we must modify the second term of the right-hand-side of (3.16) as follows:

$$\sum_{i \in I} \lambda_{it} x_{ijlt} \leq \mu_{jt}^{l} + \max_{i \in I} \left\{ \frac{1}{\tau_{it}} \ln(1 - \theta_{it}) \right\} \qquad \forall j \in V_1, l \in L \setminus \{0\}, t \in T. \tag{3.17}$$

We note that the second term must consider the customer giving the most conservative reduction of service rate in order to guarantee validity.

The last two types of inequalities are given in Jena et al. [74] for the deterministic DCFLP:

$$x_{ijlt} \leq \sum_{l_1 \in L} z_{jl_1lt} \quad \forall i \in I, \forall j \in V_1, \forall l \in L, \forall t \in T \tag{3.18}$$

$$\sum_{j \in V_1} \sum_{l_1 \in L} \sum_{l_2 \in L} \mu_{jt}^{l_2} z_{jl_1l_2t} \geq \sum_{i \in I} \lambda_{it} \quad \forall t \in T. \tag{3.19}$$

Constraints (3.18) are known to strengthen the LP bounds, whereas (3.19) are particularly useful when used with commercial solvers to derive cover cuts to further improve LP bounds.

3.5.2 Separation Problem for Cover Inequalities

Given a fractional point ($\hat{x}$, $\hat{z}$), the separation problem determines whether this point satisfies all CIs (3.11) or if there exists at least one violated cover. We rewrite (3.11) as:

$$\sum_{i \in C} \Big(\sum_{l_1 \in L} z_{jl_1lt} - x_{ijlt} \Big) \geq \sum_{l_1 \in L} z_{jl_1lt} \quad \forall j \in V_1, \forall l \in L, \forall t \in T, C \in \mathcal{C}_{jlt}. \tag{3.20}$$

For every $j \in V_1, l \in L, t \in T$, given ($\hat{x}$, $\hat{z}$), the separation problem seeks to find a set $C \in \mathcal{C}_{jlt}$ such that $\mathcal{P}_{it} < \theta_{it}$ for at least one $i \in C$, and $\sum_{i \in C} \Big(\sum_{l_1 \in L} \hat{z}_{jl_1lt} - \hat{x}_{ijlt} \Big) < \sum_{l_1 \in L} \hat{z}_{jl_1lt}$. For each $j \in V_1, l \in L, t \in T$, we solve an independent separation problem. For each $i \in I$, let variable v_i be equal to one if and only if $i \in C$, and let p'_i and p_i be the auxiliary variables for evaluating probabilities. The separation problem of CIs can be stated as follows:

$$(\mathrm{SP}_{jlt}) \quad \zeta_{jlt} = \text{minimize} \quad \sum_{i \in I} \Big(\sum_{l_1 \in L} \hat{z}_{jl_1lt} - \hat{x}_{ijlt} \Big) v_i \tag{3.21}$$

$$\text{subject to} \quad \sum_{i \in I} p'_i < \Big(\sum_{i \in I} v_i \theta_{it} \Big) \tag{3.22}$$

$$p'_i = \begin{cases} p_i & \text{, if } p_i < \theta_{it} \\ \theta_{it} & \text{, if } p_i \geq \theta_{it} \end{cases} \tag{3.23}$$

$$p_i = \mathcal{P}_{it}\Big(W_{jt} + ST_{ij} \leq \tau_{it}\Big)v_i \quad \forall i \in I \tag{3.24}$$

$$v_i \in \{0,1\}, p_i, \geq 0, p'_i \geq 0 \quad \forall i \in I. \tag{3.25}$$

Constraint (3.24) determines the probability of serving a customer. The value of variable W_{jt} is based on the assignment variable v_i. If there exists an optimal solution, at least one probabilistic constraint is violated, otherwise problem SP_{jlt} is infeasible. To show this, we observe that variable p'_i equals to θ_{it} from constraint (3.23) only when desired probability to serve customer $i \in C$ on time is met. If every customer $i \in C$ is satisfied with its desired probability, then constraint $\sum_{i \in C} p'_i < \Big(\sum_{i \in C} v_i \theta_{it}\Big)$ is violated. Therefore, to satisfy (3.22) at least one of the customers have to be served with probability strictly smaller than θ_{it}, otherwise the problem SP_{jlt} is infeasible. Since SP_{jlt} is still an integer non-convex program, we solve the this problem heuristically. The proposed heuristic is depicted in Section B.6 in the Appendix.

3.5.3 Separation Problem for Residual Service Capacity Constraints

Given a fractional point $(\hat{x}, \hat{z})$, we check for the violation of inequalities (3.12). If for any $i \in I, j \in V_1, l \in L, t \in T$, $\sum_{i' \in I \setminus \{i\}} \lambda_{i't} \hat{x}_{i'jlt} + \big(\mu^l_{jt} - \Delta_{ijlt}\big) \hat{x}_{ijlt} - \mu^l_{jt} \sum_{l_1 \in L} \hat{z}_{jl_1lt} > ST$, we add the respective inequality. To compute Δ_{ijlt}, we need to know for which value of Λ_{jt}, we have $\mathcal{P}_{it} = \theta_{it}$. We define the demand arrival rate as $\Lambda_{jt} = \lambda_{it} + \Delta_{ijlt}$ and the service rate as $\Omega_{jt} = \mu^l_{jt}$. Therefore, the net residual service capacity in that case is given as $\phi_{ijlt} = \mu^l_{jt} - \lambda_{it} - \Delta_{ijlt}$. We use a line search method to calculate the value of ϕ_{ijlt}, and subsequently Δ_{ijlt}. For a given $i \in I, j \in V_1, l \in L, t \in T$, we initialize $\phi_{min} = 0.0001$, $\phi = 5$ and $\phi_{max} = (0.975) * \mu_{jl}$, and calculate the value of $\mathcal{P}_{it}$. If $\mathcal{P}_{it} < \theta_{it} - \epsilon$ then $\phi_{min} = \phi$ and $\phi = (\phi_{min} + \phi_{max})/2$; whereas if $\mathcal{P}_{it} > \theta_{it}$ then $\phi_{max} = \phi$ and $\phi = (\phi_{min} + \phi_{max})/2$. In either case $\mathcal{P}_{it}$ is recalculated, and the procedure continues until $\mathcal{P}_{it} \in [\theta_{it} - \epsilon, \theta_{it}]$. We do not separate RSC constraints at integer solutions given that the separation of CIs and ECIs at integer solutions is faster than the separation of RSC constraints.

3.5.4 Preprocessing Phase

One way to speedup the convergence of the branch-and-cut algorithm is to reduce the size of ILP1. To do so, we propose a preprocessing step capable of eliminating assignment variables that are strictly zero in any optimal solution of the problem. If customer $i \in I$ is the only customer assigned to facility $j \in V_1$ operating at level $l \in L$ in period $t \in T$, we have $\Lambda_{jt} = \lambda_{it}$ and $\Omega_{jt} = \mu^l_{jt}$. If $\mathcal{P}_{it}(DT_{it}) < \theta_{it}$, then variable $x_{ijlt} = 0$ in any optimal solution of ILP1. For every combination of $i \in I, j \in V_1, l \in L, t \in T$, we check this condition at the beginning of our algorithm.

3.5.5 Perturbing the Problem to Compute Initial Upper Bounds

We present a procedure to generate initial upper bounds to be used as an input to the branch-and-cut algorithm at the root node. Observe that the PDF of the random variable W_{jt} depends on the difference between facility capacity (μ^l_{jt}) and total demand assigned to it (Λ_{jt}). If an integer solution leads to assignment decisions such that total demand assigned at a facility is near to the facility's capacity, chances of that integer solution violating probabilistic constraints increases. One of the negative aspects of this phenomena is that our algorithm may take a substantial amount of time (and branching) before a good feasible solution is found. Our proposed heuristic is based on the idea of solving a perturbed version of ILP1 in which the service level of each facility are reduced by a factor $SF \in (0, 1)$. The perturbed ILP1 can be stated as follows:

$$(\text{ILP1}_{\mathbf{R}}) \quad \text{minimize} \quad \sum_{j \in V_1} \sum_{l_1 \in L} \sum_{l_2 \in L} \sum_{t \in T} f_{jl_1l_2t} z_{jl_1l_2t} + \sum_{i \in I} \sum_{j \in V_1} \sum_{l \in L} \sum_{t \in T} c_{ijlt} \lambda_{it} x_{ijlt}$$

$$\text{subject to} \quad (3.4), (3.6), (3.7), (3.9), (3.10), (3.11), (3.17), (3.18), (3.19)$$

$$\sum_{i \in I} \lambda_{it} x_{ijlt} \leq SF \times \mu_{jt}^{l} \sum_{l_1 \in L} z_{jl_1llt} \quad \forall j \in V_1, \forall l \in L, \forall t \in T. \qquad (3.26)$$

$\text{ILP1}_{\mathbf{R}}$ is solved using a branch-and-cut algorithm in which cover inequalities are separated only at integer points. Constraints (3.26) prevent algorithm from going to solution spaces in which many integer solutions would have later been rejected. Thus, preliminary experiments showed $\text{ILP1}_{\mathbf{R}}$ provides much more quickly a good feasible solution as compared to solving ILP1. Evidently, the quality of the feasible solution improves as the value of SF increases. However, the computational time for solving $\text{ILP1}_{\mathbf{R}}$ also increases.

3.5.6 Local Search Heuristic

In our implementation of a local search heuristic, we use two types of neighborhood structures: *Swap* and *Shift*. In *Swap* neighborhood, we take a pair of two demand nodes $(i, i' : i \neq i')$ of the incumbent solution $(\hat{z}, \hat{x})$, and exchange their assignment to create a new solution $(\hat{z}, \hat{x}')$. In the new solution, we check feasibility of constraints (3.5) and (3.8). If solution $(\hat{z}, \hat{x}')$ satisfies these two constraints, and the objective function value $v(\hat{z}, \hat{x}')$ is lower than that of $v(\hat{z}, \hat{x})$, then $(\hat{z}, \hat{x}')$ becomes the incumbent. We use a first improvement strategy to explore the neighborhoods in sequential order. We start with swapping the assignments of first and second demand nodes until we obtain the first improvement. Similarly, the shift neighborhood is applied to each demand node sequentially until the first improved solution is found. The local search is implemented using a heuristic callback function. The heuristic function is called after every 50 nodes. The pseudo code of local search heuristic is provided in Section B.7 in the Appendix.

3.5.7 A Branch-and-Cut Algorithm

Our exact algorithm uses features from both reformulations. ILP1 has an exponential number of constraints that can be efficiently handled via a cutting plane algorithm. We use this formulation to check the validity of solution vector $(\bar{x}, \bar{z})$ at some nodes of the enumeration tree. In the case of ILP2, although inequalities (3.12) are not exponential in size and can be identified *apriori*, they are still large in number $(|I| \times |V_1| \times |L| \times |T|)$ to be explicitly included in the formulation. For this reason, we add them only if violated at some nodes of the enumeration tree. For every $i \in I, j \in V_1, l \in L, t \in T$, we approximate Δ_{ijlt} such that $\mathcal{P}_{it} \in [(\theta_{it} - \epsilon), \theta_{it}]$, where $\epsilon \in (0, 1)$.

We use cutcallback functions of CPLEX to separate these inequalities. We begin the enumeration tree with constraints (3.4)–(3.7), (3.9), (3.10), and (3.17)–(3.19), and CIs (3.11), ECIs (3.13), and RSC inequalities (3.12) are added on the fly at some nodes of the enumeration tree during the search process. In order to find these inequalities, we use *lazycutcallback* (*LAZYC*) and *usercutcallback* (*USERC*) functions. The *LAZYC* function is invoked at integer nodes and a cover inequality is added if it is violated by the current integer solution. The *USERC* takes the fractional solution of the current node and (heuristically) solves the separation problem described in Section 3.5.2 and 3.5.3 to generate violated inequalities. In the next section, we compare the performance of three variants of our branch-and-cut algorithm: B&C-B, B&C-R1, and B&C-R2. These variants differ from each other with respect to the types of inequalities included and when these are separated.

Finally, we do not include inequalities presented in Proposition 4 in any of the above versions. During our preliminary computational experiments, we noted that adding these inequalities has, on average, a negative impact on both the final optimality gap and computational times.

3.6 Computational Experiments

We conduct extensive computational experiments to assess the performance of the different variants of our branch-and-cut algorithm. All versions of the algorithm were coded in C and run on an Intel Xeon E5-2687W v3 processor at 3.10 GHz in a Linux environment. The algorithms were implemented using the Callable Library of CPLEX 12.9. To compute the probabilities of the service level constraints, we used the algorithm described in Hu et al. [65] and the GSL scientific library. In the subsequent sections, we present the summary of the results of our experiments, and sensitivity analyses to understand the impact of service level constraints and different values of input parameters on the structure of optimal solutions.

3.6.1 Test Instances

Our test bed comprises 400 problem instances (i.e., 5 scenarios $\times$ 4 classes $\times$ 20 instances) categorized into four classes: A, B, C and D, based on the source of potential facility locations (*Amazon* and *US Counties*), and type of the model (*Static-Single* and *Dynamic-Modular capacities*). We refer to Table B.1 in the Appendix for additional details. The potential locations in *Amazon* instances (classes A and B) are derived from actual locations of Amazon fulfillment centers, located in 111 unique counties in the US. The potential locations in *US Counties* instances (classes C and D) are obtained by evenly dividing 3,109 US counties into 9 regions based on the maximum and minimum latitude/longitude. From each region, we randomly select equal number of counties as potential locations (Ortiz-Astorquiza et al. [100]). Each class consists of 20 instances categorized into four to five sets based on the number of potential facilities and demand nodes. Table B.1 also provides the number of potential facilities ($|V_1|$) and demand nodes ($|I|$) for various instances in each class and set. For example, class A set I has five instances with 30 most populous counties in *Amazon* data as locations for potential facilities and $\{125, 250, 375, 500, 625\}$ most populous counties, respectively, from 3,109 US counties as aggregated demand nodes. Further, we vary the parameters τ, θ, CV, and κ (parameter to control service capacity) to generate various scenarios for each class as follows: *Base case (Scenario 1)*: $\tau = 1.5$, $\theta = 0.90$, $CV = 0.5$, $\kappa = 10$; *High response time (Scenario 2)*: $\tau = 2.5$, $\theta = 0.90$, $CV = 0.5$, $\kappa = 10$; *High service probability (Scenario 3)*: $\tau = 1.5$, $\theta = 0.95$, $CV = 0.5$, $\kappa = 10$; *High travel time variability (Scenario 4)*: $\tau = 1.5$, $\theta = 0.90$, $CV = 0.8$, $\kappa = 10$; *Tight capacity (Scenario 5)*: $\tau = 1.5$, $\theta = 0.90$, $CV = 0.5$, $\kappa = 5$. The remainder of the parameters are detailed in Section B.8 in the Appendix. Also, in Section B.9 in the Appendix, we present few CPLEX and separation algorithm parameters that are fine tuned to improve the convergence of the solution method.

3.6.2 Performance of Branch-and-Cut Algorithms

We report the performance of three variants of our branch-and-cut algorithm. The first variant, B&C-B uses separation routine at integer solutions for CIs (3.11), whereas the second variant, B&C-R1 employs separation routine at both integer and fractional solutions for CIs (3.11) and the initial and local search heuristics described in Sections 3.5.5 and in Appendix 3.5.6, respectively. The third variant, B&C-R2 includes initial and local search heuristics. Additionally, in separation routines at fractional solutions, we first add inequalities (3.12) and subsequently (3.13), provided (3.12) are not violated. At integer solutions, we add ECI_1 inequalities (3.13). Note that all three variants of the algorithm also include preprocessing (Section 3.5.4), and valid inequalities (3.17), (3.18) and (3.19). For every problem instance, the execution time limit was set to 86,400 secs (24 hours). The results are presented in Table 3.1 for sets I and II (small size instances) and in Table 3.2 for sets III, IV and V (large size instances). We summarize the results over 381 (out of 400) instances in which every variant of the algorithm provided a feasible solution. The tables report the average optimality gap, the average computation times and the number of instances solved to optimality. In these tables, the column B/R1/R2 reports the number of instances on which one variant of the algorithm outperforms the others. For example, the first entry reports the number of instances on which B&C-B algorithm outperformed others out of a total of (B+R1+R2) instances. Similarly, the second and third entries report the number of instances on which B&C-R1 and B&C-R2 algorithms outperformed other variants.

Table 3.1: Performance of Variants of Branch-and-Cut Algorithm for Sets I and II Instances (Small Size)

		Average Gap (%)			Average Time (s)			No. of Opt. Instances			
Scenario	Class	B&C-B	B&C-R1	B&C-R2	B&C-B	B&C-R1	B&C-R2	B&C-B	B&C-R1	B&C-R2	B / R1 / R2
1	A	0.17	0.17	0.06	43,317	43,415	21,166	5	5	8	3 / - / 7
	B	0.16	0.14	0.03	53,170	42,699	27,758	6	6	8	2 / 2 / 6
	C	0.01	0.01	0.01	20,644	22,426	11,703	7	6	7	4 / 1 / 3
	D	0.08	0.10	0.07	57,935	49,532	41,994	5	7	7	2 / 1 / 7
2	A	0.01	0.01	0.01	18,717	2,594	1,403	8	10	10	4 / 1 / 5
	B	0.02	0.01	0.01	42,098	28,697	20,356	6	7	9	4 / 2 / 4
	C	0.01	0.01	0.01	11,233	12,151	3,002	7	7	8	7 / - / 1
	D	0.04	0.10	0.11	55,084	48,828	47,744	4	7	6	4 / 5 / 1
3	A	0.08	0.25	0.01	27,364	17,001	14,833	5	5	5	3 / - / 3
	B	1.03	0.62	0.31	68,377	59,879	53,844	3	4	5	1 / - / 9
	C	0.03	0.02	0.01	44,342	36,488	7,544	4	5	8	2 / 1 / 5
	D	1.07	0.72	0.51	64,375	57,589	49,984	4	4	6	2 / 2 / 6
4	A	0.02	0.01	0.01	21,668	21,733	692	3	3	4	2 / 1 / 1
	B	1.10	0.19	0.12	60,921	51,174	37,132	3	5	6	- / - / 10
	C	0.19	0.17	0.01	41,818	37,357	12,535	5	5	7	2 / 1 / 5
	D	0.87	0.41	0.16	64,513	57,033	47,086	4	4	6	2 / - / 8
5	A	0.26	0.27	0.09	60,487	52,042	24,508	3	4	8	3 / 1 / 6
	B	0.61	0.19	0.01	60,700	50,539	32,462	3	5	9	2 / - / 8
	C	0.10	0.08	0.13	54,071	44,052	23,520	3	4	6	1 / 2 / 5
	D	0.83	0.27	0.28	73,074	66,723	66,073	2	3	3	2 / 1 / 7
Average		0.37	0.20	0.11	49,197	41,756	29,296	90	106	136	52 / 21 / 107

Table 3.2: Performance of Variants of Branch-and-Cut Algorithm for Sets III, IV and V Instances (Large Size)

		Average Gap (%)			Average Time (s)			No. of Opt. Instances			
Scenario	Class	B&C-B	B&C-R1	B&C-R2	B&C-B	B&C-R1	B&C-R2	B&C-B	B&C-R1	B&C-R2	B / R1 / R2
1	A	2.72	0.51	0.23	86,400	78,843	60,406	0	1	4	- / 2 / 8
	B	2.36	0.81	0.77	86,400	78,268	78,378	0	2	2	1 / 4 / 5
	C	1.24	0.94	0.85	86,400	81,451	79,259	0	1	2	1 / 6 / 5
	D	2.17	1.14	1.40	86,400	86,400	82,468	0	0	1	2 / 6 / 2
2	A	0.68	0.27	0.22	65,738	62,129	65,785	4	4	5	- / 5 / 5
	B	0.85	0.35	0.36	68,786	70,135	70,664	3	3	3	2 / 5 / 3
	C	0.57	0.75	0.69	85,016	79,593	80,697	1	2	2	6 / 3 / 3
	D	1.10	0.73	0.70	69,949	76,810	78,555	4	3	3	4 / 2 / 4
3	A	2.88	0.88	0.81	86,400	86,400	86,400	0	0	0	- / 2 / 7
	B	3.56	1.28	0.81	82,439	78,650	67,876	1	1	3	- / 1 / 8
	C	2.33	1.47	1.34	86,400	86,400	82,476	0	0	2	2 / 3 / 7
	D	3.41	1.35	1.27	86,400	86,400	86,400	0	0	0	- / 2 / 7
4	A	3.21	0.53	0.39	86,400	86,400	80,879	0	0	2	1 / - / 9
	B	3.33	1.34	1.32	71,796	68,835	68,174	1	1	1	1 / 1 / 2
	C	2.22	1.26	1.12	86,400	86,400	80,303	0	0	2	2 / 3 / 7
	D	2.74	1.67	1.88	82,962	83,357	85,311	1	1	1	2 / 8 / -
5	A	2.02	0.88	0.63	86,400	86,400	82,960	0	0	2	1 / 4 / 5
	B	3.78	2.09	2.81	86,400	86,400	86,400	0	0	0	1 / 5 / 4
	C	1.65	0.84	0.86	86,400	86,400	86,400	0	0	0	3 / 6 / 3
	D	2.16	1.04	1.24	86,400	86,400	86,400	0	0	0	1 / 7 / 2
Average		2.17	1.00	0.97	82,857	81,523	79,244	15	19	35	30 / 75 / 96

The selected feature configuration of each of these variants allows us to evaluate the usefulness of the proposed valid inequalities and the most important algorithmic refinements of our branch-and-cut algorithm. The results show that B&C-R1 solves 16 more instances for small-size and four for large-size to optimality over B&C-B. It also reduces optimality gap by 0.17% and 1.17% for small- and large-size instances, respectively. When compared to B&C-B, the average CPU time in B&C-R1 is 15% lower for small-size instances, and it outperforms B&C-B in 45 more large-size instances. It seems that B&C-R1 improves the overall performance of the basic version B&C-B.

We next asses the incremental benefits of including ECIs and RSC constraints in B&C-R2 over B&C-R1. We observe that B&C-R2 reduces on average CPU times by nearly 30% in Set I and II instances, particularly, in 4-A and 3-C where reductions are 97% and 79%, respectively. Over all, B&C-R2 solves 46 instances more to optimality and provides better result in 107 (86 small-size and 21 large-size) more instances over B&C-R1.

From Tables B.2 and B.3 given in the Appendix, we observe the average number of added CIs are maximum for B&C-B. As we start separating CIs at fractional solutions, and providing initial upper bounds, the number of CIs in B&C-R1 decrease to nearly one-half and one-third of the inequalities required in B&C-B for small- and large-size instances, respectively. This shows that B&C-R1 is able to reject many infeasible solutions. Also, a good quality initial upper bound deemed many nodes infeasible. We observe a similar behavior when ECIs and RSCs are included in B&C-R2 algorithm. When considering a low value of ϵ (e.g., 0.001), the RSC inequalities provide a very good approximation, and ECIs (being stronger than CIs) are helpful in eliminating many solutions at early stages of the enumeration tree.

3.6.2.1 Effect of varying service level requirements.

We next analyze the impact of varying service level requirements on the network configurations. We consider a static version with a single capacity level. The set of potential facilities and customer nodes obtained from the states of US: Washington, Oregon, California, Nevada, Idaho, Utah, Arizona, New Mexico, and Texas. We randomly select two counties from each state as potential facility locations, and every county in these states correspond to customer nodes. The changes in network configuration by varying the parameters are analyzed using the following three scenarios.

We first vary the minimum acceptable probability θ while τ and CV are fixed to 1.5 and 0.75, respectively. The effect on location decisions and its impact on total cost with respect to the solution without any service level requirements (base case) are depicted in Figure 3.2 and Figure 3.3, respectively. The change in colors in the US map represents the change in location decisions from one scenario to another, and all counties that are of same color are assigned to the same facility. In the base case, the solution opens two facilities, one in Carson City, NV and another in Real, TX (Figure 3.1). By increasing the minimum acceptable probability, the problem becomes more constrained which increase the number of opened facilities thereby increasing the total costs. For instance, when $\theta = 0.9$, three facilities are now opened, the location of facilities changes to Humboldt, NV, and McKinley, NM, the number of counties allocated to the facility in Real, TX decreases (Figure 3.2a). This increases the network cost by 4%. When θ is further set to 0.95, a total of five facilities are opened, shipping distance decreases, and the cost required to operate this network is 9% higher compared to the base case. Upon increasing θ to 0.975, a total of six facilities are opened, shipping distance decreases substantially, and the corresponding network cost is 22% higher from base case (Figure 3.2c). A service probability of 0.99 can be achieved at the expense of 40% increase in total network cost. Therefore, from the left portion of the plot in Figure 3.3 we observe that a significant increase in service level can be achieved with a small increase in total cost (e.g. varying $\theta = 0.80$ to 0.925 results in 4% increase). However, after a certain point ($\theta = 0.975$ and above), substantial investments are required for higher service level guarantees.

We next vary the committed delivery time τ while θ and CV are fixed to 0.9 and 0.75, respectively. The impact of changing τ on location-allocation decisions is shown in Figure 3.4. In the base case, there is no restriction on delivery time, hence the model recommends shipping to distant customer nodes. For instance, in Figure 3.1, the facility in Real, TX is

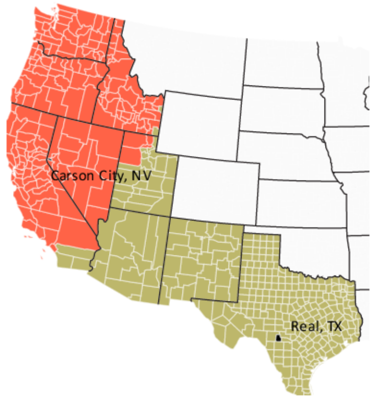
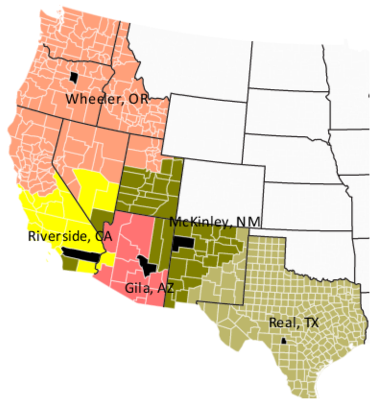
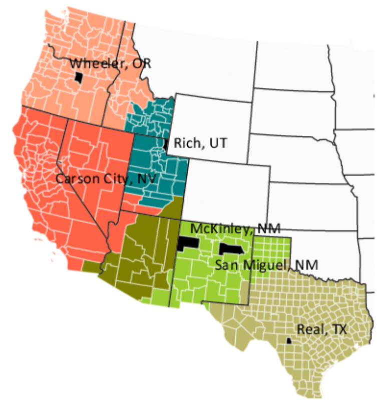


Figure 3.1: Base case: Without service level constraints

(a) θ=0.90



(b) θ=0.95



(c) θ=0.975

Figure 3.2: Effect of varying minimum acceptable probability (θ) on network configuration (τ=1.5 and CV=0.75)

shipping to counties at the edge of the state of California. However, as we commit two days of delivery time (τ = 2.0), those longer shipping distances are no longer valid, and model prescribes closing the facility at Carson city, NV and opening one in Humboldt, NV. The allocation decisions changes and the shipping distances are comparatively shorter as shown in Figure 3.4a. A similar behavior is observed as τ is decreased to 1.5 units, and when τ is set to 1, the service level requirements are so stringent that the model opens six facilities (Figure 3.4c) to serve every customer node within 1 day of response time. Figure 3.5 shows the effect of varying the committed delivery time on the change in total cost. We observe that response time can be reduced by 1 day with only 4% increase in network cost (from τ = 2.50 to 1.5 days). However, further reduction in delivery time can be achieved with substantial increase in total cost (e.g. 22% increase for τ = 1.0).

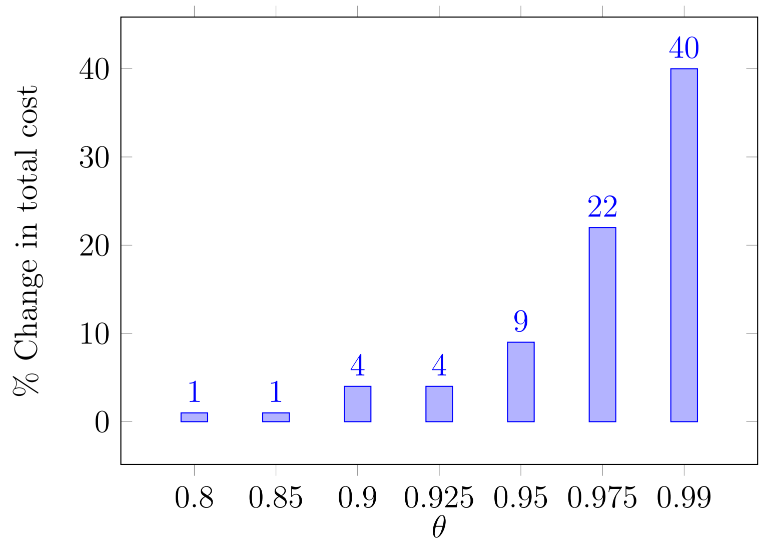


Figure 3.3: Percentage increase in total cost from no service level scenario to the scenarios with different value of θ (τ=1.5, CV=0.75)

Finally, we vary the coefficient of variation of travel times CV whereas θ and τ are held constant at 0.9 and 1.5, respectively. An increase in CV leads to a change in location and allocation decisions (Figure 3.6), and an increase in total cost (Figure 3.7). As we introduce service level constraints with $CV = 0.5$, the model changes the location of facility by closing the facility at Carson city, NV and opening one in Humboldt, NV (Figure 3.6a). As we further increase CV to 1.0, the model recommends opening another facility in McKinley, MN (Figure 3.6b). When CV to 1.5, the model opens two facilities, one in Wheeler, OR and the other in Carson City, NV while closing the Humboldt, NV facility (Figure 3.6c). An increase in the value of CV increases the uncertainty in the travel time, therefore model opens more facility for service guarantee, which increases the network cost, as shown in Figure 3.7. The left portion of this plot indicates that a small variability in travel time results in a small increase in network cost. However, increase in variability ($CV = 1.25$ and above) results in a substantial increase in total cost.

(a) τ=2.0 (b) τ=1.5 (c) τ=1.0

Figure 3.4: Effect of varying threshold of delivery time (τ) on network configuration (θ=0.90 and CV=0.75)

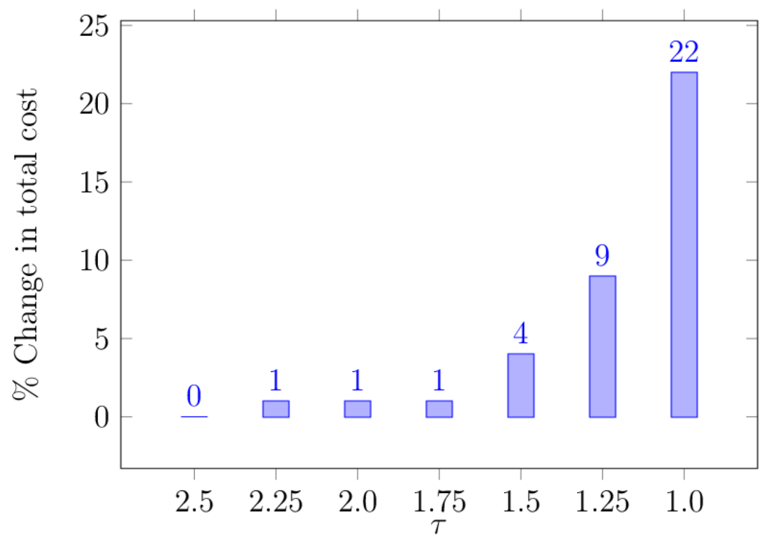


Figure 3.5: Percentage increase in total cost from no Service level scenario to scenarios with different value of τ (θ=0.90 and CV=0.75)

3.6.2.2 Effect of varying levels of stochasticity in service level constraints.

We analyze the impact of increasing variability in processing times at the facilities and travel times on the links. We consider one of the problem instances in class C, and set: θ=0.90. τ=1.0, and CV = 0.25. The configuration in Figure 3.8a represents the scenario without any service level constraints, whereas Figure 3.8b corresponds to the scenario with stochastic processing times only. Figure 3.8c depicts the scenario with stochastic travel times

(a) CV=0.5 (b) CV=1.0 (c) CV=1.5

Figure 3.6: Effect of varying coefficient of variation of travel times (CV) on network configuration (θ=0.90, τ=1.5)

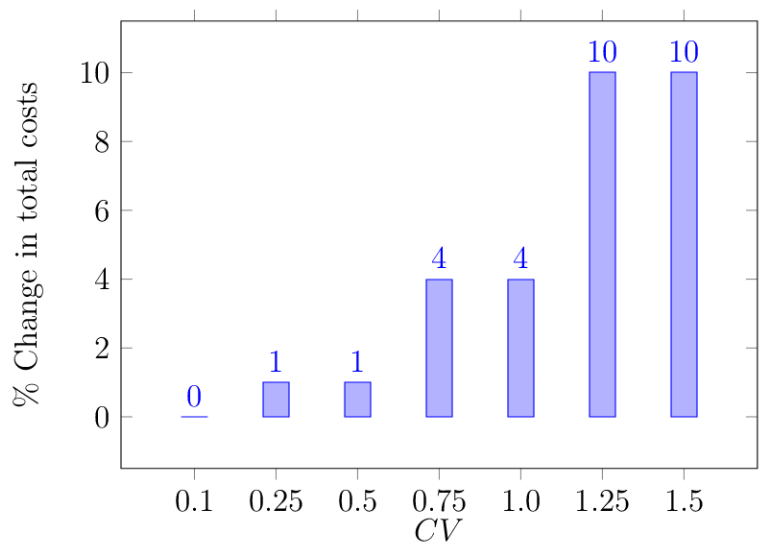
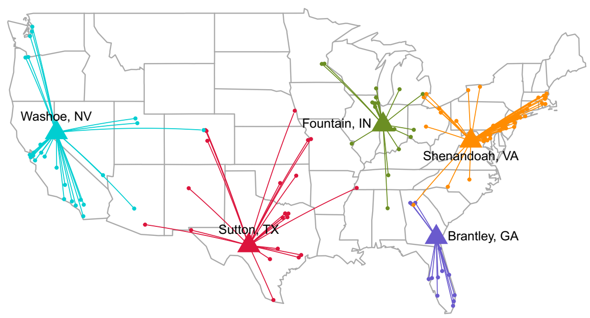


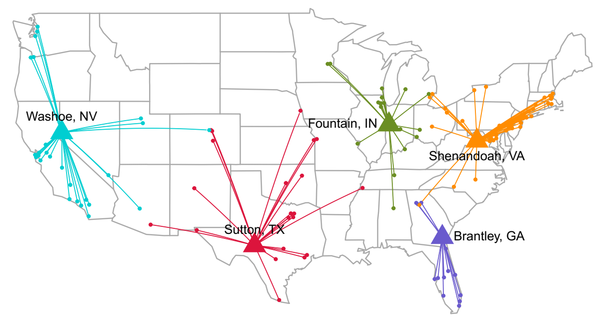
Figure 3.7: Percentage increase in total cost from no service level scenario to scenarios with different value of CV (θ=0.90 and τ=1.5)

only, whereas the scenario with both stochastic processing and travel times is depicted in Figure 3.8d. There figures show the impact of increasing variability in processing times and travel times on location and allocation decisions. Note that the locations of three facilities, i.e. Washoe (NV), Shenandoan (VA) and Brantley (GA) remains unchanged across these scenarios, however, their allocation decisions change. The customers in Colorado and Arizona that were served by Washoe (NV) before are now served by Dona Ana (NM) in Figure 3.8d. Similarly, one of the customers in Georgia served by Shenandoan (VA) is now assigned to

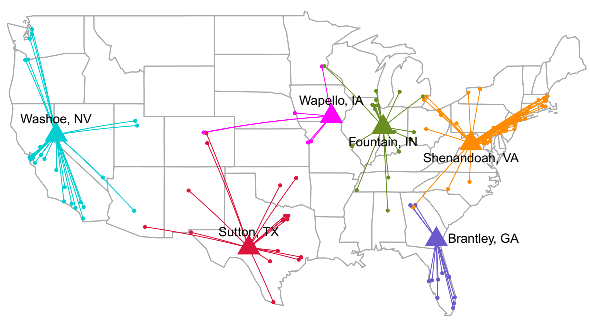
Osage (MO). The variability in processing times and travel times has led facilities in Sutton, TX and Fountain, IN to move to Dona Ana, NM and Osage, MO, respectively in the last scenario thereby increasing the total cost by 5.7% .



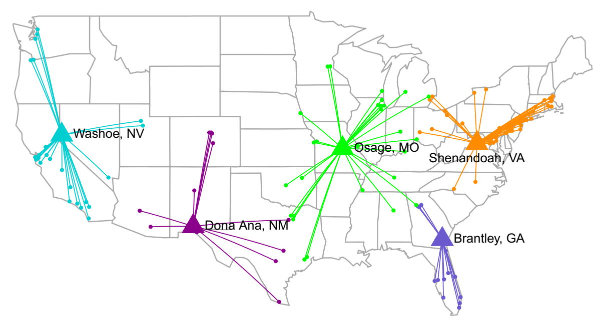
(a) Without Service Level Constraints



(b) With Stochastic Processing Times (↑ 0%)



(c) With Stochastic Travel Times (↑ 1.6%)



(d) With Stochastic Processing and Travel times (↑ 5.7%)

Figure 3.8: Network configurations with different service level constraints (θ=0.90. τ=1.0, and CV = 0.25)

Finally, Section B.11 in the Appendix provides a sensitivity analysis on the impact in network configuration when varying demand in a dynamic setting.

3.7 Conclusion

We studied a general class of dynamic and modular capacitated facility location problems with probabilistic service level constraints on delivery times. These constraints ensure that every customer is served within τ units of times after an order is placed with a minimum acceptable probability of θ. The total time to serve a customer is dependent on stochastic order processing time at facilities as well as the stochastic travel time from facilities to customers. We provided two integer linear reformulations and proposed an exact branch-and-cut algorithm along with several algorithmic refinements to improve its convergence. Instance with up to 225 facilities and 2,500 customers for static and single capacity variants, and 54 facilities and 700 customers for dynamic and multiple capacity variants were solved efficiently. To the best of our knowledge, these instances are by far the largest ever solved for any type of stochastic capacitated facility location problem with probabilistic service level constraints.

The proposed polyhedral representations of probabilistic service level constraints can be used to develop tractable representations of other challenging classes of stochastic problems with service level constraints, as long as the steady-state distribution of delivery times (involving stochastic wait, service and travel times) can be evaluated.

Chapter 4

Demand Allocation, Inventory Positioning and Distribution Network Design in Omnichannel Retailing

Abstract

With the emergence of e-commerce and the growing trend of online shopping, many retailers are adding online sales channel to their traditional brick-and-mortar stores. In this paper, we study the problem of demand allocation, inventory positioning, and distribution network design for a retailer that seeks to integrate online sales channel into their traditional brick-and-mortar retail channel. The proposed analytical model seeks to locate distribution centers, select stores for online demand fulfillment, assign stores to fulfillment centers, and allocate safety stocks at distribution centers and stores to deal with variability in demands and lead time. Using the model, we show the benefits of integrating distribution centers in the omnichannel network to serve in-store demand as well as online demand. The model also determines if the safety stock inventory should be pooled at the distribution centers (i.e. centralized inventory) or held at the stores (i.e. decentralized inventory). We present a

second-order cone mixed integer programming formulation and an exact branch-and-cut algorithm, based on this formulation,in which extended polymatroid inequalities are separated. Extensive computations under different cost scenarios and parameter settings confirm the efficiency of our exact algorithm. We conduct sensitivity analyses to understand the impact of variation in cost and model parameters and present managerial insights.

4.1 Introduction

The retail industry has seen a phenomenal growth in recent years with the advances in mobile technologies and digital disruptions, accompanied by increasing consumer expectations of convenience. In 2020, retail e-commerce sales worldwide amounted to 4.28 trillion US dollars and e-retail revenues are projected to grow to 5.4 trillion US dollars in 2022 (Statista, 2021). The outbreak of the COVID-19 pandemic and the closure of brick-and-mortar stores has fueled the surge in retail e-commerce. Factors such as ease of shopping, wider range of products, quick home delivery of orders, (e.g., within hours, or one to two days of placing the order) have motivated customers to shop online. Retailers are facing mounting pressures as a result of the explosion of smartphone-driven shopping, the concept of physical showrooms, and the popularity of free returns with no-questions asked among others. Many retailers are responding to this pressure by adapting to the new paradigm of "omnichannel retailing", which seeks to fully integrate the traditional brick-and-mortar stores with several online channels seamlessly.

While e-commerce is gaining popularity, in-store shopping is also desirable, as many customers like to physically interact with the product before buying, the products are available immediately, the returns are easy and hassle free, and there are no shipping costs involved. Thus, to capture the best of both, brick-and-mortar (B&M) retailers have started to introduce online sales channel (websites, mobile apps etc.) along with their traditional in-store options. For example, Walmart launched its e-commerce channel in the year 2011 (Walmart [125]). In the third quarter of 2020, Walmart's online sales in the US increased by 79% making it the second highest e-commerce retailer in 2020 (IndigoDigital [70]). Similarly, Target committed nearly $7 billion in the year 2017 to improve customer's digital shopping experience (Target [117]). At the same time, pure e-commerce retailers are also in the process of establishing physical stores to serve their customers (Verhoef et al. [121]).

To fulfill the demand arising from different channels, retailers use an omnichannel distribution strategy. Under that strategy, stores are no longer only a last point of sales (in-store demand), but also function as fulfillment centers (FCs), wherein store's inventory is used to meet online demand in several ways. For instance, customers can buy online and pick up their orders at store/curb-side (BOPS); customers also have an option of home delivery of their orders which can originate from the stores. Stores also provide 'showroom' experience, where customers physically experience the product in the store and have the items (ordered online or in the store) delivered to their preferred location. Additionally, traditional distribution centers and warehouses are being used as drop shipping facilities from which online orders can be directly shipped to the customers.

For the ease of operations, many retailers plan logistics activities for in-store demand and online demand using separate distribution centers, which results in higher inventory levels and cost of operations (Hübner et al. [69]). Retailers that attempt deliveries of the online orders using their current distribution network are faced with delayed deliveries, particularly in case of orders with same day and one-day delivery commitments (Arslan et al. [9]). Omnichannel strategy can be an efficient approach to facilitate sales and distributions through various channels and to improve customer satisfaction. Stores play an integral role in online demand fulfillment under omnichannel strategy. To use stores as fulfillment centers, retailers need extra floor space and extra workers, and new contracts with 3PLs for faster deliveries. Thus, such integrated approach requires significant investments affecting the profitability of the retailer. Another common issue with omnichannel distribution revolves around imbalanced inventory at facilities which leads to stockouts at some facilities and/or excess inventory levels at others. A survey of 350 major retailers, conducted by JDA and PwC in 2017, points out that only 10% of the retailers managed to run their omnichannel network profitably (Businesswire [21]).

Motivated by the growing need for integrated design of distribution networks for omnichannel retailing, in this paper, we focus on the following research questions that would assist retailers in building an efficient omnichannel distribution network: (i) What are the benefits of integrating distribution centers in the omnichannel network to serve in-store as well as online demand? (ii) How would the configuration of the logistics network change if we use stores as well as distribution centers in the distribution planning of omnichannel retail chains? Should the online demand be allocated to (and served from) fulfillment centers, or retail stores, or both? (iii) Where should the inventory be positioned in the logistics network? Should we pool the safety stock inventory at the distribution centers (i.e. centralized inventory policy) or should we hold the safety stock inventory at the stores (i.e. decentralized inventory policy)?

In this paper, we consider a distribution network with three sets of nodes: customer zones, B&M stores, and fulfillment centers (FCs). Each customer zone has online demand orders for multiple products with home delivery requirements. The B&M store locations are already established and operational, and the model selects stores from this set for expansion to keep inventory for fulfilling online demand. However, expansion of store incurs fixed cost for creating additional space and hiring extra labor. In our problem, however, the objective is to determine the location of FCs. Our model also provides flexibility to incorporate scenarios where FCs are already established, and a retailer is looking to use existing FCs or open additional FCs. We model two fulfillment strategies to satisfy online demand, namely, ship-from-FC (SFFC) and ship-from-store (SFS), whereas a store is always assigned to FC. There is no transshipment of products between stores. We also determine safety stock inventory at selected stores and FCs based on the demand assigned to these facilities.

Potential application of our proposed model include the design of omnichannel distribution network for retailers in general merchandise, consumer electronics, and fashion industries, among others. These retailers frequently use stores for home delivery of online orders. For example, Best Buy started shipment of online orders from stores in 2014. In the third quarter of 2020, Best Buy used 250 stores as fulfillment hubs, and had plans to add another 90 stores to this list in the fourth quarter (SupplychainDive [116]). The primary reason for such a strategy is the already well established network of Best Buy stores that are within 15 minutes drive of 70% of US population (Forbes [48]). Our model also determines which stores to use as fulfillment hubs from an existing set of stores. Additionally, in 2019, Best Buy added three new *metro e-commerce centers* (CNBC [27]): Compton, California (a suburb of Los Angeles), Chicago and Piscataway, New Jersey. The committed time to deliver online orders is an important aspect customers keep in mind while shopping online. For Best Buy, the delivery of majority of its orders matches Amazon Prime's two-days delivery commitment (Radial [104]). To ensure such service guarantees, we include service distance restrictions in our model, which stipulate that an online order can only be served from a facility (FC / store) located within a specified distance from the customer.

The main contribution of this paper is outlined as follows:

- Modeling contribution: We present an analytical model that captures the trade-off among logistics cost (i.e. facility fixed costs, transportation costs, inventory holding costs), safety stock inventory (risk pooling), and service time commitments arising in the design of distribution networks for an onmnichannel retail environment.

- Methodological contribution: We formulate the model as a *second-order cone mixed integer program* (CQMIP). We present a branch-and-cut algorithm in which this formulation is strengthened with polymatroid inequalities. These inequalities are heuristically separated at fractional nodes of the enumeration tree. We also conduct extensive computational experiments under different cost scenarios and parameter settings to test the efficiency of our solution algorithm. In particular, we generate instances to capture scenarios with balanced cost, dominant fixed cost, dominant transportation cost, dominant safety stock cost, service distance restriction, tight capacities and loose capacities of the facilities (FCs and stores).

- Case study contribution: The applicability of the model and solution approach is illustrated using the case study of Best Buy, a major electronics retailer in USA. We use the thirteen actual locations of Best Buy as B&M stores and 50 zip codes from the city of New York, Newark, and Jersey City as customer zones. Through the case, we show the benefits of an integrated distribution channel over dedicated distribution channels. We also analyze the benefits of partially integrated distribution channels (i.e. online demand fulfilled either from FCs or store) over fully integrated distribution channels (i.e. online demand can be fulfilled from FCs and/or store). We conduct a series of sensitivity analyses to understand the impact of variation in model parameters. We also present sensitivity analyses of the model to parameters such as service distance restrictions and inventory holding costs.

The remainder of the paper is organized as follows. Section 4.2 presents a brief review of literature relevant to our work, followed by a formal definition and mathematical model in Section 4.3. Section 4.4 provides details on solution method, and the experimentation and computational analysis of the methodology are presented in Section 4.5. In Section 4.6, we present case study and managerial insights, followed by conclusions in Section 4.7.

4.2 Literature Review

The growing trend and impact of adopting an omnichannel strategy on operations has received significant attention in the academic literature. Rigby [105] emphasizes the effect of digital revolution on retailing and the need of adopting omnichannel strategy. Brynjolfsson et al. [19] provide new strategies for successful omnichannel environment such as pricing decision, emphasize on building interactive shopping experience, and building customer relationships. Bell et al. [14] provide four categories of retail environment: traditional retail (in-store sales), showroom based retailers with home delivery, pure e-commerce retailers, and retailers who sell online and fulfill demand from stores. More recently, Chopra [25] identifies the challenges that omnichannel retailing poses and opportunities it can provide. The author also discusses the relative variation in various supply chain costs based on the channels adopted, and presents a framework to select appropriate channel based on product portfolio and customer characteristics.

The literature on omnichannel retailing is broadly classified into two streams: empirical studies and modeling of operations. Ansari et al. [8] develop an empirical model that evaluates the impact of web-based channels on long term demand, and identifies the factors that affect the migration of customers to these channels. Gallino and Moreno [52] empirically analyze the impact of buy online, pick-up in the store option (BOPS). Their analysis show that BOPS increases store traffic, and hence, in-store sales. This is due to cross-selling i.e. customers buy additional products when visit the stores, and due to channel-shift effect converting non-store customers to store patrons. Gao and Su [53] further provides evidence for channel-shift effect in BOPS option, and shows that this option is not profitable for in-store fast selling items. Hübner et al. [67] conducted an exploratory study to identify the distribution practices in omnichannel retailing. Hübner et al. [68] analyze empirical data and conducted personal interviews to provide a framework for last mile delivery in grocery retail business. Hübner et al. [69] surveyed 24 German retailers to show that BOPS channel is particularly useful in high density areas.

From prescriptive modeling point of view, prevalent research questions in operations of omnichannel practice are from the perspective of pricing, returns, inventory management, and fulfillment. Based on different product and customer characteristics, Cao et al. [23] present an analytical framework to determine the products to sell online, their pricing, and demand allocation. Harsha et al. [63] solve a dynamic price optimization problem and partition inventories for in-store and online demand while maximizing profits in deterministic settings and worst-case scenario. Mandal et al. [86] develop a model that suggests appropriate omnichannel environment for different product classes and recommend strategies on managing returns. Gao et al. [54] suggest a model to identify the number and size of the stores, and also study the impact of using stores as showrooms, allowing returns at the stores, and allowing flexible fulfillment from the stores. Park et al. [101] investigate a problem of selecting products to display in the stores (used as showrooms), by including product value and showroom capacity in planning.

Chen and Graves [24] solve an inventory placement problem at fulfillment centers that are already opened, capacitated and there is a fixed cost of placing an item while minimizing the total cost. Saha and Bhattacharya [107] develop various inventory control policies at stores for BOPS option under single and separate ownership (franchise) of physical store and online channel. The impact on inventory levels is also studied by Hu et al. [66], who set forth conditions under which BOPS option leads to pooling or depooling of inventory at stores. Another important aspect of omnichannel strategy is the fulfillment of online demand. Acimovic and Graves [6] develop a heuristic that makes fulfillment decisions based on immediate and future expected outbound shipment costs. Torabi et al. [118] also consider an order fulfillment problem, which allows transshipment of inventories between stores and transfer of orders from one store to another in case of shortages. Govindarajan et al. [57] study a multi-period planning problem in which inventory decisions are made at the beginning of the horizon and fulfillment decisions are made in every time period. Bayram and Cesaret [13] consider separate inventory for online demand orders, and their model decides whether to use DC or a store to fulfill online demand. Yang and Zhang [127] optimize retailer's profit in ship-from-store option by including customer utility function in the model.

The current network of stores play a crucial role in last mile delivery in omnichannel retailing, as showcased by Ishfaq et al. [72]. Some papers have studied the problems that include location decisions, in addition to inventory and allocation decisions. For example, Ishfaq and Bajwa [71] includes binary decision of locating only direct-to-customers distribution centers, along with an option of fulfilling demand from existing stores, DC, and vendors. Their model also assumes linear inventory cost at every supply node and a convex demand function. Arslan et al. [9] include the decision of selecting stores as fulfillment centers and opening FCs with inventory decisions at both locations. Their problem utilizes a scenario based approach to model uncertainty in demand. We refer readers to Melacini et al. [93] and Cai and Lo [22] for detailed reviews on omnichannel strategy, and to Jasin et al. [73] for the review on modeling practices.

This paper focuses on the integrated design of a distribution network to fulfill in-store and online demand while maintaining safety stocks at FCs and stores. In our work, we also include the decisions of opening new FCs, selecting stores for online order fulfillment. However, we assume that demand is normally distributed with a mean and variance, and use basic inventory theory to calculate required safety stocks at stores and FCs.

4.3 Problem Description and Model Formulation

Let I be the set of customer zones, V_1 be the set of existing locations of B&M stores, and V_2 be the set of potential FC locations. We assume that a store $j \in V_1$ is supplied from FC $k \in V_2$, whereas a customer zone $i \in I$ is supplied either from a FC k or a store j. Also, let P be the set of product categories and D be the demand type originated from each customer zone. The demand type represents different service level commitments in terms of order delivery time. For example, in case of Amazon, prime and non-prime customers/orders are two types of demand with different delivery commitments for the same product. The stochastic demand for every product-demand type combination from each customer zone is assumed to follow a normal distribution with known mean and variance. Similarly, in-store stochastic demand for every product is assumed to follow a normal distribution with known mean and variance. To prevent the risk of stockouts during lead time or due to variation in demand, we consider pooling safety stock inventory at stores and at opened FC locations.

We also introduce service distance restrictions to ensure that customer located in zone i with demand type d is assigned to a facility within a specified distance from i. The notations are listed in Table 4.1.

To compute the safety stock at facilities, let us assume for a moment that the assignment of customer zones and stores to FCs are known a priori. We consider a single period model, and assume that demand from a customer zone i, for product p and demand type d is normally distributed, with a mean of μ_{idp} and standard deviation of σ_{idp}. Also assume that the in-store demand for a product p at store j is normally distributed, with a mean of α_{jp} and standard deviation of β_{jp}. From basic inventory theory, we know that if the demands from customer zones are uncorrelated, then the demand of product p during the lead time at FC k is normally distributed with mean $\left(\mathcal{L}_{2k}\sum_j \alpha_{jp} + \mathcal{L}_{2k}\sum_i\sum_{d\in D}\mu_{idp}\right)$ and variance $\left(\mathcal{L}_{2k}\sum_j \beta_{jp}^2 + \mathcal{L}_{2k}\sum_{i\in I}\sum_{d\in D}\sigma_{idp}^2\right)$. The safety stock level for product $p \in P$ at FC $k \in V_2$ is given by

$$z_\theta \sqrt{\mathcal{L}_{2k}\left(\sum_{j\in V_1}\beta_{jp}^2 + \sum_{i\in I}\sum_{d\in D}\sigma_{idp}^2\right)}.$$

Let $I' \subseteq I$ and $V_1' \subseteq V_1$ be the set of customers and the set of stores, respectively that are assigned to an open FC $k \in V_2$ ($z_{2k} = 1$). We further assume that the stores belonging to set V_1' are not selected to fulfill customer zones i.e. $z_{1j} = 0\,\forall j \in V_1'$. In that case, the demand during the lead time at FC k is normally distributed with mean of $\mathcal{L}_{2k}\left(\sum_{j\in V_1'}\alpha_{jp} + \sum_{i\in I'}\sum_{d\in D}\mu_{idp}\right)$ and variance of $\mathcal{L}_{2k}\left(\sum_{j\in V_1'}\beta_{jp}^2 + \sum_{i\in I'}\sum_{d\in D}\sigma_{idp}^2\right)$, hence, the safety stock level at FC k is given by

$$z_\theta \sqrt{\mathcal{L}_{2k}\left(\sum_{j\in V_1'}\beta_{jp}^2 + \sum_{i\in I'}\sum_{d\in D}\sigma_{idp}^2\right)}.$$

If the decisions regarding the assignment of customer zones to stores (v_{kj}), assignment of customer zones to FCs (x_{2kid}), and store selection (z_{1j}) are to be determined endogenously, then the expression for total safety stock levels at FC $k \in V_2$ is given by $\sum_{p \in P} z_\theta r_{2kp}$, where,

$$r_{2kp} = \sqrt{\mathcal{L}_{2k}\Big(\sum_{j \in V_1} \beta_{jp}^2 v_{kj}(1 - z_{1j}) + \sum_{i \in I}\sum_{d \in D} \sigma_{idp}^2 x_{2kid}\Big)} \tag{4.1}$$

and z_θ is the standard score (or z-score) corresponding to the desired service level θ.

Table 4.1: Table of Notations

Indices	
i	Index for customer zones, $i \in I$
j	Index for B & M Stores, $j \in V_1$
k	Index for fullfilment centers (FCs), $k \in V_2$
d	Index for demand types at customer zones, $d \in D$
p	Index for product categories, $p \in P$
Parameters	
F_{1j}	Fixed cost to use store j as a fulfillment center
F_{2k}	Fixed cost of opening a FC at location k
h_{1jp}	per unit inventory holding cost for product p at store j
h_{2kp}	per unit inventory holding cost for product p at FC k
$\mathcal{L}_{1j}$	Lead time at store j
$\mathcal{L}_{2k}$	Lead time at FC k
μ_{idp}, σ_{idp}	Mean and standard deviation of online demand of product p and demand type d at customer zone i
α_{jp}, β_{jp}	Mean and standard deviation of in-store demand for product p at store j
t_{1ji}, t_{2ki}	travel times from store j to customer zone i, and FC k to customer zone i, respectively
M_{1j}	Capacities at store j
M_{2k}	Capacities at FC k
$\chi_1(d)$	Service distance restriction from stores for demand type d
$\chi_2(d)$	Service distance restriction from FCs for demand type d
c_{kj}	per unit transportation cost from FC k to store j
c_{2kidp}	per unit transportation cost for product p from FC k to customer i for demand type d
c_{1jidp}	per unit transportation cost for product p from store j to customer i for demand type d
θ	service level requirement
Decision Variables	
z_{1j}	1, if store j is selected (for expansion) to be used as a fulfillment center; 0, otherwise
z_{2k}	1, if FC k is opened; 0, otherwise.
x_{1jid}	1, if customer zone i for demand type d is assigned to store j; 0, otherwise
x_{2kid}	1, if customer zone i for demand type d is assigned to FC k; 0, otherwise.
v_{kj}	1, if store j is assigned to FC k; 0, otherwise
w_{kjp}	amount of product p shipped from FC k to store j.
r_{1jp}	auxiliary variable for product p at store j
r_{2kp}	auxiliary variable for product p at FC k

Similarly, the amount of safety stock to be held at those stores $j \in V_1$ that are selected for expansion (i.e. $z_{1j} = 1$) to fulfill online demand is given by $\sum_{p \in P} z_\theta r_{1jp}$, where,

$$r_{1jp} = \sqrt{\mathcal{L}_{1j}\left(\beta_{jp}^2 z_{1j} + \sum_{i \in I}\sum_{d \in D} \sigma_{idp}^2 x_{1jid}\right)} \tag{4.2}$$

Note that in cases where store $j \in V_1$ is assigned to FC $k \in V_2$ i.e. $v_{kj} = 1$ and the store j is also selected for expansion to fulfill online customer demand i.e., $z_{1j} = 1$, the term $\beta_{jp}^2 v_{kj}(1 - z_{1j})$ in equation (4.1) is equal to zero. This is to avoid the double counting of safety stock for the variation in demand originating from store j. Also, due to single assignment assumption of online customer demand, safety stock corresponding to variation in online demand for any customer i for demand type d is allocated to only one facility (a FC or store).

With the above settings, the distribution network design model seeks to simultaneously select stores (from an existing set) to operate as fulfillment centers as well, to locate FCs, assign stores to FCs, and assign online customer demand to stores/FCs while minimizing fixed costs, transportation costs and safety stock inventory holding costs at facilities. The model can be stated as the following mixed integer nonlinear program:

$$\begin{aligned}
(\mathbf{P}) \quad \min \quad & \sum_{k \in V_2} F_{2k} z_{2k} + \sum_{j \in V_1} F_{1j} z_{1j} + \sum_{k \in V_2}\sum_{p \in P}\sum_{j \in V_1} c_{kj}\, w_{kjp} + \sum_{k \in V_2}\sum_{p \in P}\sum_{i \in I}\sum_{d \in D} c_{2kidp}\, \mu_{idp}\, x_{2kid} + \\
& \sum_{j \in V_1}\sum_{i \in I}\sum_{d \in D}\sum_{p \in P} c_{1jidp} \mu_{idp} x_{1jid} + \sum_{k \in V_2}\sum_{p \in P} h_{2kp} z_\theta r_{2kp} + \sum_{j \in V_1}\sum_{p \in P} h_{1jp} z_\theta r_{1jp} && (4.3) \\
\text{s.t.} \quad & \sum_{p \in P}\left(\sum_{j \in V_1} w_{kjp} + \sum_{i \in I}\sum_{d \in D} \mu_{idp} x_{2kid} + z_\theta r_{2kp}\right) \le M_{2k} z_{2k} \;\; \forall k \in V_2 && (4.4) \\
& \sqrt{\mathcal{L}_{2k}\left(\sum_{j \in V_1} \beta_{jp}^2 v_{kj}(1 - z_{1j}) + \sum_{i \in I}\sum_{d \in D} \sigma_{idp}^2 x_{2kid}\right)} = r_{2kp} \;\; \forall k \in V_2, p \in P && (4.5) \\
& \sum_{p \in P}\left(\sum_{i \in I}\sum_{d \in D} \mu_{idp} x_{1jid} + z_\theta r_{1jp}\right) \le M_{1j} z_{1j} \;\; \forall j \in V_1 && (4.6)
\end{aligned}$$

$$\sqrt{\mathcal{L}_{1j}\left(\beta_{jp}^2 z_{1j} + \sum_{i\in I}\sum_{d\in D}\sigma_{idp}^2 x_{1jid}\right)} = r_{1jp} \ \ \forall j \in V_1 \tag{4.7}$$

$$\sum_{k\in V_2} w_{kjp} = \ \alpha_{jp} + \sum_{i\in I}\sum_{d\in D}\mu_{idp}x_{1jid} + z_\theta r_{1jp} \ \ \forall j \in V_1, p \in P \tag{4.8}$$

$$w_{kjp} \le \min\{M_{2k}, M_{1j}\}v_{kj} \ \ \forall k \in V_2, j \in V_1, p \in P \tag{4.9}$$

$$v_{kj} \le z_{2k} \ \ \forall k \in V_2, j \in V_1 \tag{4.10}$$

$$\sum_{k\in V_2} v_{kj} = 1 \ \ \forall j \in V_1 \tag{4.11}$$

$$\sum_{k\in V_2} x_{2kid} + \sum_{j\in V_1} x_{1jid} = 1 \ \ \forall i \in I, d \in D \tag{4.12}$$

$$x_{1jid} = 0 \quad \text{if } t_{1ji} > \chi_1(d) \quad \forall j \in V_1, i \in I, d \in D \tag{4.13}$$

$$x_{2kid} = 0 \quad \text{if } t_{2ik} > \chi_2(d) \ \ \forall k \in V_2, i \in I, d \in D \tag{4.14}$$

$$z_{2k}, z_{1j}, v_{kj}, x_{2kidp}, x_{1jidp} \in \{0,1\}, w_{kjp} \ge 0 \tag{4.15}$$

The first term in the objective function is the fixed cost of opening new FCs whereas the second term is the fixed cost to use an existing B&M store as a fulfillment center. The third and fourth terms represent transportation costs from FCs to stores and from FCs to online customer demand zones, respectively. Similarly, the fifth term represents the cost to ship from stores to customer zones. The sixth and seventh terms are inventory holding costs at FCs and stores, respectively. Equations (4.5) and (4.7) are auxiliary constraints for safety stock at FCs and stores, respectively.

Constraints (4.4) and (4.6) are capacity restrictions at FCs and stores, respectively. Constraints (4.8) are flow conservation at stores, which ensures that for every product that flows through the store, a part of it is held in safety stock and the rest is used to satisfy customer demand. Constraints (4.9) and (4.10) link flows between FCs and stores. Constraints (4.11) and (4.12) are the demand satisfaction constraints that also model single assignment requirement at stores and customer zones, respectively. Constraints (4.13) and (4.14) guarantee service distance restriction for online demand arising from customer zones. Constraint

(4.15) are standard integrality and non-negativity restrictions.

4.4 Solution Method

For problem P, we can use any approach that is valid for concave minimization problems, for instance Branch and Bound (Soland [115]). However, in this case we exploit the properties of the model to reformulate P as a *conic quadratic mixed integer programming* problem that can be solved directly using any state-of-the-art commercial solver (that can solve second order conic problems).

4.4.1 Conic Quadratic Mixed Integer Reformulation

To reformulate **(P)**, we exploit following properties: (i) in constraint (4.5), we let s_{kj} be an auxiliary binary variable, such that $s_{kj} = v_{kj}(1 - z_{1j})$, (ii) as x_{2kid} is a binary variable, thus $x_{2kid} = x_{2kid}^2$ (similar is the case with $z_{1j}, x_{1jid},$ and s_{kj}), and (iii) since we are minimizing over variables r_{2kp} and r_{1jp} in the objective function (4.3), the constraints (4.5) and (4.7) can be expressed in *less than equal to* form. With these transformations and linearization of the product of binary variables, the reformulated conic quadratic mixed integer programming **(CQMIP)** problem is stated as follows.

$$
\begin{aligned}
\textbf{(CQMIP)}\ \min & \sum_{k \in V_2} F_{2k} z_{2k} + \sum_{j \in V_1} F_{1j} z_{1j} + \sum_{k \in V_2} \sum_{p \in P} \left(\sum_{j \in V_1} c_{kj}\, w_{kjp} + \sum_{i \in I} \sum_{d \in D} c_{2kidp}\, \mu_{idp}\, x_{2kid} \right) + \\
& \sum_{j \in V_1} \sum_{i \in I} \sum_{d \in D} \sum_{p \in P} c_{1jidp} \mu_{idp} x_{1jid} + \sum_{k \in V_2} \sum_{p \in P} h_{2kp} z_\theta r_{2kp} + \sum_{j \in V_1} \sum_{p \in P} h_{1jp} z_\theta r_{1jp} \\
\text{s.t.}\quad & (4.4), (4.6), (4.8) - (4.14) \\
& \mathcal{L}_{2k} \left(\sum_{j \in V_1} \beta_{jp}^2 s_{kj}^2 + \sum_{i \in I} \sum_{d \in D} \sigma_{idp}^2 x_{2kid}^2 \right) \leq r_{2kp}^2 \quad \forall k \in V_2, p \in P && (4.16) \\
& \mathcal{L}_{1j} \left(\beta_{jp}^2 z_{1j}^2 + \sum_{i \in I} \sum_{d \in D} \sigma_{idp}^2 x_{1jid}^2 \right) \leq r_{1jp}^2 \quad \forall j \in V_1 && (4.17)
\end{aligned}
$$

$$s_{kj} \leq v_{kj} \ \ \forall k \in V_2, j \in V_1 \tag{4.18}$$

$$s_{kj} \leq (1 - z_{1j}) \ \ \forall k \in V_2, j \in V_1 \tag{4.19}$$

$$s_{kj} \geq v_{kj} - z_{1j} \quad \forall k \in V_2, j \in V_1 \tag{4.20}$$

$$z_{2k}, z_{1j}, v_{kj}, x_{2kid}, x_{1jid}, s_{kj}, \in \{0, 1\} \quad \forall i, j, k, d, p \tag{4.21}$$

$$w_{kjp}, r_{2kp}, r_{1jp} \geq 0 \quad \forall i, j, k, d, p \tag{4.22}$$

Constraints (4.16) and (4.17) are conic quadratic constraints, and (4.18)-(4.20) linearize the product of binary variables.

4.4.2 Polymatroids Inequalities

We next describe polymatroid inequalities which can be obtained from constraints (4.16) and (4.17) to strengthen the (CQMIP) formulation. To do that, we provide few additional definitions. First, let N be a finite set, and $f : 2^N \rightarrow \mathbb{R}$ be a real-valued set function on N.

Definition 1. *The set function $f : 2^N \rightarrow \mathbb{R}$ is a submodular function if $f(K \cup k) - f(K) \geq f(M \cup k) - f(M)$, $\forall\, K \subseteq M \subseteq N$, and $k \in N \setminus K$.*

Definition 2. *For any set function f on N, the polyhedron EP_f is defined as*

$$EP_f := \{\pi \in \mathbb{R}^N : \ \pi(T) \leq f(T) \ \ \forall T \subseteq N\},$$

where $\pi(T) = \sum_{1 \leq i \leq |T|} \pi_i$. If the set function f is submodular, then EP_f is the extended polymatroid of f.

Definition 3. *For any set function f on N, the problem we seek to optimize is defined as*

$$\min_{T \subseteq N} \ f(T). \tag{4.23}$$

Further, let $y \in \{0, 1\}^{|N|}$ be the indicator vector that is supported by the set $S_y \subseteq N$. Then, *the problem in* (4.23) *can be rewritten as:*

$$\min_{(y,t)\in\Theta_f} t, \tag{4.24}$$

where, Θ_f *is the convex lower envelop of* f*, i.e.,*

$$\Theta_f := conv\{(y,t) \in \{0,1\}^{|N|} \times \mathbb{R} : f(y) \leq t\}. \tag{4.25}$$

Atamtürk and Narayanan [10] show that inequality

$$\pi y \leq t, \tag{4.26}$$

is valid for Θ_f *if and only if* $\pi \in EP_f$*. These inequalities are the so-called extended polymatroid inequalities of* Θ_f*.*

In constraint (4.16), since the coefficients $\mathcal{L}_{2k}, \beta_{jp}$ and σ_{idp} are non-negative, $s_{kj}^2 = s_{kj}$ and $x_{kid}^2 = x_{kid}$, the function

$$f(s,x) = \sqrt{\mathcal{L}_{2k}\bigg(\sum_{j\in V_1}\beta_{jp}^2 s_{kj} + \sum_{i\in I}\sum_{d\in D}\sigma_{idp}^2 x_{2kid}\bigg)}$$

is submodular, where (s,x) represents the indicator vector and variable r_{2kp} is analogous to variable t in equation (4.26). Using similar arguments for equation (4.17), we can reformulate these constraints using the *extended polymatroid inequalities* (4.26).

4.4.3 A Branch-and-cut algorithm

We next present a branch-and-but (B&C) algorithm to solve **(CQMIP)**. To improve the convergence of this algorithm, we use the *extended polymatroid inequalities* described in the previous section. Although extended polymatroid inequalities are exponential in size, we only need to add in practice a small subset of them during the enumeration process. In the B&C algorithm, we separate these inequalities at selected fractional solutions using the *usercutcallback* (UCC) function of CPLEX Callable library. To check whether a fractional solution $(\bar{y}, \bar{t})$ violates one of these inequalities requires solving the optimization problem **[PT]**, $\xi := \max\{\pi\bar{y} : \pi \in EP_f\}$. Let π^* be the optimal solution of **[PT]**, and if $\xi > \bar{t}$ then the inequality $\pi^* y \leq t$ separates the point $(\bar{y}, \bar{t})$. We adopt the greedy procedure (Algorithm (2)) of Edmonds [37] to separate these inequalities.

Algorithm 2: A greedy heuristic algorithm to find polymatroid inequalities

Data: Fractional point $\bar{\mathbf{y}}$ and $\bar{\mathbf{t}}$

At a fractional solution $\bar{y} \in [0,1]^{|N|}$ and $\bar{t}$, sort $\bar{y}$ in nonincreasing order $\bar{y}_{(1)} \geq \bar{y}_{(2)} \geq ... \geq \bar{y}_{(|N|)}$.

Let $T_i = \{(1), (2), ..., (|N|)\}$, and $\bar{\pi}_i = f(T_{(i)}) - f(T_{(i-1)})\ \forall\, 1 \leq i \leq |N|$ and $f(\emptyset) = 0$.

If $\xi = \bar{\pi}\bar{y} > \bar{t}$, we add the extended polymatroid inequality $\bar{\pi} y \leq t$.

We assume $f(\emptyset) = 0$ without loss of generality (Atamtürk and Narayanan [10])

4.5 Computational Experiments

We perform a set of extensive computational experiments to assess the performance of our algorithm under different cost settings and capacity scenarios. All the algorithms are coded in C and ran on an Intel Xeon E5-2687W v3 processor at 3.10 GHz in a Linux environment. The algorithm is implemented using CPLEX 12.9 Callable Library using a single thread. Next, we present the summary of the results of experiments. We also study the change in network configuration and safety stock allocation under different cost and capacity scenarios.

4.5.1 Test Instances

To create test instances, we use thirteen locations (currently supporting home delivery channel) of Best Buy Stores as B&M stores. For FCs, we use two potential locations: *Piscataway, New Jersey*, where Best Buy has an already operational FC, which is nearly 30 miles from Newark, and *Yonkers*, a town outside of New York City. Therefore, in every instance, we have total 15 facilities: 13 stores and 2 FCs. The latitude and longitude information of these facilities is obtained from a search on google map, and in-store demand at the stores is based on the population *(PB)* of the borough (Manhatttan, Bronx, Brookly, Queens, and Staten Island) or city (Newark and Jersey City) in which the store is located[1]. For customers zones, we consider all the zip codes from New York City, Newark and Jersey City. With this our data consists of 197 customer locations with population (PC), latitude and longitude based on the zip code of each location[1]. We consider $|I|$ most populous locations as customer zones. Finally, at every B&M store and customer location, we assume that there is a demand for every product $p \in P$ with every demand type $d \in D$.

The data is generated based on the scheme used in Vidyarthi et al. [123]. In every instance, we set $\mu_{idp} = PC_i \times U[0.0001, 0.0005] + U[50, 100]$, $\sigma_{idp} = \mu_{idp} \times U[0.01, 0.10]$, $\alpha_{jp} = PB_j \times U[0.004, 0.008] + U[100, 500]$, and $\beta_{jp} = \alpha_{jp} \times U[0.001, 0.1]$. The capacities at B&M stores and FCs are first generated randomly on $U[500, 1000]$ and then scaled such that $\frac{\sum_j M_{1j}}{Total\ Demand} = \frac{\sum_k M_{2k}}{Total\ Demand} = \kappa$, where $Total\ Demand = \sum_{jp} \alpha_{jp} + \sum_{idp} \mu_{idp}$ and κ is capacity factor. The fixed costs to expand a store and to open a FC are $F_{1j} = f_1^c \times 10 \times U[0.5, 1.0] \times M_{1j}^{0.75}$ and $F_{2k} = f_2^c \times 10 \times U[0.5, 1.0] \times M_{2k}^{0.75}$, respectively, where $f_1^c \geq 0$ and $f_2^c \geq 0$ are scaling constants. The holding costs for every product at stores and FCs are $h_{1jp} = h_1^c \times 2 \times U[5, 20]$ and $h_{2kp} = h_2^c \times 1.5 \times U[5, 20]$, respectively, where $h_1^c \geq 0$ and $h_2^c \geq 0$ are scaling constants. The transportation costs from: (i) FC k to store j, $c_{kj} = t^c \times 0.005 \times U[5, 10] \times t_{kj}$, (ii) store j to customer i for product p and demand type $d = 1$, $c_{1jidp} = t_1^c \times 0.01 \times U[5, 10] \times t_{1ji}$, and for $d > 1$, $c_{1jidp} = 1.25 \times c_{1ji(d-1)p}$, and (iii) FC k to customer i for product p and demand type $d = 1$, $c_{2kidp} = t_2^c \times 0.025 \times U[5, 10] \times t_{2ki}$, and for $d > 1, c_{2kidp} = 1.25 \times c_{2ki(d-1)p}$, where t_{kj}, t_{1ji}, and t_{2ki}, are distance between two nodes calculated using greater circle distance formula and $t^c \geq 0, t_1^c \geq 0$, and $t_2^c \geq 0$ are scaling constants.

We generate twenty problem instances, where the largest instance consists of 2 FCs ($|V_2|$), 13 B&M stores ($|V_1|$), 60 customers zones ($|I|$), 2 demand types ($|D|$) and 100 product

[1] http://zipatlas.com/us/ny/zip-code-comparison/population-density.htm

Table 4.2: Specifics of the Instances Used in the Computational Experiments

Set	Instance#	$\lvert V_2\rvert$	$\lvert V_1\rvert$	$\lvert I\rvert$	$\lvert D\rvert$	$\lvert P\rvert$
I	1, 2, 3, 4	2	13	20	2	25, 50, 75, 100
II	5, 6, 7, 8	2	13	30	2	25, 50, 75, 100
III	9, 10, 11, 12	2	13	40	2	25, 50, 75, 100
IV	13, 14, 15, 16	2	13	50	2	25, 50, 75, 100
V	17, 18, 19, 20	2	13	60	2	25, 50, 75, 100

categories ($|P|$). Details about the test instances are provided in Table 4.2. Further, we vary model parameters to include seven scenarios that captures different cost and capacity settings in the computational analysis. The scenarios are obtained by setting the parameters as follows:

- Balanced Cost (BC) scenario: Under this scenario, $\kappa = 5$ whereas all scaling constants are set to their default value of 1. There are no restrictions on $\chi_2(d)$ and $\chi_1(d_1)$, whereas $\chi_1(d_2) = 15$ miles.

- Dominant Fixed Cost (DFC) scenario: Under this scenario, f_1^c and f_2^c are set to 5, and $\kappa = 5$ whereas all scaling constants are set to their default value of 1.

- Dominant Safety Stock Cost (DSC) scenario: Under this scenario, we set h_1^c and h_2^c to 4, and $\kappa = 5$ whereas all scaling constants are set to their default value of 1.

- Dominant Transportation Cost (DTC) scenario: We set $t^c = t_1^c = t_2^c = 4$ and $\kappa = 5$ whereas all scaling constants are set to their default value of 1.

- Service Distance (SD) scenario: We set $\chi_2(d_2) = 15$ miles. There are no service distance restrictions on $\chi_1(d_1)$ and $\chi_2(d_1)$. $\kappa = 5$. All other scaling constants are set to their default value of 1.

- Tight Capacity (TCap) Scenario: The instances are generated by setting $\kappa = 3$ whereas all other scaling constants are set to their default value of 1.

- Loose Capacity (LCap) Scenario: The instances are generated by setting $\kappa = 10$ whereas all other scaling constants are set to their default value of 1.

4.5.2 Computational Results

In this section, we compare the performance of the proposed solution approach with CPLEX, where **(CQMIP)** is directly solved using the SOCP solver of CPLEX with its default settings. In the proposed solution method, referred as B&C, we solve **(CQMIP)** by adding extended polymatroid inequalities at fractional solutions.

In our experiments, we observed that the performance of B&C algorithm is sensitive to various parameters used in the implementation of UCC function. The parameters tuned to improve the convergence of the B&C algorithm are described as follows. We first define, separation tolerance (ST) as the tolerance value for violation of inequalities by the fractional solution. Also, we control the number of cuts to be added at a node in the enumeration tree using two parameters: (i) *CSN*, which is defined as the limit on number of calls to be made to UCC function from the same node, and (ii) EPSOBJ, which defines that UCC is called again only if the improvement in the lower bound is above a fixed percentage after adding cuts. Finally, the parameter *Depth* is defined to control how frequently cuts be added in the enumeration tree. Based on our preliminary experiment, we set the parameters as follows: $ST = 0.001$, $CSN = 10$, $EPSOBJ = 0.1$, and $Depth = 10$.

The summary of results are presented in Table 4.3 for all five sets of instances under seven scenarios that provided reasonable feasible solution using CPLEX and B&C algorithm. The detailed results of 140 instances are provided in Tables C.1 - C.7 in the Appendix. In all our experiments, we solve each instance to the optimality gap of 0.005% within a time limit of 86,400 seconds. In the table, column headings "Gap (%)", "# Inst.Opt." and "Time (sec)" are average optimality gap, number of instances solved up to desired optimality gap and average computation time, respectively, for every Scenario-Set. In column "%Red", we compare the savings in computational time of the B&C algorithm over CPLEX. Finally, we use the notation $N_{CPLEX}/N_{B\&C}$ in the last column, where N_{CPLEX} represents the number of instances out of (N_{CPLEX} + $N_{B\&C}$) in which CPLEX outperformed B&C, and similarly, $N_{B\&C}$ represents the number of instances where B&C outperformed CPLEX.

First, we compare the performance of two solution approaches in terms of average optimality gap and number of instances solved to optimality. Results in Table 4.3 indicate that the overall average optimality gap provided by B&C is 0.23% lower than that by CPLEX. The optimality gap is significantly lower in SD scenario (0.74%) followed by TCap (0.4%) and BC (0.33%) scenarios, whereas the reduction in optimality gap is moderate for DSC (0.15%) scenario. The optimality gaps obtained in DTC, DFC and LCap are approximately equal. This lower gap provided by B&C is particularly due to better performance of B&C method in large size instances of Sets III, IV and V. For instance, in SD-III (Scenario-Set) the optimality gap from CPLEX is 2.89% higher than that by B&C. Similarly, the reduction in optimality gap by B&C is 1.49% in TCap-IV, 1.31% in BC-IV, 0.72% in TCap-V and 0.34% in DSC-V scenarios. Additionally, B&C solves 109 of the 140 instances whereas CPLEX solves 101 of the 140 instances to the desired optimality gap.

Next, we compare the average computation times taken to solve the instances. Over all scenarios and set sizes, B&C takes 35% less computational time than CPLEX. Primarily, lower computation times by B&C are observed in DTC scenario with 52% reduction, followed by 48% in LCap, 45% reduction for DSC, and 29% in SD. Whereas in case of DFC (17%) the reduction is moderate, and in TCap the average computation times are almost same. We observe that B&C is particularly faster for large size instances. For example, in the instances belonging to DFC-IV, SD-IV, and DTC-IV, using B&C algorithm reduces computational time by 92%, 73% and 61%, respectively.

In Table 4.4, we compare the average performances of B&C and CPLEX by instance sizes. We observe that B&C method performs well for all five sets in terms of providing lower optimality gap and reducing computational times. E.g., in large size instances of Sets III, IV and V, the average reduction in computational time by B&C is 23%, 28% and 28%,

Table 4.3: Summary of Performance of Branch-and-Cut Algorithm and CPLEX under different cost and capacity scenarios

		Gap (%)		# Inst. Opt.		Time (sec)			
Scenario	Set	CPLEX	B&C	CPLEX	B&C	CPLEX	B&C	%Red.	N_{CPLEX} / $N_{B\&C}$
BC	I	0.00	0.00	4	4	21,156	14,220	33	0 / 4
	II	0.00	0.00	4	4	21,262	13,369	37	0 / 4
	III	0.48	0.00	2	3	38,936	32,619	16	1 / 2
	IV	1.36	0.05	2	2	33,808	30,531	10	0 / 3
	V	0.00	0.00	2	2	25,982	25,811	1	0 / 2
	Average	**0.34**	**0.01**	**14**	**15**	**27,492**	**21,964**	**20**	**1 / 15**
DFC	I	0.00	0.00	4	4	314	1117	-72	3 / 1
	II	0.00	0.00	4	4	697	278	60	0 / 4
	III	0.00	0.01	4	3	12,915	27,188	-52	2 / 2
	IV	0.00	0.00	4	4	14,972	1,163	92	0 / 4
	V	0.05	0.00	2	3	38,159	23,703	38	0 / 3
	Average	**0.01**	**0.00**	**18**	**18**	**12,109**	**10,005**	**17**	**5 / 14**
DSC	I	0.21	0.00	3	4	35,284	20,507	42	0 / 4
	II	0.00	0.00	4	4	36,587	22,650	38	0 / 4
	III	0.31	0.00	1	2	56,050	33,887	40	0 / 2
	IV	0.00	0.00	2	2	15,831	9,299	41	0 / 2
	V	0.34	0.00	0	2	86,278	37,627	56	0 / 2
	Average	**0.15**	**0.00**	**10**	**14**	**43,129**	**23,875**	**45**	**0 / 14**
DTC	I	0.00	0.00	4	4	4,346	2,020	54	0 / 4
	II	0.00	0.00	4	4	5,596	1,741	69	0 / 4
	III	0.00	0.00	4	4	25,898	12,876	50	1 / 3
	IV	0.00	0.00	4	4	23,241	8,991	61	0 / 4
	V	0.09	0.00	3	4	34,594	19,081	45	0 / 4
	Average	**0.02**	**0.00**	**19**	**20**	**18,735**	**8,942**	**52**	**1 / 19**
SD	I	0.00	0.00	4	4	16,145	10,885	33	0 / 4
	II	0.00	0.00	4	4	18,068	12,667	30	0 / 4
	III	3.10	0.21	3	3	41,860	40,094	4	1 / 3
	IV	0.00	0.00	2	2	22,048	6,063	73	0 / 2
	V	0.14	0.00	1	2	43,867	20,826	53	0 / 2
	Average	**0.79**	**0.05**	**14**	**15**	**27,258**	**19,272**	**29**	**1 / 15**
TCap	I	0.00	0.29	4	3	19,112	27,943	-32	2 / 2
	II	0.35	0.22	3	3	28,559	29,687	-4	2 / 2
	III	0.41	0.01	2	2	63,549	50,530	20	0 / 3
	IV	1.94	0.45	1	2	65,916	69,407	-5	1 / 2
	V	2.12	1.40	0	0	85,466	85,757	0	1 / 1
	Average	**0.79**	**0.39**	**10**	**10**	**46,876**	**47,615**	**-2**	**6 / 10**
LCap	I	0.00	0.00	4	4	1,047	419	60	0 / 4
	II	0.00	0.00	4	4	21,961	924	96	1 / 3
	III	0.03	0.00	2	3	30,974	95,93	69	0 / 3
	IV	0.00	0.00	3	3	5,346	4,334	19	0 / 3
	V	0.00	0.00	3	3	30,881	35,522	-13	1 / 2
	Average	**0.01**	**0.00**	**16**	**17**	**17,273**	**9,042**	**48**	**2 / 15**
	Average	**0.30**	**0.07**	**101**	**109**	**27,553**	**20,102**	**27**	**16 / 102**

respectively. Also, for instances in Sets III and IV, the average optimality gap provided by B&C version is 0.58%, and 0.4% lower than that of CPLEX, respectively.

The results from the column 'N_{CPLEX} / $N_{B\&C}$' in Table 4.3 show that B&C outperforms CPLEX on 19 out of 20 instances under DTC, and on all 14 instances in DSC scenario. B&C also performs better than CPLEX on 15 out of 17 and 14 out of 19 instances under LCap and DFC scenarios respectively. Furthermore, out of 16 instance, B&C performs better on 15 instances each under BC and SD scenarios, and on 10 instance under TCap. In summary, B&C outperforms CPLES on 102 out of 118 instances. The performance of B&C is particularly better for large size instances (Table 4.4), where B&C outperforms CPLEX in 54 out of 62 instances belonging to Sets III, IV, and V.

While gauging the ability of the solution method to solve instances under different scenarios, we note that instances under DFC and DTC scenarios are easy to solve in comparison to those under BC, SD, and DSC scenarios. Not only the average optimality gaps provided by DFC and DTC are near to optimality, but also these two scenarios have among the lowest average computational times. Additionally, in both these scenarios almost all instances are solved to optimality. Whereas, both average optimality gap and computational time increased in BC and SD scenario, followed by DSC. Additionally, we measure the performance of our solution method under different capacity scenarios. For this we compare results of BC, LCap and TCap scenarios. As expected, instances with loose capacity scenario are comparatively easy of solve, followed by BC in which average CPU time is twice of LCap. Whereas, TCap instance are most difficult with average CPU time nearly five times of LCap and 2 times of BC, and with significant increase in average optimality gap.

Table 4.4: Summary of performance of branch-and-cut algorithm and CPLEX by Instance sets

	Gap(%)		# Inst. Opt.		Time (sec)			
Set	CPLEX	B&C	CPLEX	B&C	CPLEX	B&C	%Red.	N_{CPLEX} / $N_{B\&C}$
I	0.03	0.04	27	27	13,915	11,016	21	5 / 23
II	0.05	0.03	27	27	18,962	11,617	39	3 / 25
III	0.62	0.04	18	20	38,597	29,541	23	5 / 18
IV	0.47	0.07	18	19	25,880	18,541	28	1 / 20
V	0.39	0.20	11	16	49,318	35,475	28	2 / 16

4.6 Sensitivity Analysis and Managerial Insights

In this section, we address our research questions and provide some managerial insights. We also vary scaling constants related to safety stock costs to analyze how sensitive the model is with respect to this cost parameters.

4.6.1 Benefits of Channel Integration: Dedicated Channels vs. Integrated Distribution Channel

Many retailers use dedicated distribution centers for in-store and online sales channels due to ease of operations. In fact, many retailers consider online demand channel as another business entity (or store) for the purpose of planning (Hübner et al. [67]). However, integrating distribution channels offer the potential for lowering inventory requirements due to risk pooling and reducing overall network costs. We analyze the benefits of using integrated distribution channels over dedicated distribution channels. To show these benefits, we compare two scenarios. The first scenario is the *dedicated channel*, where the planning for in-store and online demand is done separately using different sets of potential FCs for both demands, and the second one is *integrated channel* in which two demand types and their respective sets of potential FCs are combined as one set, and the logistic activities are planned jointly. We consider four potential locations for FCs: Piscataway (NJ), Monsey (NY), Freeport (NY), and Yonkers (NY), where Piscataway and Monsey are a set of potential FCs to satisfy in-store demand only, and Freeport and Yonkers are a potential set to fulfill online demand. Further, we use thirteen locations of Best Buy as B&M stores and 50 zip codes from the city of New York, Newark, and Jersey City as customer zones. Additionally, we assume that both demands are to be served directly from FCs only. Table 4.5 shows the difference between the results of dedicated distribution channels and integrated distribution channel. This leads to the following observation:

Observation 1: *Integrated distribution channel is cost effective compared to the dedicated distribution channels mainly due to risk pooling and reduction in safety stock and the associated holding costs.*

In the dedicated channels scenario, to satisfy in-store demand, model opens a FC at Monsey, with total network cost of $49,550 and safety stock of 401 units. Similarly, to serve online demand, a FC is opened at Yonkers with the optimal cost of $62,607, and safety stock level of 220 units. Therefore, the total costs to serve both demands in the dedicated channels scenario is $112,157, and total safety stock is 621 units. Next, we consider both in-store and online demands, and all four potential locations for FCs as a set for the integrated channels scenario. The optimal cost in this scenario is $70,437, and safety stock is 359 units. Thus, integrated channels offer a savings of 37% in total costs, and 42% reduction in safety stock. The major savings in costs are attributed to facility fixed cost, as dedicated channel locates FCs at Monsey and Yonkers, whereas in integrated channel all the distribution activities are carried out from FC opened at Yonkers, thus it reduces fixed cost by 69%, safety stock cost by 13% and transportation cost by 9%. The reduction in safety stock is due to the fact that inventory is now pooled only at one location. The transportation cost is reduced because Yonkers is closer to the stores and customer zones than other locations for FCs, and it also has sufficient capacity to serve both types of demands.

Table 4.5: Dedicated Distribution Channels vs. Integrated Distribution Channel

	Dedicated Channel		Integrated Channel
	In-Store Demand	Online Demand	In-Store and Online Demand
FC Location	Monsey	Yonkers	Yonkers
Safety Stock Levels	401	220	359
Facility Fixed Cost ($)	35,980	16,253	16,253
Transportation Cost ($)	9,251	41,452	46,197
Safety Stock Cost ($)	4,319	4,902	7,987
Total Cost ($)	**49,550**	**62,607**	**70,437**

4.6.2 Benefits of Integrating FCs and Stores to Fulfill Online Demand

In the previous subsection 4.6.1, we studied the benefits of jointly planning for in-store and online demand using integrated FCs assuming that only FCs can be used to serve both demands. Next, we evaluate whether is it beneficial to integrate stores as well in distribution activities to fulfill online demand. We consider an integrated model with same instance used in Section 4.6.1, and compare the performance of three scenarios in Table 4.6. In *partial integration-FC* (PI-FC) scenario, online demand is served only from FCs; in *partial integration-Store* (PI-Store) scenario, online demand is served only from stores, and in *full integration* (FI), online demand can be allocated to both stores and FCs. In all three scenarios, in-store demand is served from store only.

Table 4.6: Analysis of integrating FCs and Stores in omnichannel distribution

	Scenario		
	Partial Integration, PI-FC	Partial Integration, PI-Store	Full Integration
Locations of FCs	Yonkers	Yonkers	Yonkers
Locations of Stores		5102 Ave, Brooklyn	5102 Ave, Brooklyn
		N. Blvd, Queens	
Online Demand Assigned to FCs (%)	100	0	31
Online Demand Assigned to Stores (%)	0	100	69
Facility Fixed Cost	16,253	39,477	27,727
Transportation Cost	46,197	20,559	26,002
Safety Stock Cost	7,987	17,820	13,009
Total Cost ($)	**70,437**	**77,856**	**66,738**
Safety Stock at FCs (Units)	359	272	305
Safety Stock at Stores (Units)		660	378
Total Safety Stock (Units)	**359**	**932**	**683**

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost

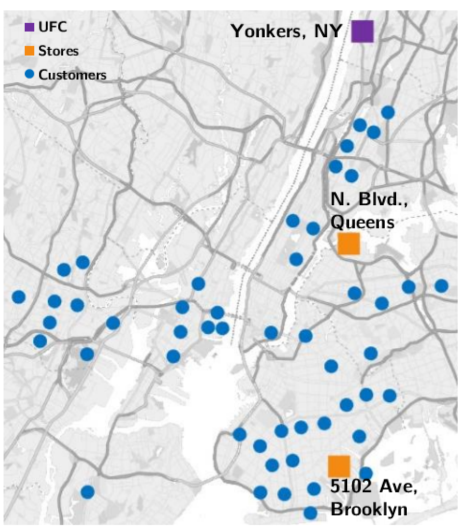
Note that none of the store has sufficient capacity to satisfy all the online demand. Therefore, in PI-Store scenario, the model is forced to open two stores (5102 Ave, Brooklyn and N. Blvd, Queens) and one FC, thereby making it the scenario with highest fixed cost. Also, in this scenario, the safety stock is highest, and subsequently, safety stock holding cost is very high. Thus, decentralization of safety stock for the variation in online demand is the most expensive strategy with network cost of $77,856 (Table 4.6). An alternative to this strategy is the centralization of safety stock at FCs (scenario PI-FC). This reduces the network cost by 10%. Since, Yonkers FC has sufficient capacity to satisfy both in-store and online demand, therefore model suggest opening only Yonkers, which reduces fixed cost by 59%. Also, due to risk pooling, safety stock and the corresponding costs are decreased by 61% and 55%, respectively. However, transportation cost in PI-FC scenario is 2.25 times higher than that in PI-Store because Yonkers is comparatively far from customer zones (Figure 4.1a).

The best strategy in terms of network cost is scenario FI, when model can choose to allocate online demand to both stores and FCs. The network cost in FI scenario is 5% and 14% lower than that from PI-FC and PI-Store scenarios, respectively. Due to the partitioning of online demand between stores and FCs in scenario FI, the fixed cost is higher and the benefits of risk pooling are not realized as much in comparison to PI-FC. Hence, FI, leads to 47% higher safety stock (39% higher safety stock cost) than PI-FC. But due to assignment of 61% of demand to store, transportation cost in scenario FI is reduced by 44%. On the contrary, when compared with PI-Store scenario, the major reduction in network cost in FI is mainly due to fixed cost (30%), and safety stock cost (27%).

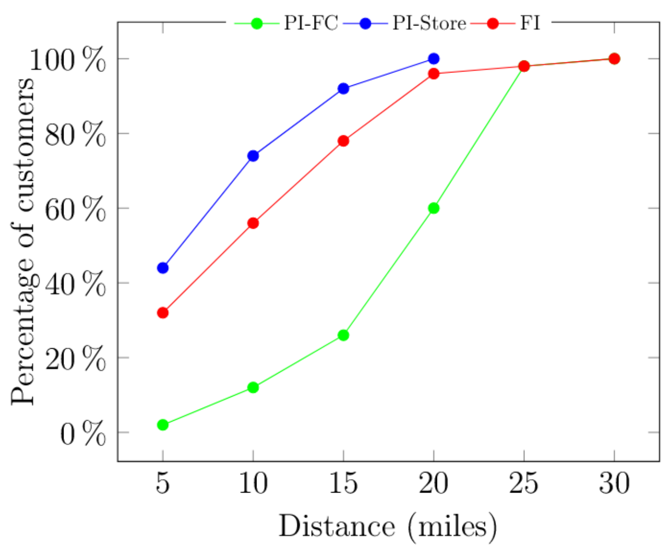
In addition to the network costs and safety stock levels, we also compare the performance of three scenarios in terms of their service distance at it effects delivery time commitments. Figure 4.1a shows the location of opened FC and selected stores, and Figure 4.1b depicts the percentage of customers served within various distances in every setting. For instance, in scenario PI-FC, 2% customers are assigned to facilities that are within 5 miles of these customers. We observe that scenario PI-Store provides best service in terms of delivery times as 74%, 92% and 100% customers are within 10, 15 and 20 miles of any facility (FC / Store) respectively. For the same distances i.e. 10, 15 and 20 miles, scenario FI serves 56%, 78% and 96% customers, respectively, and PI-FC scenario serves 12%, 26% and 60% customers.

This is because in PI-Store the selected stores (N. Blvd., Queens and 5102 Ave, Brooklyn) to serve online customer demand are closer to many customer zones, whereas the FC opened in other two scenarios to serve online demand is on the edge of the city. This leads to the following observation:

Observation 2: *Fulfilling all the online demand from FCs (PI-FC scenario) results in lower safety stock (due to risk pooling) and moderate logistics cost, however it comes at the expense of longer delivery time (due to greater service distance). Fulfilling all the online demand from stores (PI-Store scenario) provides shortest delivery times among the three settings but it has the high logistics cost, and the higher safety stock. Fulfilling the online demand from stores and FCs (FI Scenario) has the lowest logistics cost and provides a balance between service time and safety stock.*



(a) Opened FCs, stores and customer zones



(b) Percentage of customers served within a specified distance for three scenarios

Figure 4.1: Service Distance Analysis when FCs and Stores are integrated to serve online demand

4.6.3 Sensitivity of the model to service distance requirements for online demand

In this set of experiment, we study the change in location and allocation decisions, and network costs to different service distance (SD) requirement. A service distance is defined as the maximum distance between a customer and a facility (FC or store) from which online customer demand is satisfied. We begin with no restrictions on service distance, and gradually decrease this distance from 20 miles to 11 miles (beyond 11 miles the model is infeasible). The outputs of this analysis are presented in Figure 4.2 (percentage increase in total cost with respect to no SD scenario) and Table 4.7 (distribution of various costs, FC: Fixed cost, TrC: Transportation cost, SsC: Safety stock cost, and SS: amount of safety stock). We consider Piscataway and Yonerks as potential locations for FC, thirteen Best Buy stores as potential location to select stores for fulfilment, and fifty zip codes (depicted in Figure 4.3a) from the city of New York, Newark, and Jersey City. Also, in Figures 4.3b - 4.3f, we show the change in location and allocation decisions to satisfy online customer demand for various SD restrictions. This leads to the following observation:

Observation 3: *As the service distance requirements are tightened, more stores are required to fulfill online demand and the assignment of customers become more clustered due to distance restrictions. This decreases the utilization of facilities as percentage share of online demand served by each store decreases with increase in SD restriction. Also, the total safety stock in the system increases.*

In the no SD scenario, a store is opened at 5^{th} Ave, Manhattan because it has lowest per unit safety stock cost and it is in the center of demand points, therefore, the transportation costs is reduced as well. The store operates at 99.9% of its capacity, and satisfies 75% of online demand. The remaining 25% online demand is satisfied from Yonkers FC. The maximum SD obtained in None SD scenario is 20 miles. When SD is decreased, the locations remain the

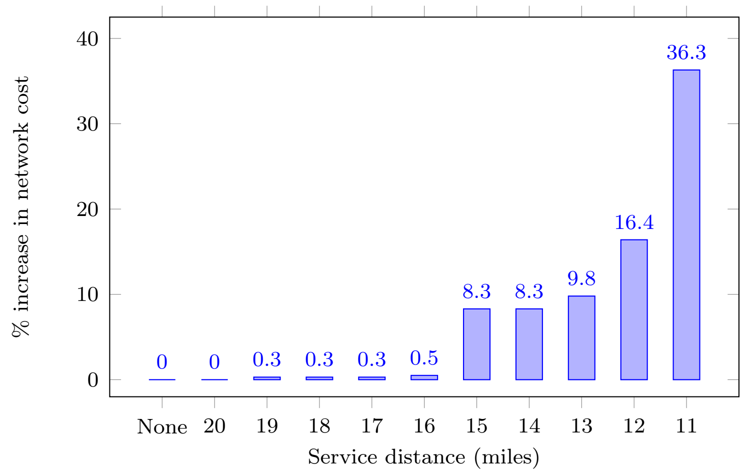
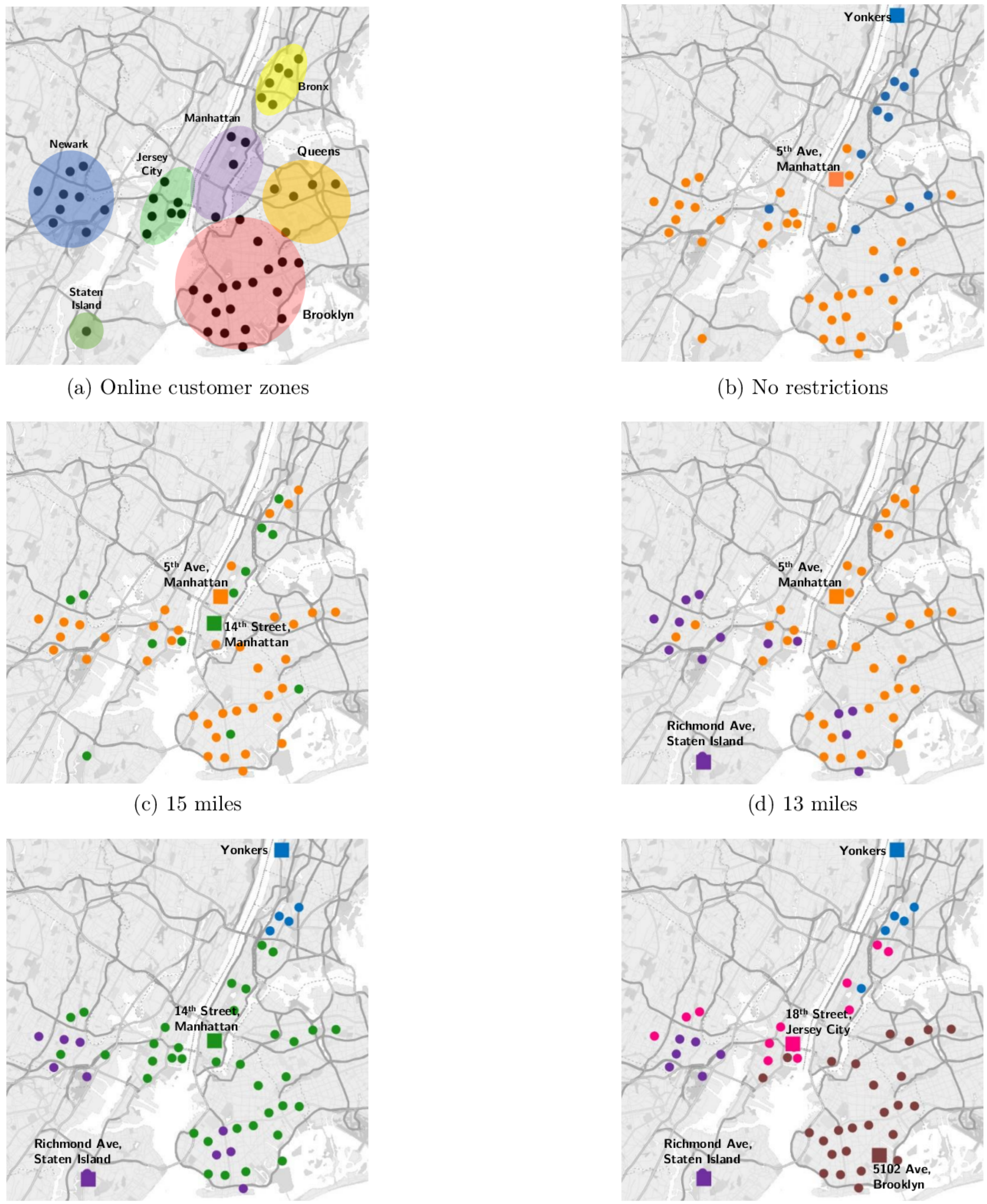


Figure 4.2: Effect of Changing Service Distance Restriction on Network Cost

Table 4.7: Impact of varying SD restriction on the network cost and safety stock levels

Service Distance (Miles)	Facility Fixed Cost ($)	Transportation Cost ($)	Safety Stock Cost ($)	Total Cost ($)	Safety Stock Level (Units)
None	22,286	18,064	11,672	52,022	856
20	22,286	18,064	11,672	52,022	856
19, 18, 17	22,286	18,069	11,843	52,198	858
16	22,286	18,126	11,858	52,270	858
15,14	19,777	22,406	14,181	56,364	913
13	21,076	20,482	15,576	57,134	956
12	25,100	15,341	20,131	60,572	970
11	31,197	16,724	22,971	70,892	1,080

same till SD restrictions up to 16 miles with minor increase in total cost due to reassignment of online demand (Figure 4.2). Note that in no SD scenario, the two customers in Brooklyn region that are served by Yonkers are 17 and 20 miles away from the facility. Therefore, when SD is reduced to 15 miles, the model recommends opening a store at 14^{th} Street Manhattan. However, its per unit safety stock cost is twice that of 5^{th} Ave store. Therefore, 5^{th} Ave store satisfies 75% demand and 25% is satisfied by 14^{th} Street Manhattan. In Table 4.7, from None to 15 miles restriction, fixed cost decreases because Yonkers location is closed, and Piscataway is opened to satisfy in-store demand. The total fixed cost of Piscataway and 14^{th} street store is lower than that of Yonkers. Thus, increase in 8.3% of total network cost is primarily due to transportation and safety stock costs. As Piscataway does not satisfy any online customer, this facility does note appear in the Figure 4.3c.



(a) Online customer zones

(b) No restrictions

(c) 15 miles

(d) 13 miles

(e) 12 miles

(f) 11 miles

Figure 4.3: Service Distance Analysis: Selected FCs, stores and assignments to serve online customer demand

Next, at SD restriction of 13 miles, a store at Richmond Avenue, Staten Island is selected mainly to serve farther customer in Newark region. The increase in total cost from SD 15 miles to 13 miles is only 1.4% because the increase in total cost due to higher fixed cost and per unit safety stock cost at Richmond Ave store is offset by decrease in transportation cost. In this scenario (Figure 4.3d), four customers on the edge of Brooklyn and one customer in Newark are 13 miles away from 5^{th} Ave store. Thus when we set SD as 12 miles, the selected store shifts to 14^{th} street and Yonkers FC is reopened to reach out to the customers in Bronx borough. We observe similar pattern when SD is 11 miles. Additional stores are selected at 18^{th} Street, Jersey and 5102 Ave, Brooklyn, thereby increasing the total cost by 36.3%. The major increase is due to fixed cost as model needs more stores to reach out the customers. Also, the amount of safety stock is 21% higher (856 to 1,080 units in Table 4.7) in comparison to safety stock in no SD scenario.

4.6.4 Sensitivity of the model to safety stock cost parameter

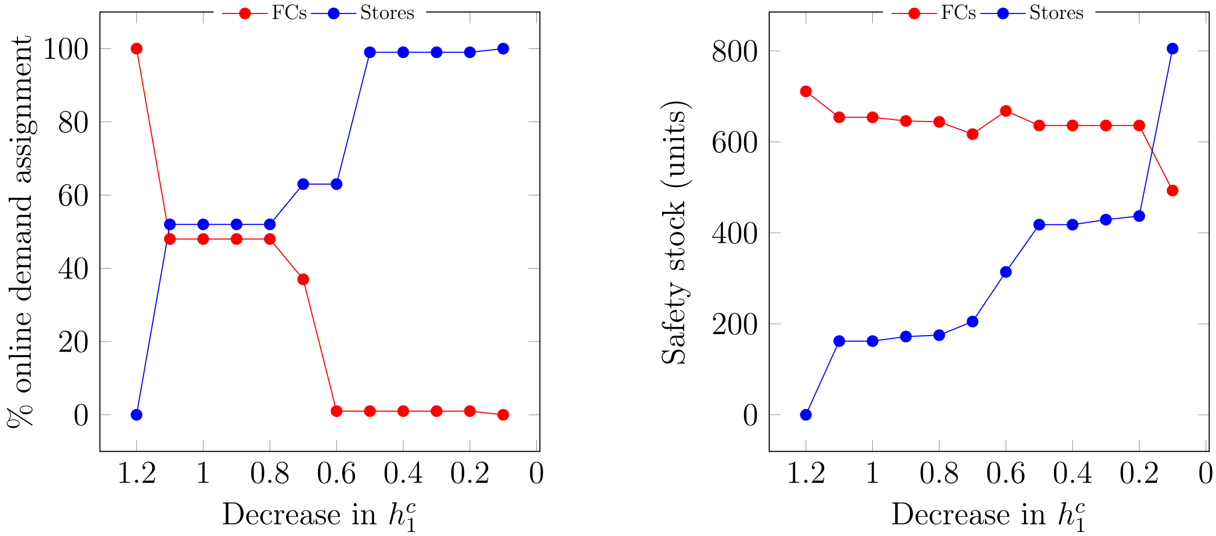
In Section 4.6.2 we observe that introduction of stores for fulfillment recommends partial decentralization of safety stock. Thus, we further tests the sensitivity of our results to per unit safety stock cost parameter. In particular, we are interested in observing the level of centralization vs. decentralization of safety stock, and changes in the location decisions as a result of this. For this we consider integrated planning model with two FCs (Piscataway and Yonkers), thirteen locations of Best Buy stores and fifty most populous zip codes from the city of New York, Newark and Jersey City. We vary scaling cost constant h_1^c from 0.1 to 1.2. The results are presented in Table 4.8, in which second and third columns are location of opened FC and selected store for expansion to assign online demand. Next two columns depict the percentage assignment of online demand to FC and store, and last three columns are safety stock units at facilities.

We observe that as h_1^c decreases, the model reduces the percentage of online demand allocated to FC. Since, the higher value of h_1^c means higher per unit safety stock holding cost at stores, therefore to reduce the total cost (particularly safety stock holding cost), at h_1^c = 1.2, 100% of online demand is assigned to FC. With further decrease in per unit safety stock cost at stores, the model recommends assigning more online demand to the stores. At h_1^c = 0.8, this assignment to the stores is 48% which has also resulted in the increase in safety stock by 15% (711 to 819 units). Similarly, at $h_1^c = 0.1$, it becomes economical to allocate 100% of the online demand to the stores only. Also, the location of FC changes to Piscataway because the fixed cost of Piscataway is nearly 23% lower than that of Yonekrs. However, this has increased the safety stock by another 58% (from 822 to 1,298 units) because to cater 100% of online demand model needs to open two stores, and it leads to the decentralization of inventory. Note that for the value of h_1^c from 0.2 to 0.6, the solution is nearly the same to that at $h_1^c = 0.1$, and then suddenly changes at $h_1^c = 0.7$. These results for every step of h_1^c are also presented in Figure 4.4, where the allocated demand and amount of safety stock stay constant for a range, then suddenly change due to single assignment assumption of customer zones. This leads to the following observation:

Observation 4: *As the safety stock holding cost at stores decreases, the model allocates more demand to stores. Thus, the safety stock at stores increases, thereby recommending decentralization of inventory. However, it increases the fixed cost as more stores are required to accommodate an increase online demand allocation to the stores.*

Table 4.8: Effect of varying safety stock holding cost (h_1^c) on the optimal solutions

	Facility Location		% Online Demand Assigned		Safety Stock Level (Units)		
h_1^c	FC	Stores	FC	Stores	FC	Stores	Total
1.2	Yonkers	-	100	0	711	0	711
0.8	Yonkers	40th Road Queens	48	52	644	175	819
0.7	Yonkers	Gateway dr, Brooklyn	37	63	617	205	822
0.1	Piscataway	Parkway, Brooklyn; 5102 Ave, Brooklyn	0	100	493	805	1,298



(a) Change in online demand allocation to FCs and Stores

(b) Change in safety stock levels at FCs and Stores

Figure 4.4: Analysis of decrease in per unit safety stock cost at stores by decreasing scaling constant h_1^c

4.7 Conclusions

This chapter presents a novel omnichannel distribution network design model for retailers aiming to integrated online sales channel with traditional store sales. We consider a settings where the online orders can be fulfilled from both stores and FCs, whereas in-store demand is fulfilled only from FCs. The model considers both demands together for distribution planning and for the allocation of safety stock to the stores and FCs. The original model is nonlinear for which, we present a conic quadratic mixed integer programming reformulation. The reformulated problem was be solved directly using CPLEX. We also presents a B&C algorithm that includes extended polymatroid inequalities added at fractional solutions. The B&C algorithm reduces computation time by 35%, provides lower optimality gap by 0.25% and performs well on 90% of large size instances. Finally, we present several managerial insights related to network configuration and inventory positioning, using a small example drawn from actual locations of Best Buy stores and FC in the city of New York, Newark and Jersey City.

Chapter 5

Conclusions

This thesis addressed three prominent features that appears frequently while designing supply chains. First, we studied *concave costs* that model economies of scale in variable facility operating and transportation costs, and environmental costs from transportation activities in a typical production-distribution system supply chains where goods and services are produced and shipped at a large scale to satisfy customer (e.g., large retail stores) demands. Second, we focused on designing e-commerce supply chain while considering *order delivery time*, which we assume to be uncertain. Third aspect is *Omnichannel distribution* that entails fulfilling both -online and offline- demands through integrated channels to improve customer experience with minimal costs. All these three supply chains are mathematically stated using facility location models and as mixed-integer nonlinear programming problems.

In Chapter 2, we studied two formulations for a multilevel capacitated facility location problem with concave costs. The first formulation is a traditional mixed integer programming problem in which facility opening decisions and fixed costs are associated with binary variables. In the second formulation, we considered discontinuous fixed-charge functions for facility decision, facility fixed and operating costs, which makes the formulation purely nonlinear. For these two formulations, we developed a branch-and-bound based solution method. The results from extensive computations showed that purely nonlinear model reduces computation times by 30% and performs well in 80 out of 100 test instances.

Chapter 3 addressed a capacitated facility location problem while considering the service level requirements for customers in a dynamic (multiperiod) setting. The service level requirements are presented as time taken to fulfill online orders. This order fulfillment time is dependent on the sum of time spent at facilities for order preparation (waiting time) and time taken to ship order from facilities to customers (travel time). These times are uncertain in practice, however, can be modeled using known probability distributions. Thus, we ensure that customers are served within committed time with some fixed probabilistic guarantee. The mathematical model for the problem is nonconvex due to probabilisitic constraints. We provided two linear reformulations and a branch-and-cut exact solution algorithm to solve the large sizes instance. The computational analysis showed that the best version of our problem provides average optimality gap of 0.11% in 29,296 secs (nearly 8 hours) and 0.97% in 79,244 secs (nearly 22 hours)for small and large size instances, respectively.

Finally, Chapter 4 presented a novel model for omnichannel distribution network design that incorporates fulfillment center location and store selection decisions to fulfill online demand, assignment of online customer demand and safety stock allocation to fulfillment centers and stores. A conic quadratic mixed integer programming based reformulation was presented. We developed a branch-and-cut algorithm using extended polymatroids inequalities for the reformulated problem. The branch-and-cut algorithm clearly showed computational benefits in obtaining lower optimality gaps and reducing solution time.

In addition to their contributions, the three problems studied in this thesis present new research avenues for facility location in supply chain design. For the first problem, exploring a more general variant with network design decisions and their impact on the performance is an important research direction. The exact solution method in the Chapter 3 is based on the fact that the convolution of Exponential (waiting time) and Gamma distributions (travel time) can be computed exactly to give probability values. Many researchers have modeled waiting using general distribution and travel time with lognormal distribution. It would be interesting to investigate the possibilities of finding exact probability values using different distribution for two sources of uncertainty, and how these assumptions related to the distributions would affect the solution quality in terms of computational time and optimality gaps. In Omnichannel distribution network problem in Chapter 4 our focus is only on home delivery channel for online order. A logical forward step in this area could be the inclusion of other channels as well, for instance, use of stores and lockers for the pick-up of online orders. Another line of research could be to model service distance restrictions using stochastic constraints.

Bibliography

[1] K. Aardal, F. A. Chudak, and D. B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters*, 72(5–6): 161–167, 1999.

[2] R. Aboolian, O. Berman, and Z. Drezner. Location and allocation of service units on a congested network. *IIE Transactions*, 40(4):422–433, 2008.

[3] R. Aboolian, O. Berman, and Z. Drezner. The multiple server center location problem. *Annals of Operations Research*, 167(1):337–352, 2009.

[4] R. Aboolian, E. Carrizosa, V. Guerrero, and J. Wang. Locating service facilities with concave variable costs. *Journal of Supply Chain and Operations Management*, 7(3), 2019.

[5] R. Aboolian, O. Berman, and J. Wang. Responsive make-to-order supply chain network design. *Naval Research Logistics (NRL)*, 68(2):241–258, 2021.

[6] J. Acimovic and S. C. Graves. Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing & Service Operations Management*, 17(1): 34–51, 2015.

[7] A. Ahmadi-Javid, P. Seyedi, and S. S. Syam. A survey of healthcare facility location. *Computers & Operations Research*, 79:223–263, 2017.

[8] A. Ansari, C. F. Mela, and S. A. Neslin. Customer channel migration. *Journal of marketing research*, 45(1):60–76, 2008.

[9] A. N. Arslan, W. Klibi, and B. Montreuil. Distribution network deployment for omnichannel retailing. *European Journal of Operational Research*, 294(3):1042–1058, 2021.

[10] A. Atamtürk and V. Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Operations Research Letters*, 36(5):618–622, 2008.

[11] O. Baron, O. Berman, and D. Krass. Facility location with stochastic demand and constraints on waiting time. *Manufacturing & Service Operations Management*, 10(3): 484–505, 2008.

[12] K. Baumgartner, A. Fuetterer, and U. W. Thonemann. Supply chain design considering economies of scale and transport frequencies. *European Journal of Operational Research*, 218(3):789–800, 2012.

[13] A. Bayram and B. Cesaret. Order fulfillment policies for ship-from-store implementation in omni-channel retailing. *European Journal of Operational Research*, 294(3): 987–1002, 2020.

[14] D. R. Bell, S. Gallino, and A. Moreno. How to win in an omnichannel world. *MIT Sloan Management Review*, 56(1):45, 2014.

[15] O. Berman and Z. Drezner. The multiple server location problem. *Journal of the Operational Research Society*, 58(1):91–99, 2007.

[16] O. Berman and D. Krass. Facility location problems with stochastic demands and congestion. In Z. Drezner and H. W. Hamacher, editors, *Facility Location: Applications and Theory*, pages 329–369. Springer-Verlag, New York, 2001.

[17] O. Berman and D. Krass. Stochastic location models with congestion. In G. Laporte,

S. Nickel, and F. Saldanha da Gama, editors, *Location Science*. Springer, Cham, 2nd edition, 2020.

[18] B. Boffey, R. Galvao, and L. Espefo. A review of congestion models in the location of facilities with immobile servers. *European Journal of Operational Research*, 178: 643–662, 2007.

[19] E. Brynjolfsson, Y. J. Hu, and M. S. Rahman. Competing in the age of omnichannel retailing. *MIT Sloan Management Review*, 54(4):23–29, 2013.

[20] S. Burer and A. N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[21] Businesswire. JDA and PWC Omnichannel survey. `https://www.businesswire.com/news/home/20170223005209/en/`, 2017. [Online; Accessed: 2021-05-31].

[22] Y.-J. Cai and C. K. Lo. Omni-channel management in the new retailing era: A systematic review and future research agenda. *International Journal of Production Economics*, 229:107729, 2020.

[23] J. Cao, K. C. So, and S. Yin. Impact of an "online-to-store" channel on demand allocation, pricing and profitability. *European Journal of Operational Research*, 248 (1):234–245, 2016.

[24] A. I. Chen and S. C. Graves. Item aggregation and column generation for online-retail inventory placement. *Manufacturing & Service Operations Management*, 2020. https://doi.org/10.1287/msom.2020.0867.

[25] S. Chopra. The evolution of omni-channel retailing and its impact on supply chains. *Transportation research procedia*, 30:4–13, 2018.

[26] R. Church and C. ReVelle. The maximal covering location problem. *Regional Science Association Papers*, 32:101–118, 1974.

[27] CNBC. Best Buy Fulfillment Centers. https://www.cnbc.com/2019/08/28/best-buy-revs-up-supply-chain-ahead-of-the-holiday-season.html, 2019. [Online; accessed 2021-05-31].

[28] M. A. Cohen and S. Moon. An integrated plant loading model with economies of scale and scope. *European Journal of Operational Research*, 50(3):266–279, 1991.

[29] I. Contreras and C. Ortiz-Astorquiza. Hierarchical facility location problems. In *Laporte G., Nickel S., Saldanha da Gama F. (eds) Location Science.*, pages 365–389. Springer, Cham, 2nd edition, 2019.

[30] I. Contreras, J.-F. Cordeau, and G. Laporte. The dynamic uncapacitated hub location problem. *Transportation Science*, 45(1):18–32, 2011.

[31] J.-F. Cordeau, W. Klibi, and S. Nickel. Logistics network design. In T. Crainic, M. Gendreau, and B. Gendron, editors, *Network Design with Applications to Transportation and Logistics*. Springer, Cham, 2021.

[32] C. D'Ambrosio and A. Lodi. Mixed integer nonlinear programming tools: an updated practical overview. *Annals of Operations Research*, 204:301–320, 2013.

[33] M. S. Daskin and K. L. Maass. The p-median problem. In *Laporte G., Nickel S., Saldanha da Gama F. (eds) Location science.*, pages 21–45. Springer, Cham, 2015.

[34] M. S. Daskin, C. R. Coullard, and Z.-J. M. Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of operations research*, 110(1):83–106, 2002.

[35] DigitalCommerce360. Amazon Prime Membership. https://www.digitalcommerce360.com/article/amazon-prime-membership/, 2021. Accessed: 2021-08-03.

[36] Z. Drezner and H. W. Hamacher. *Facility Location: Applications and Theory*. Springerr-Verlag, New York, NY, 2002.

[37] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Jünger M., Reinelt G., Rinaldi G. (eds) Combinatorial Optimization—Eureka, You Shrink!. Lecture Notes in Computer Science*, volume 2570, pages 11–26. Springer, Berlin, Heidelberg, 2003.

[38] M. Efroymson and T. Ray. A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368, 1966.

[39] S. Elhedhli. Service system design with immobile servers, stochastic demand, and congestion. *Manufacturing & Service Operations Management*, 8(1):92–97, 2006.

[40] S. Elhedhli and R. Merrick. Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17(5):370–379, 2012.

[41] J. E. Falk and R. M. Soland. An algorithm for separable nonconvex programming problems. *Management science*, 15(9):550–569, 1969.

[42] R. Z. Farahani and M. Hekmatfar. *Facility location: concepts, models, algorithms and case studies*. Contributions to management science, Heidelberg: Physica-Verlag, 2009.

[43] R. Z. Farahani, M. Hekmatfar, B. Fahimnia, and N. Kazemzadeh. Hierarchical facility location problem: Models, classifications, techniques, and applications. *Computers & Industrial Engineering*, 68:104–117, 2014.

[44] E. Feldman, F. Lehrer, and T. Ray. Warehouse location under continuous economies of scale. *Management Science*, 12(9):670–684, 1966.

[45] E. Fernández and M. Landete. Fixed-charge facility location problems. In *Laporte G., Nickel S., Saldanha da Gama F. (eds) Location Science.*, pages 47–77. Springer, Cham, 2nd edition, 2015.

[46] M. Fischetti, I. Ljubić, and M. Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569, 2016.

[47] M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management science*, 63(7):2146–2162, 2017.

[48] Forbes. Black Friday's Deal Pit. `https://www.forbes.com/sites/petercohan/2020/11/27/black-friday-deals-pit-walmart-and-best-buy-against-amazon/?sh=30c7ac4f64a0`, 2020. [Online; accessed 2021-05-31].

[49] B. Fortz. Location problems in telecommunications. In *Laporte G., Nickel S., Saldanha da Gama F. (eds) Location Science.*, pages 537–554. Springer, Cham, 2015.

[50] Freightos. AMAZON FULFILLMENT LOCATIONS. `https://www.freightos.com/freight-resources/amazon-fba-fulfillment-warehouse-locations/`, 2021. Accessed: 2021-08-03.

[51] S. L. Gadegaard, A. Klose, and L. R. Nielsen. An improved cut-and-solve algorithm for the single-source capacitated facility location problem. *EURO Journal on Computational Optimization*, 6(1):1–27, 2018.

[52] S. Gallino and A. Moreno. Integration of online and offline channels in retail: The impact of sharing reliable inventory availability information. *Management Science*, 60 (6):1434–1451, 2014.

[53] F. Gao and X. Su. Omnichannel retail operations with buy-online-and-pick-up-in-store. *Management Science*, 63(8):2478–2492, 2017.

[54] F. Gao, V. V. Agrawal, and S. Cui. The effect of multichannel and omnichannel retailing on physical stores. *Management Science*, 2021.

[55] S. García and A. Marín. Covering location problems. In *Laporte G., Nickel S., Saldanha da Gama F. (eds) Location Science.*, pages 93–114. Springer, Cham, 2015.

[56] S. Gokarn and T. S. Kuthambalayan. Creating sustainable fresh produce supply chains by managing uncertainties. *Journal of cleaner production*, 207:908–919, 2019.

[57] A. Govindarajan, A. Sinha, and J. Uichanco. Joint inventory and fulfillment decisions for omnichannel retail networks. *Naval Research Logistics (NRL)*, 2018.

[58] C. M. Grinstead and J. L. Snell. *Introduction to probability.* American Mathematical Soc., 2012.

[59] T. A. Grossman and M. L. Brandeau. Optimal pricing for service facilities with self-optimizing customers. *European Journal of Operational Research*, 141(1):39–57, 2002.

[60] M. T. Hajiaghayi, M. Mahdian, and V. S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1):42–47, 2003.

[61] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

[62] J. Harkness and C. ReVelle. Facility location with increasing production costs. *European Journal of Operational Research*, 145(1):1–13, 2003.

[63] P. Harsha, S. Subramanian, and J. Uichanco. Dynamic pricing of omnichannel inventories: honorable mention—2017 m&som practice-based research competition. *Manufacturing & Service Operations Management*, 21(1):47–65, 2019.

[64] K. L. Hoffman. A method for globally minimizing concave functions over convex sets. *Mathematical Programming*, 20(1):22–32, 1981.

[65] C. Hu, V. Pozdnyakov, and J. Yan. Density and distribution evaluation for convolution of independent gamma variables. *Computational Statistics*, 35(1):327–342, 2020.

[66] M. Hu, X. Xu, W. Xue, and Y. Yang. Demand pooling in omnichannel operations. *Management Science*, 2021. https://doi.org/10.1287/mnsc.2021.3964.

[67] A. Hübner, A. Holzapfel, and H. Kuhn. Distribution systems in omni-channel retailing. *Business Research*, 9(2):255–296, 2016.

[68] A. Hübner, H. Kuhn, and J. Wollenburg. Last mile fulfilment and distribution in omni-channel grocery retailing: A strategic planning framework. *International Journal of Retail & Distribution Management*, 44(3):228–247, 2016.

[69] A. Hübner, J. Wollenburg, and A. Holzapfel. Retail logistics in the transition from multi-channel to omni-channel. *International Journal of Physical Distribution & Logistics Management*, 46(6/7):562–583, 2016.

[70] IndigoDigital. Walmart Ecommerce Sucess. `https://www.indigo9digital.com/blog/4-secrets-to-walmarts-ecommerce-sucess`, 2020. [Online; accessed 2021-05-31].

[71] R. Ishfaq and N. Bajwa. Profitability of online order fulfillment in multi-channel retailing. *European Journal of Operational Research*, 272(3):1028–1040, 2019.

[72] R. Ishfaq, C. C. Defee, B. J. Gibson, and U. Raja. Realignment of the physical distribution process in omni-channel fulfillment. *International Journal of Physical Distribution & Logistics Management*, 46(6/7):543–561, 2016.

[73] S. Jasin, A. Sinha, and J. Uichanco. Omnichannel operations: Challenges, opportunities, and models. In *Gallino S., Moreno A. (eds) Operations in an Omnichannel World. Springer Series in Supply Chain Management*, volume 8, pages 15–34. Springer, Cham, 2019.

[74] S. D. Jena, J.-F. Cordeau, and B. Gendron. Dynamic facility location with generalized modular capacities. *Transportation Science*, 49(3):484–499, 2015.

[75] L. Kaufman, M. V. Eede, and P. Hansen. A plant and warehouse location problem. *Journal of the Operational Research Society*, 28(3):547–554, 1977.

[76] D. L. Kelly and B. M. Khumawala. Capacitated warehouse location with concave costs. *Journal of the Operational Research Society*, 33(9):817–826, 1982.

[77] R. Kramer, M. Iori, and T. Vidal. Mathematical models and search algorithms for the capacitated p-center problem. *INFORMS Journal on Computing*, 32(2):444–460, 2020.

[78] M. Kubo and H. Kasugai. A Lagrangean approach to the facility location problem with concave costs. *Journal of the Operations Research Society of Japan*, 34(2):125–136, 1991.

[79] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management science*, 9(4):643–666, 1963.

[80] G. Laporte, S. Nickel, and F. S. da Gama. *Location Science*. Springer, Cham, 2nd edition, 2019.

[81] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.

[82] J. Lee and S. Leyffer. *Mixed Integer Nonlinear Programming*. Springer, New York, 2012.

[83] J.-R. Lin, L. K. Nozick, and M. A. Turnquist. Strategic design of distribution systems with economies of scale in transportation. *Annals of Operations Research*, 144(1): 161–180, 2006.

[84] M.-L. Liu, N. V. Sahinidis, and J. P. Shectman. Planning of chemical process networks via global concave minimization. In *Global Optimization in Engineering Design*, pages 195–230. Springer, Boston, 1996.

[85] D. Lu, F. Gzara, and S. Elhedhli. Facility location with economies and diseconomies of scale: models and column generation heuristics. *IIE Transactions*, 46(6):585–600, 2014.

[86] P. Mandal, P. Basu, and K. Saha. Forays into omnichannel: An online retailer's strategies for managing product returns. *European Journal of Operational Research*, 292(2):633–651, 2021.

[87] V. I. Manousiouthakis, N. Thomas, and A. M. Justanieah. On a finite branch and bound algorithm for the global minimization of a concave power law over a polytope. *Journal of Optimization Theory and Applications*, 151(1):121–134, 2011.

[88] V. Marianov and D. Serra. Probabilistic, maximal covering location—allocation models for congested systems. *Journal of Regional Science*, 38(3):401–424, 1998.

[89] V. Marianov and D. Serra. Hierarchical location–allocation models for congested systems. *European Journal of Operational Research*, 135(1):195–208, 2001.

[90] V. Marianov and D. Serra. Location–allocation of multiple-server service centers with constrained queues or waiting times. *Annals of Operations Research*, 111(1-4):35–50, 2002.

[91] A. Mathai. Storage capacity of a dam with gamma type inputs. *Annals of the Institute of Statistical Mathematics*, 34(3):591–597, 1982.

[92] V. R. Máximo, M. C. Nascimento, and A. C. Carvalho. Intelligent-guided adaptive search for the maximum covering location problem. *Computers and Operations Research*, 78:129–137, 2017.

[93] M. Melacini, S. Perotti, M. Rasini, and E. Tappia. E-fulfilment and distribution in omni-channel retailing: a systematic literature review. *International Journal of Physical Distribution & Logistics Management*, 48(4):391–414, 2018.

[94] M. Melo, S. Nickel, and F. S. da Gama. Facility location and supply chain management – a review. *European Journal of Operational Research*, 196(2):401–412, 2009.

[95] M. T. Melo, S. Nickel, and F. Saldanha-Da-Gama. Facility location and supply chain management–a review. *European journal of operational research*, 196(2):401–412, 2009.

[96] S. Nahmias and T. L. Olsen. *Production and operations analysis.* Waveland Press, 2015.

[97] OliverWyman. Oliver Wyman. `https://www.oliverwyman.com/our-expertise/insights/2018/may/supply-chain-optimization--levers-for-rapid-ebitda.html`. : 2021-07-05.

[98] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. Formulations and approximation algorithms for multilevel uncapacitated facility location. *INFORMS Journal on Computing*, 29(4):767–779, 2017.

[99] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. Multi-level facility location problems. *European Journal of Operational Research*, 267(3):791–805, 2018.

[100] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. An exact algorithm for multi-level uncapacitated facility location. *Transportation Science*, 53(4):1085–1106, 2019.

[101] J. Park, I. Dayarian, and B. Montreuil. Showcasing optimization in omnichannel retailing. *European Journal of Operational Research*, 294(3):895–905, 2020.

[102] H. Pirkul and V. Jayaraman. Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Science*, 30(4):291–302, 1996.

[103] A. Polus. A study of travel time and reliability on arterial routes. *Transportation*, 8 (2):141–151, 1979.

[104] Radial. Best Buy Omnichannel Strategy. https://www.radial.com/insights/best-buy-omnichannel-strategy-model-other-brick-and-mortar-retailers, 2016. [Online; accessed 2021-05-31].

[105] D. Rigby. The future of shopping. *Harvard business review*, 89(12):65–76, 2011.

[106] H. S. Ryoo and N. V. Sahinidis. A branch-and-reduce approach to global optimization. *Journal of Global Optimization*, 8(2):107–138, 1996.

[107] K. Saha and S. Bhattacharya. Buy online and pick up in-store: Implications for the store inventory. *European Journal of Operational Research*, 294(3):906–921, 2020.

[108] G. Şahin and H. Süral. A review of hierarchical facility location models. *Computers & Operations Research*, 34(8):2310–2331, 2007.

[109] A. Saif and S. Elhedhli. A lagrangian heuristic for concave cost facility location problems: the plant location and technology acquisition problem. *Optimization Letters*, 10 (5):1087–1100, 2016.

[110] F. Saldanha-da Gama and S. Nickel. Multi-period facility location. In G. Laporte, S. Nickel, and F. Saldanha-da Gama, editors, *Location Science*. Springer, Cham, 2019.

[111] F. S. Salman and E. Yücel. Emergency facility location under random network damage: Insights from the istanbul case. *Computers and Operations Research*, 62:266–281, 2015.

[112] Z.-J. M. Shen and M. S. Daskin. Trade-offs between customer service and cost in integrated supply chain design. *Manufacturing & service operations management*, 7 (3):188–207, 2005.

[113] Z.-J. M. Shen, C. Coullard, and M. S. Daskin. A joint location-inventory model. *Transportation science*, 37(1):40–55, 2003.

[114] Shopify. Future of Commerce. https://www.shopify.com/future-of-commerce/2021, 2021. Accessed: 2021-08-06.

[115] R. M. Soland. Optimal facility location with concave costs. *Operations Research*, 22 (2):373–382, 1974.

[116] SupplychainDive. Best Buy Store Fulfillment. `https://www.supplychaindive.com/news/best-buy-limit-SKU-store-centric-fullfillment-ship/589697/`, 2020. [Online; accessed 2021-05-31].

[117] Target. Target's Online Commitments. `https://corporate.target.com/article/2017/02/financial-community-meeting/`, 2017. [Online; Accessed: 2021-03-25].

[118] S. Torabi, E. Hassini, and M. Jeihoonian. Fulfillment source allocation, inventory transshipment, and customer order transfer in e-tailing. *Transportation Research Part E: Logistics and Transportation Review*, 79:128–144, 2015.

[119] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

[120] H. Tuy, T. Thieu, and N. Q. Thai. A conical algorithm for globally minimizing a concave function over a closed convex set. *Mathematics of Operations Research*, 10(3): 498–514, 1985.

[121] P. C. Verhoef, P. K. Kannan, and J. J. Inman. From multi-channel retailing to omni-channel retailing: introduction to the special issue on multi-channel retailing. *Journal of retailing*, 91(2):174–181, 2015.

[122] N. Vidyarthi and S. Jayaswal. Efficient solution of a class of location–allocation problems with stochastic demand and congestion. *Computers & Operations Research*, 48: 20–30, 2014.

[123] N. Vidyarthi, E. Çelebi, S. Elhedhli, and E. Jewkes. Integrated production-inventory-distribution system design with risk pooling: model formulation and heuristic solution. *Transportation Science*, 41(3):392–408, 2007.

[124] N. Vidyarthi, S. Elhedhli, and E. Jewkes. Response time reduction in make-to-order and assemble-to-order supply chain design. *IIE transactions*, 41(5):448–466, 2009.

[125] Walmart. Walmart History. `https://corporate.walmart.com/our-story/our-history#timeline/`, 2019. [Online; Accessed: 2021-03-25].

[126] Q. Wang, R. Batta, and C. M. Rump. Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of operations Research*, 111 (1-4):17–34, 2002.

[127] D. Yang and X. Zhang. Omnichannel operations with ship-from-store. *Operations Research Letters*, 48(3):257–261, 2020.

[128] W. I. Zangwill. Minimum concave cost flows in certain networks. *Management Science*, 14(7):429–450, 1968.

Appendices

Appendix A

Detailed Tables of Computational Results: Chapter 2

Algorithm 3: Branch-and-Bound Algorithm

Input: Problem P
Initialize r=0; ϵ (small value); endFlag = 0; $\pi = \{\}$; $N^s = N^0$
Find $LB(N^0)$ and $UB(N^0)$ by solving $\mathrm{P}_{\mathrm{L}}^{0}$ at node N^0.
BestLB = $LB(N^0)$; BestUB = $UB(N^0)$; $\hat{x}^B = \hat{x}^0$; OptGap = $\frac{\text{(BestUB - BestLB)}}{\text{BestUB}}$
if (OptGap $\leq \epsilon$) **then** endFlag = 1
while (endFlag == 0) **do**
  Branch N^s to create N^{2r+1} and N^{2r+2}
  Process Left Node N^{2r+1}:
  Find $LB(N^{2r+1})$ and $UB(N^{2r+1})$ by solving $\mathrm{P}_{\mathrm{L}}^{2r+1}$
  if ($UB(N^{2r+1}) <$ BestUB) **then** BestUB = $UB(N^{2r+1})$ and $\hat{x}^B = \hat{x}^{2r+1}$
  if ($LB(N^{2r+1}) <$ BestUB) **then** update $\pi = \pi \cup \{N^{2r+1}\}$
  Process Right Node N^{2r+2}:
  Find $LB(N^{2r+2})$ and $UB(N^{2r+2})$ by solving $\mathrm{P}_{\mathrm{L}}^{2r+2}$
  if ($UB(N^{2r+2}) <$ BestUB) **then** BestUB = $UB(N^{2r+2})$ and $\hat{x}^B = \hat{x}^{2r+2}$
  if ($LB(N^{2r+2}) <$ BestUB) **then** update $\pi = \pi \cup \{N^{2r+2}\}$
  Select node N^s from the set π
  Update: $\pi = \pi \setminus \{N^s\}$, BestLB = $LB(N^s)$, OptGap and r = r + 1
  if (OptGap $\leq \epsilon$) **then** endFlag = 1
end while
Output: Optimal Solution Value = BestUB, and solution vector = $\hat{x}^B$.

Table A.1: Comparison of Branching Strategies for MINLP Formulation

			Gap(%)		Time(s)			Efficient
Scenario	Instance	FFC(%)[1]	SP_g	MAP_g	SP_t	MAP_t	%Reduction	Strategy
BC	9	33	0.13	0.15	time	time	-	SP (gap)
	10	35	0.10	0.15	time	time	-	SP (gap)
	11	32	0.13	0.14	time	time	-	SP (gap)
	12	27	0.11	0.12	time	time	-	SP (gap)
	13	30	0.23	0.19	time	time	-	MAP (gap)
	14	37	0.20	0.18	time	time	-	MAP (gap)
Average		**32**	**0.15**	**0.16**	**time**	**time**	**-**	**4/2**
DFC	9	65	0.10	0.10	18,293	14,813	19	MAP (time)
	10	66	0.10	0.10	6,383	2,199	66	MAP (time)
	11	66	0.10	0.10	6,755	7,033	-4	SP (time)
	12	61	0.09	0.10	2,108	1,643	22	MAP (time)
	13	65	0.13	0.13	time	time	-	MAP (gap)
	14	68	0.13	0.10	time	33,481	61	MAP (time,gap)
Average		**65**	**0.11**	**0.10**	**34409**	**24271**	**29**	**1/5**
DVC	9	14	0.11	0.13	time	time	-	SP (gap)
	10	15	0.10	0.14	44,740	time	-93	SP (time,gap)
	11	13	0.16	0.15	time	time	-	MAP (gap)
	12	14	0.13	0.16	time	time	-	SP (gap)
	13	13	0.35	0.33	time	time	-	MAP (gap)
	14	15	0.34	0.23	time	time	-	MAP (gap)
Average		**14**	**0.20**	**0.19**	**79476**	**time**	**-9**	**3/3**
EC	9	30	0.10	0.10	13,135	6,706	49	MAP (time)
	10	28	0.10	0.10	5,565	4,192	25	MAP (time)
	11	30	0.10	0.10	22,685	43,119	-90	SP (time)
	12	32	0.12	0.18	time	time	-	SP (gap)
	13	25	0.24	0.16	time	time	-	MAP (gap)
	14	28	0.22	0.19	time	time	-	MAP (gap)
Average		**29**	**0.15**	**0.14**	**50115**	**52208**	**-4**	**2/4**
TC	9	41	0.39	0.37	time	time	-	MAP (gap)
	10	34	0.32	0.28	time	time	-	MAP (gap)
	11	39	0.48	0.48	time	time	-	MAP (gap)
	12	38	0.40	0.40	time	time	-	MAP (gap)
	13	40	0.35	0.36	time	time	-	SP (gap)
	14	38	0.46	0.47	time	time	-	SP (gap)
Average		**38**	**0.40**	**0.39**	**time**	**time**	**-**	**2/4**

[1] FFC: Facility fixed cost is expressed as percentage of the objective function value.

Table A.2: Comparison of Branching Strategies for NLP Formulation

			Gap(%)		Time(s)			Efficient
Scenario	Instance	FFC(%)[1]	SP_g	MAP_g	SP_t	MAP_t	%Reduction	Strategy
BC	9	33	0.13	0.14	time	time	-	SP (gap)
	10	33	0.10	0.14	11,928	time	-624	SP (time,gap)
	11	32	0.10	0.13	time	time	-	SP (gap)
	12	27	0.10	0.12	25,381	time	-240	SP (time,gap)
	13	30	0.11	0.14	time	time	-	SP (gap)
	14	37	0.15	0.18	time	time	-	SP (gap)
Average		**32**	**0.11**	**0.14**	**63697**	**time**	**-36**	**6/0**
DFC	9	65	0.10	0.10	11,034	2,191	80	MAP (time)
	10	66	0.10	0.10	814	171	79	MAP (time)
	11	66	0.10	0.10	1,109	842	24	MAP (time)
	12	61	0.10	0.10	714	341	52	MAP (time)
	13	65	0.10	0.09	22,154	2,328	89	MAP (time)
	14	68	0.10	0.10	36,512	4,318	88	MAP (time)
Average		**65**	**0.10**	**0.10**	**12056**	**1699**	**-86**	**0/6**
DVC	9	14	0.10	0.12	28,943	time	-199	SP (time,gap)
	10	15	0.10	0.11	8,081	time	-969	SP (time,gap)
	11	13	0.11	0.14	time	time	-	SP (gap)
	12	14	0.10	0.13	72,368	time	-19	SP (time,gap)
	13	13	0.24	0.19	time	time	-	MAP (gap)
	14	15	0.14	0.16	time	time	-	SP (gap)
Average		**14**	**0.13**	**0.14**	**61433**	**time**	**-41**	**5/1**
EC	9	30	0.10	0.10	2,821	13,479	-378	SP (time)
	10	28	0.10	0.09	7,712	11,169	-45	SP (time)
	11	30	0.10	0.10	3,460	3,536	-2	SP (time)
	12	32	0.10	0.14	time	time	-	SP (gap)
	13	25	0.10	0.11	27,611	time	-213	SP (time,gap)
	14	28	0.10	0.10	71,533	56,085	22	MAP (time)
Average		**29**	**0.10**	**0.11**	**33257**	**42846**	**-29**	**5/1**
TC	9	41	0.35	0.33	time	time	-	MAP (gap)
	10	34	0.13	0.14	time	time	-	SP (gap)
	11	39	0.37	0.38	time	time	-	SP (gap)
	12	38	0.29	0.25	time	time	-	MAP (gap)
	13	40	0.27	0.27	time	time	-	SP (gap)
	14	38	0.38	0.38	time	time	-	SP (gap)
Average		**38**	**0.30**	**0.29**	**time**	**time**	**-**	**4/2**

[1] FFC: Facility fixed cost is expressed as percentage of the objective function value.

Table A.3: Performance of Preprocessing on MINLP Formulation

			Gap(%)		Time(s)			Preprocessing
Scenario	Instance	FFC(%)[1]	WoP_g	WP_g	WoP_t	WP_t	%Reduction	Without/With
BC	9	33	0.13	0.12	time	time	-	With (gap)
	10	35	0.10	0.09	time	39,257	55	With (time,gap)
	11	32	0.13	0.13	time	time	-	With (gap)
	12	27	0.11	0.10	time	46,037	47	With (time,gap)
	13	30	0.23	0.22	time	time	-	With (gap)
	14	37	0.20	0.17	time	time	-	With (gap)
Average		**32**	**0.15**	**0.14**	**time**	**71834**	**17**	**0/6**
DFC	9	65	0.10	0.10	14,813	1,745	88	With (time)
	10	66	0.10	0.10	2,199	737	66	With (time)
	11	66	0.10	0.09	7,033	2,176	69	With (time)
	12	61	0.10	0.10	1,643	1,761	-7	Without (time)
	13	65	0.13	0.10	time	35,000	59	With (time,gap)
	14	68	0.10	0.10	33,481	11,662	65	With (time)
Average		**65**	**0.10**	**0.10**	**24271**	**8847**	**64**	**1/5**
DVC	9	14	0.11	0.11	time	time	-	Without (gap)
	10	15	0.10	0.10	44,740	57,588	-29	Without (time)
	11	13	0.16	0.16	time	time	-	With (gap)
	12	14	0.13	0.13	time	time	-	With (gap)
	13	13	0.35	0.34	time	time	-	With (gap)
	14	15	0.34	0.34	time	time	-	Without (gap)
Average		**14**	**0.20**	**0.20**	**79476**	**81610**	**-3**	**3/3**
EC	9	30	0.10	0.09	13135	6007	54	With (time)
	10	28	0.10	0.08	5,565	4,788	14	With (time)
	11	30	0.10	0.10	22,685	13,201	42	With (time)
	12	32	0.12	0.12	time	time	-	With (gap)
	13	25	0.24	0.16	time	time	-	With (gap)
	14	28	0.22	0.15	time	time	-	With (gap)
Average		**29**	**0.15**	**0.12**	**50115**	**47203**	**6**	**0/6**
TC	9	41	0.39	0.36	time	time	-	With (gap)
	10	34	0.32	0.26	time	time	-	With (gap)
	11	39	0.48	0.46	time	time	-	With (gap)
	12	38	0.40	0.40	time	time	-	With (gap)
	13	40	0.35	0.33	time	time	-	With (gap)
	14	38	0.46	0.45	time	time	-	With (gap)
Average		**38**	**0.40**	**0.38**	**time**	**time**	**-**	**0/6**

[1] FFC: Facility fixed cost is expressed as percentage of the objective function value.

Table A.4: Performance of Preprocessing on NLP Formulation

			Gap(%)		Time(s)			Preprocessing
Scenario	Instance	FFC(%)[1]	$\mathrm{WoP_g}$	$\mathrm{WP_g}$	$\mathrm{WoP_t}$	$\mathrm{WP_t}$	%Reduction	Without/With
BC	9	33	0.13	0.11	time	time	-	With (gap)
	10	35	0.10	0.10	11,928	7,073	41	With (time)
	11	32	0.10	0.10	85,670	20,896	76	With (time)
	12	27	0.10	0.10	25,381	13,085	48	With (time)
	13	30	0.11	0.12	time	time	-	Without (gap)
	14	37	0.15	0.13	time	time	-	With (gap)
Average		**32**	**0.11**	**0.11**	**63697**	**50043**	**21**	**1/5**
DFC	9	65	0.10	0.10	2,191	1,431	35	With (time)
	10	66	0.10	0.10	171	625	-265	Without (time)
	11	66	0.10	0.10	842	1,542	-83	Without (time)
	12	61	0.10	0.10	341	1,031	-202	Without (time)
	13	65	0.09	0.10	2,328	7,397	-218	Without (time)
	14	68	0.10	0.09	4,318	2,544	41	With (time)
Average		**65**	**0.10**	**0.10**	**1699**	**2428**	**-43**	**4/2**
DVC	9	14	0.10	0.10	28,943	74,333	-157	Without (time)
	10	15	0.10	0.10	8,081	11,358	-41	Without (time)
	11	13	0.11	0.11	time	time	-	Without (gap)
	12	14	0.10	0.14	72,368	time	-19	Without (time,gap)
	13	13	0.24	0.26	time	time	-	Without (gap)
	14	15	0.14	0.15	time	time	-	Without (gap)
Average		**14**	**0.13**	**0.14**	**61433**	**71883**	**-17**	**6/0**
EC	9	30	0.10	0.08	2,821	2,873	-2	Without (time)
	10	28	0.10	0.07	7,712	7,661	1	With (time)
	11	30	0.10	0.10	3,460	4,213	-22	Without (time)
	12	32	0.10	0.11	time	time	-	Without (gap)
	13	25	0.10	0.10	27,611	18,919	31	With (time)
	14	28	0.10	0.10	71,533	37,979	47	With (time)
Average		**29**	**0.10**	**0.10**	**33257**	**26341**	**21**	**3/3**
TC	9	41	0.35	0.32	time	time	-	With (gap)
	10	34	0.13	0.12	time	time	-	With (gap)
	11	39	0.37	0.35	time	time	-	With (gap)
	12	38	0.29	0.29	time	time	-	With (gap)
	13	40	0.27	0.27	time	time	-	Without (gap)
	14	38	0.38	0.38	time	time	-	Without (gap)
Average		**38**	**0.30**	**0.30**	**time**	**time**	**-**	**2/4**

[1] FFC: Facility fixed cost is expressed as percentage of the objective function value.

Table A.5: Computational Results for MINLP and NLP under Balanced Cost Scenario

	Root Node Performance				Performance at Termination								
	Gap(%)		Time(s)		Gap(%)		Time(s)			Efficient	Cost(%)[1]		
Instance	MINLP_{rg}	NLP_{rg}	MINLP_{rt}	NLP_{rt}	MINLP_{g}	NLP_{g}	MINLP_{t}	NLP_{t}	%Reduction	Formulation	FFC	FVC	TrC
1	1.43	2.38	< 1	< 1	0.18	0.20	39	35	10	NLP (time)	38	29	33
2	2.29	3.57	< 1	< 1	0.20	0.20	120	65	46	NLP (time)	32	31	37
3	0.66	2.61	< 1	< 1	0.15	0.20	70	87	-24	MINLP (time)	39	26	35
4	0.49	3.15	< 1	< 1	0.20	0.19	133	124	7	NLP (time)	31	28	41
5	1.24	3.79	1	< 1	0.18	0.19	422	346	18	NLP (time)	39	25	36
6	0.64	1.51	< 1	< 1	0.20	0.19	347	341	2	NLP (time)	35	31	34
7	0.87	2.31	< 1	< 1	0.20	0.20	1,438	687	52	NLP (time)	33	29	38
8	1.01	1.65	2	< 1	0.20	0.20	548	494	10	NLP (time)	25	35	40
9	0.91	2.02	4	< 1	0.20	0.20	9,897	5,963	40	NLP (time)	34	26	39
10	0.69	2.49	9	< 1	0.20	0.20	7,089	3,559	50	NLP (time)	33	31	36
11	0.47	2.22	4	< 1	0.20	0.20	1,886	1,527	19	NLP (time)	32	25	43
12	1.54	3.40	5	< 1	0.20	0.20	11,385	4,046	64	NLP (time)	27	33	40
13	1.07	2.80	150	< 1	0.21	0.20	time	28,892	67	NLP (time,gap)	30	32	38
14	0.51	1.90	11	< 1	0.20	0.20	54,642	7,017	87	NLP (time)	35	25	40
15	0.69	2.12	14	< 1	0.30	0.22	time	time	-	NLP (gap)	37	27	36
16	0.73	1.62	6	< 1	0.20	0.20	78,960	9,530	88	NLP (time)	30	26	44
17	0.68	1.73	96	< 1	0.38	0.20	time	55,481	36	NLP (time,gap)	28	31	41
18	0.55	1.45	80	< 1	0.28	0.20	time	31,124	64	NLP (time,gap)	36	25	38
19	0.67	2.03	57	< 1	0.36	0.26	time	time	-	NLP (gap)	32	29	39
20	0.68	1.57	93	< 1	0.60	0.46	time	time	-	NLP (gap)	29	29	42
Average	**0.89**	**2.32**	**27**	**<1**	**0.24**	**0.22**	**34269**	**20426**	**40**		**33**	**29**	**39**

[1] FFC: Facility fixed cost; FVC: Facility variable cost; TrC: Transportation cost. All these costs are expressed as percentage of the objective function value.

Table A.6: Computational Results for MINLP and NLP under Dominant Fixed Cost Scenario

	Root Node Performance				Performance at Termination								
	Gap(%)		Time(s)		Gap(%)		Time(s)			Efficient	Cost(%)[1]		
Instance	MINLP_{rg}	NLP_{rg}	MINLP_{rt}	NLP_{rt}	MINLP_{g}	NLP_{g}	MINLP_{t}	NLP_{t}	%Reduction	Formulation	FFC	FVC	TrC
1	0.78	1.92	< 1	< 1	0.18	0.18	15	14	7	NLP (time)	68	16	15
2	0.67	4.11	< 1	< 1	0.20	0.19	17	17	0	MINLP (time)	65	16	19
3	0.34	4.02	< 1	< 1	0.20	0.20	25	27	-8	MINLP (time)	73	11	15
4	0.42	1.25	< 1	< 1	0.20	0.20	33	34	-3	MINLP (time)	61	20	19
5	0.44	1.46	< 1	< 1	0.20	0.19	157	146	7	NLP (time)	68	15	18
6	0.35	0.35	< 1	< 1	0.20	0.20	223	223	0	NLP (time)	65	17	17
7	0.41	0.41	< 1	< 1	0.19	0.19	181	177	2	NLP (time)	68	14	18
8	0.63	1.88	2	< 1	0.19	0.19	383	358	7	NLP (time)	55	19	26
9	0.29	0.40	< 1	< 1	0.20	0.20	653	658	-1	MINLP (time)	65	16	19
10	0.22	0.36	< 1	< 1	0.19	0.20	461	458	1	NLP (time)	66	17	17
11	0.46	1.55	1	< 1	0.20	0.20	885	808	9	NLP (time)	66	13	21
12	0.68	2.13	2	< 1	0.20	0.20	673	666	1	NLP (time)	61	17	22
13	0.23	1.20	4	< 1	0.20	0.20	3,857	3,339	13	NLP (time)	65	16	19
14	0.46	2.78	5	< 1	0.20	0.20	1,832	1,666	9	NLP (time)	68	12	20
15	0.35	3.10	6	< 1	0.20	0.20	7,089	2,648	63	NLP (time)	71	13	16
16	0.36	2.02	10	< 1	0.20	0.20	10,410	3,887	63	NLP (time)	65	14	21
17	0.28	1.18	30	< 1	0.20	0.20	18,107	7,509	59	NLP (time)	63	16	21
18	0.27	0.80	8	< 1	0.21	0.19	time	19,725	77	NLP (time,gap)	67	16	18
19	0.31	2.04	11	< 1	0.20	0.20	13,732	11,969	13	NLP (time)	61	18	21
20	0.31	1.59	65	< 1	0.20	0.20	35,397	19,297	45	NLP (time)	64	16	21
Average	**0.41**	**1.73**	**8**	**<1**	**0.20**	**0.20**	**9027**	**3681**	**59**		**65**	**16**	**19**

[1] FFC: Facility fixed cost; FVC: Facility variable cost; TrC: Transportation cost. All these costs are expressed as percentage of the objective function value.

Table A.7: Computational Results for MINLP and NLP under Dominant Variable Cost Scenario

	Root Node Performance				Performance at Termination								
	Gap(%)		Time(s)		Gap(%)		Time(s)			Efficient	Cost(%)[1]		
Instance	MINLP$_{rg}$	NLP$_{rg}$	MINLP$_{rt}$	NLP$_{rt}$	MINLP$_{g}$	NLP$_{g}$	MINLP$_{t}$	NLP$_{t}$	%Reduction	Formulation	FFC	FVC	TrC
1	2.93	5.26	< 1	< 1	0.20	0.16	127	93	27	NLP (time)	17	38	45
2	1.29	5.22	2	< 1	0.19	0.19	116	34	71	NLP (time)	15	40	46
3	0.78	2.86	1	< 1	0.20	0.20	50	28	44	NLP (time)	17	29	53
4	2.10	6.23	2	< 1	0.20	0.20	405	196	52	NLP (time)	12	37	51
5	2.19	4.40	4	< 1	0.19	0.20	1,312	1,193	9	NLP (time)	15	34	51
6	0.92	4.18	12	< 1	0.17	0.19	224	1,680	-650	MINLP (time)	18	33	49
7	1.81	6.31	5	< 1	0.20	0.18	3,237	1,757	46	NLP (time)	15	33	52
8	1.19	3.18	8	< 1	0.20	0.20	3,644	4,333	-19	MINLP (time)	11	41	48
9	1.12	2.81	9	< 1	0.20	0.20	17,289	15,228	12	NLP (time)	15	34	51
10	1.15	2.87	9	< 1	0.18	0.17	17,381	8,297	52	NLP (time)	16	31	53
11	1.12	3.18	14	< 1	0.20	0.20	61,744	16,815	73	NLP (time)	12	31	57
12	1.84	4.06	10	< 1	0.20	0.20	43,136	64,532	-50	MINLP (time)	14	32	54
13	0.71	2.40	49	< 1	0.34	0.26	time	time	-	NLP (gap)	13	39	48
14	1.02	2.76	39	< 1	0.33	0.19	time	34,961	60	NLP (time,gap)	14	31	54
15	0.73	2.57	133	< 1	0.28	0.21	time	time	-	NLP (gap)	20	29	52
16	0.89	2.21	14	< 1	0.29	0.20	time	36,634	58	NLP (time,gap)	12	30	58
17	1.40	2.70	68	1	0.82	1.07	time	time	-	MINLP (gap)	13	33	55
18	0.65	2.00	50	1	0.33	0.24	time	time	-	NLP (gap)	18	29	54
19	1.29	2.74	63	< 1	0.43	1.21	time	time	-	MINLP (gap)	16	31	54
20	1.21	2.95	117	< 1	0.74	1.63	time	time	-	MINLP (gap)	18	29	53
Average	**1.32**	**3.54**	**31**	**<1**	**0.29**	**0.37**	**41993**	**35209**	**16**		15	33	52

[1] FFC: Facility fixed cost; FVC: Facility variable cost; TrC: Transportation cost. All these costs are expressed as percentage of the objective function value.

Table A.8: Computational Results for MINLP and NLP under Excess Capacity Scenario

	Root Node Performance				Performance at Termination								
	Gap(%)		Time(s)		Gap(%)		Time(s)			Efficient	Cost(%)[1]		
Instance	MINLP$_{rg}$	NLP$_{rg}$	MINLP$_{rt}$	NLP$_{rt}$	MINLP$_{g}$	NLP$_{g}$	MINLP$_{t}$	NLP$_{t}$	%Reduction	Formulation	FFC	FVC	TrC
1	3.21	7.52	<1	<1	0.20	0.20	32	26	18	NLP (time)	34	32	33
2	0.86	0.86	<1	<1	0.20	0.20	24	26	-4	MINLP (time)	38	29	33
3	0.65	2.59	<1	<1	0.20	0.20	48	53	-9	MINLP (time)	29	24	47
4	0.24	0.24	1	<1	0.20	0.20	30	32	-6	MINLP (time)	29	28	43
5	1.74	7.09	2	<1	0.20	0.20	771	629	18	NLP (time)	37	27	36
6	1.58	4.05	<1	<1	0.20	0.20	146	189	-23	MINLP (time)	28	25	47
7	0.86	2.50	1	<1	0.20	0.20	264	295	-10	MINLP (time)	31	27	42
8	1.04	3.82	2	<1	0.20	0.20	362	471	-23	MINLP (time)	28	27	45
9	0.97	2.38	4	<1	0.20	0.20	3,232	2,564	21	NLP (time)	30	27	43
10	1.19	3.51	17	<1	0.20	0.10	3,969	6,523	-39	MINLP (time)	26	31	43
11	1.76	4.94	11	<1	0.20	0.20	6,567	3,102	53	NLP (time)	31	25	44
12	0.38	1.48	4	<1	0.20	0.20	3,588	2,117	41	NLP (time)	32	27	41
13	0.81	2.82	37	<1	0.20	0.20	58,291	15,011	74	NLP (time)	26	27	47
14	0.74	2.17	32	<1	0.20	0.20	34,133	11,077	68	NLP (time)	28	27	45
15	0.58	1.22	6	<1	0.20	0.20	2,061	1,807	12	NLP (time)	28	25	46
16	0.59	2.15	9	<1	0.20	0.20	2,930	3,356	-13	MINLP (time)	27	27	46
17	0.48	2.44	40	<1	0.20	0.20	24,597	12,767	48	NLP (time)	33	23	44
18	0.63	1.42	5	<1	0.20	0.20	6,491	2,573	60	NLP (time)	22	30	48
19	0.61	1.67	19	<1	0.20	0.20	65,860	8,659	87	NLP (time)	22	30	47
20	0.96	2.69	52	<1	0.20	0.20	time	16,840	81	NLP (time,gap)	30	23	47
Average	**0.99**	**2.88**	**12**	**<1**	**0.20**	**0.20**	**14990**	**4406**	**71**		**29**	**27**	**43**

[1] FFC: Facility fixed cost; FVC: Facility variable cost; TrC: Transportation cost. All these costs are expressed as percentage of the objective function value.

Table A.9: Computational Results for MINLP and NLP under Tight Capacity Scenario

	Root Node Performance				Performance at Termination								
	Gap(%)		Time(s)		Gap(%)		Time(s)			Efficient	Cost(%)[1]		
Instance	$MINLP_{rg}$	NLP_{rg}	$MINLP_{rt}$	NLP_{rt}	$MINLP_{g}$	NLP_{g}	$MINLP_{t}$	NLP_{t}	%Reduction	Formulation	FFC	FVC	TrC
1	1.19	1.66	< 1	< 1	0.20	0.20	82	266	-224	MINLP (time)	42	31	27
2	0.96	2.83	< 1	< 1	0.20	0.20	10,340	2,060	80	NLP (time)	42	31	27
3	0.94	1.46	< 1	< 1	0.20	0.20	155	131	15	NLP (time)	37	36	27
4	1.41	3.53	< 1	< 1	0.20	0.20	255	94	63	NLP (time)	37	32	31
5	0.57	1.25	2	< 1	0.21	0.20	time	time	-	NLP (gap)	37	34	29
6	0.44	1.63	3	< 1	0.19	0.20	35,311	2,983	92	NLP (time)	40	30	30
7	0.69	2.68	3	< 1	0.20	0.20	78,928	6,183	92	NLP (time)	37	31	32
8	0.39	1.35	7	< 1	0.21	0.20	time	1,375	98	NLP (time,gap)	37	30	32
9	0.60	1.29	8	< 1	0.36	0.32	time	time	-	NLP (gap)	40	31	29
10	0.45	1.16	22	< 1	0.26	0.20	time	16,350	81	NLP (time,gap)	35	35	31
11	0.71	2.25	18	< 1	0.46	0.35	time	time	-	NLP (gap)	40	28	33
12	0.59	1.05	9	< 1	0.40	0.29	time	time	-	NLP (gap)	38	31	31
13	0.53	0.88	28	< 1	0.33	0.27	time	time	-	NLP (gap)	39	32	29
14	0.66	1.25	13	< 1	0.45	0.38	time	time	-	NLP (gap)	38	29	33
15	0.44	0.90	34	< 1	0.33	0.31	time	time	-	NLP (gap)	38	30	31
16	0.54	1.28	58	< 1	0.48	0.37	time	time	-	NLP (gap)	39	27	34
17	0.56	1.02	44	< 1	0.43	0.32	time	time	-	NLP (gap)	37	32	32
18	0.69	0.99	32	< 1	0.67	0.55	time	time	-	NLP (gap)	34	33	33
19	0.66	1.02	236	1	0.54	0.42	time	time	-	NLP (gap)	42	26	31
20	0.51	0.81	40	1	0.50	0.39	time	time	-	NLP (gap)	37	31	32
Average	**0.68**	**1.51**	**28**	**<1**	**0.34**	**0.29**	**66734**	**53312**	**20**		**38**	**31**	**31**

[1] FFC: Facility fixed cost; FVC: Facility variable cost; TrC: Transportation cost. All these costs are expressed as percentage of the objective function value.

Appendix B

Proof of Proposition: Chapter 3

B.1 Proof of Proposition 1

Proof. We first prove the validity of constraints (3.11) by showing that if a binary vector $(\bar{x}, \bar{z})$ does not satisfy these inequalities for at least one set $C \in \mathcal{C}_{jlt}$, then $(\bar{x}, \bar{z}) \notin X$. For a given $C \in \mathcal{C}_{jlt}$, $j \in V_1$, $l \in L$, $t \in T$, if $\sum_{l_1 \in L} \bar{z}_{jl_1lt} = 0$ and $\sum_{i \in C} \bar{x}_{ijlt} > 0$, then $(\bar{x}, \bar{z}) \notin X$ given that customers cannot be assigned to facility j at level l if this level is not open. If $\sum_{l_1 \in L} \bar{z}_{jl_1lt} = 1$ and $\sum_{i \in C} \bar{x}_{ijlt} = |C|$, then the set of customers R assigned to facility j at level l in $(\bar{x}, \bar{z})$ satisfies $|R \cap C| = |C|$ and thus $C \subseteq R$. Then, by definition of C there exists at least one customer $i \in C$ such that $\mathcal{P}_{it}(\overline{W}_{jt}(C, l) + ST_{ij} \leq \tau_{it}) < \theta_{it}$, and thus $(\bar{x}, \bar{z}) \notin X$.

Now, to prove that ILP1 is a vaid formulation, we show that inequalities (3.11) provide sufficient conditions for feasibility for the DCFLP. This can be done by noting that any solution satisfying (3.4)–(3.7), (3.9), (3.10) but violating at least one probabilistic constraint (3.8) can be removed by at least one cover inequality given that the sets $\mathcal{C}_{jlt}$, $j \in V_1$, $l \in L$, $t \in T$, contain all infeasible assignments with respect to (3.8). Therefore, the result follows. $\square$

B.2 Proof of Proposition 2

Proof. We first prove the validity of constraints (3.12) by showing that if $(\bar{x}, \bar{z}) \in X$, then it must satisfy constraint (3.12). For any $i \in I, j \in V_1, l \in L, t \in T$, if $\sum_{l_1 \in L} z_{jl_1lt} = 0$ and $x_{ijlt} = 1$ then $(\bar{x}, \bar{z}) \notin X$ because a customer i can not be assigned to a facility j at level l if it is not open at that level. If $\sum_{l_1 \in L} z_{jl_1lt} = 1$ and $x_{ijlt} = 0$, then (3.12) is equivalent to (3.5), and in that case $(\bar{x}, \bar{z}) \in X$. Whereas, for $\sum_{l_1 \in L} z_{jl_1lt} = 1$ and $x_{ijlt} = 1$, if (3.12) is violated then

$$\sum_{i' \in I \setminus \{i\}} \lambda_{i't} x_{i'jlt} > \Delta_{ijlt}.$$

and thus, $\mathcal{P}_{it}(\bar{W}_{jt} + ST_{ij} \leq \tau_{it}) < \theta_{it}$ by definition of Δ_{ijlt}. Hence, $(\bar{x}, \bar{z}) \notin X$.

We now show that inequalities (3.12) are sufficient to reformulate INLP. Note that any $(\bar{x}, \bar{z})$ satisfying (3.4)–(3.7), (3.9), (3.10) must also satisfy (3.12) for $(\bar{x}, \bar{z}) \in X$. As, we add (3.12) for every $i \in I, j \in V_1, l \in L, t \in T$, this ensures that in period t customer i is served with probability greater or equal to θ_{it}. Therefore, such $(\bar{x}, \bar{z})$ is a feasible solution to INLP. This completes the proof. □

B.3 Proof of Lemma 1

Proof. Lets assume that in period $t \in T$, a customer i is assigned to facility j opened at level $l \in L \setminus \{0\}$. As a facility j can be opened only at one level, therefore, in the expression,

$$DT_{it}(x, z) = \sum_{j \in V_1} \sum_{l \in L} (W_{jt}(x, z) + ST_{ij}) \, x_{ijlt}$$

we drop the summations and set $x_{ijlt} = 1$ (due to single assignment assumption) to give

$$DT_{it}\,(x,z) = (W_{jt}(x,z) + ST_{ij})$$

Let X be the random variable for gamma distributed travel time ST_{it}, Y be the random variable for exponentially distributed waiting time W_{jt}, and Z be the random variable for delivery time D_{it}. Also, let X and Y are independent with probability density functions f_X and f_Y, respectively. Then $F_Z(\tau_{it}) = \mathcal{P}_{it}(DT_{it}\,(x,z) \leq \tau_{it})$ for random variable DT_{it} can be given as follows [see Theorem 7.1, 58].

$$F_Z(\tau) = \mathcal{P}(X + Y \leq \tau) = \int_{-\infty}^{\infty} f_X(a) F_Y\,(\tau - a)\,da \tag{B.1}$$

(for simplification we drop the subscripts from the notations.) where, $F_Y(\tau - a)$ is the cumulative distribution function (cdf) of Y, and

$$F_Y\,(\tau - a) = \left(1 - \exp^{-\lambda'(\tau - a)}\right) \tag{B.2}$$

where, λ' is the rate of the exponential distribution which is equal to the residual service capacity $(\Omega - \Lambda)$ at the facility. As, $f_X(a)$ is defined for $a \geq 0$, and $F_Y\,(\tau - a)$ is defined for $\tau - a \geq 0 \implies a \in [0, \tau]$, and

$$F_Z(\tau) = \int_0^{\tau} f_X(a)(1 - \exp^{-\lambda'(\tau - a)})da \tag{B.3}$$

We know that integration is area under the curve, which is computed as the sum of the values of the function at every point between the limits of the integration. At any fixed point $a \in [0, \tau]$, the value of $f_X(a)$ is independent of λ'. However, with increasing λ' (by decreasing the rate of demand arrival Λ at the facility), the cdf of exponential distribution i.e.

$\left(1-\exp^{-\lambda'(\tau-a)}\right)$ is nondecreasing, and hence, the probability. This completes the proof. $\square$

B.4 Proof of Proposition 3

Proof. We show that if a binary vector $(\bar{x}, \bar{z})$ does not satisfy these inequalities for at least one set $C \in \mathcal{C}_{jlt}$, then $(\bar{x}, \bar{z}) \notin X$. For a given $C \in \mathcal{C}_{jlt}$, $j \in V_1$, $l \in L$, $t \in T$, if $\sum_{l_1 \in L} \bar{z}_{jl_1lt} = 0$ and $\sum_{i \in E(C)} \bar{x}_{ijlt} > 0$, then $(\bar{x}, \bar{z}) \notin X$ given that customers cannot be assigned to facility j at level l if this level is not open. If $\sum_{l_1 \in L} \bar{z}_{jl_1lt} = 1$ and $\sum_{i \in E(C)} \bar{x}_{ijlt} \geq |C|$, then the set of customers R assigned to facility j at level l in $(\bar{x}, \bar{z})$ satisfies $|R| \geq |C|$. There exists two possible relationships between C and R, either $C \subseteq R$ or $C \nsubseteq R$. If $C \subseteq R$, by definition of C there exists at least one customer $i \in C$ such that $\mathcal{P}_{it}(\overline{W}_{jt}(C,l) + ST_{ij} \leq \tau_{it}) < \theta_{it}$, and thus $(\bar{x}, \bar{z}) \notin X$. If $C \nsubseteq R$, there exists at least one customer $i \in R$ such that $\mathcal{P}_{it}(\overline{W}_{jt}(R,l) + ST_{ij} \leq \tau_{it}) \leq \mathcal{P}_{it}(\overline{W}_{jt}(C,l) + ST_{ij} \leq \tau_{it}) < \theta_{it}$, where the first inequality holds given that $\sum_{i \in R} \lambda_{it} \geq \sum_{i \in C} \lambda_{it}$ (refer to Lemma 3.5.1), and that for any $i \in R \setminus C$ the expected travel time and coefficient of variation of i are greater than the expected travel time and coefficient of variation of any customer in C that cannot be satisfied with desired probability, and thus $(\bar{x}, \bar{z}) \notin X$. $\square$

B.5 Proof of Proposition 4

Proof. For a given $C \in \mathcal{C}_{jlt}$, $j \in V_1$, $l \in L$, $t \in T$, from the previous proposition, we know that

$$\sum_{i \in E(C)} x_{ijlt} \leq (|C| - 1),$$

is valid for X, and when we combine this with the fact that, if a cover C is infeasible for capacity level l it will also be infeasible for each level $l = 1 \cdots, l$, e.i., $\mathcal{C}_{jlt} \subseteq \mathcal{C}_{jl-1t}$, the result follows. $\square$

B.6 Pseudocode for Separation Algorithm for SP$_{jlt}$

Algorithm 4: A heuristic separation algorithm for SP$_{jlt}$

Data: Fractional point $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$

1 **for** $j \in V_1,\ l \in L \setminus \{0\},\ t \in T$ **do**
2   **if** $\sum_{l_1 \in L} \hat{z}_{jl_1lt} > 0$ **then**
3   Create vectors $\{CustWt_i\} \leftarrow \hat{x}_{ijlt}\lambda_{it}\ \forall i \in I$ and $\{CustPos_i\} \leftarrow i\ \forall i \in I$;
4   Arrange $[CustWt]$ in the descending order of values, and arrange $[CustPos]$ according to the order of $[CustWt]$;
5      $DmdAssigned \leftarrow 0$, $ViolationFlag \leftarrow 0$;
6   **for** $i \in I$ **do**
7   **if** $CustWt_i > 0$ **then**
8          $DmdAssigned{+}= \lambda_{it}$;
9          $\Lambda = \mu^l_{jt} - DmdAssigned$;
10          $int\ i' \leftarrow 0$;
11   **for** $i' = 1$ *to* i **do**
12            $Prob \leftarrow CalculateProbability(\Lambda, E[t_{CustPos_{i'}j}])$;
13   **if** $Prob < \theta_{it}$ **then**
14              $ViolationFlag \leftarrow 1$ and goto line 15;
15   **if** $ViolationFlag = 1$ **then**
16            $\zeta = \sum_{i'=1}^{i}(\hat{z}_{jl_1lt} - \hat{x}_{CustPos_{i'}jlt})$;
17   **if** $(\hat{z}_{jl_1lt} - \zeta) > ST$ **then** // *ST > 0*
18   Add violated cover inequality, and goto line 1;

B.7 Pseudocode for Local Search Heuristic

Let $\hat{x}$ be the incumbent solution, $f(\hat{x})$ be the objective function value, and the set of decision vectors that are neighbors of $\hat{x}$ is denoted as $\mathcal{N}(\hat{x}) = \{\hat{x}_1, \hat{x}_2, .., \hat{x}_t\}$. Moreover, $\mathcal{N}_1$ and $\mathcal{N}_2$ represent the *Swap* and *Shift* neighborhoods, respectively.

Algorithm 5: Pseudocode for local search heuristic

Data: Integer solution $\hat{x}$

$ImprovementFlag \leftarrow 0$;
$i \leftarrow 1$;
while $i \leq |\mathcal{N}_1(\hat{x})|$ **do**
    if *($\hat{x}_i$ is feasible and $f(\hat{x}_i) < f(\hat{x})$)* **then**
        $\hat{x} \leftarrow \hat{x}_i$;
        $i \leftarrow |\mathcal{N}_1(\hat{x})| + 1$;
        $ImprovementFlag \leftarrow 1$;
$i \leftarrow 1$;
while $i \leq |\mathcal{N}_2(\hat{x})|$ **do**
    if *($\hat{x}_i$ is feasible and $f(\hat{x}_i) < f(\hat{x}_i)$)* **then**
        $\hat{x}_i \leftarrow \hat{x}_i$;
        $i \leftarrow |\mathcal{N}_2(\hat{x}_i)| + 1$;
        $ImprovementFlag \leftarrow 1$;

Output: Integer Solution $\hat{x}$

Output: *ImprovementFlag*

B.8 Details of Test Instances

We generate the parameters based on the scheme used in [74], and use the population, latitude/longitude from the United States Census Bureau (`https://www2.census.gov/geo/docs/reference/cenpop2010/county`), along with average housing price (`https://www.census.gov/support/USACdataDownloads.html#HSG`) for every county in the

US. For $t = 1$, the arrival rate is set to $\lambda_{it} = 0.0001 \times p_i$, where p_i is the population of county i, and for $t \geq 2, \lambda_{it} = (1 \pm U[0.1, 0.4]) \times \lambda_{i(t-1)}$. We assume same value of capacity levels in every period $t \in T$. Therefore, for any $t \in T$, the service rates μ^L_{jt}, at the highest capacity level L are first generated uniformly on $U[10, 160]$, and then scaled such that $(|T| \times \sum_{j \in V_1} \mu^L_{jt})/(\sum_{i \in I} \sum_{t \in T} \lambda_{it}) = \kappa$, where $\kappa \geq 0$ is an scaling parameter varied to represent different capacity scenarios. The service rates at lower levels are set as: $\mu^l_{jt} = l/|L| \times \mu^L_{jt} \, \forall j \in V_1, \forall l = \{0, 1, .., |L| - 1\}$. The fixed cost of a facility consists of cost to open/close, change the capacity level as well as the maintenance cost of the facility at current level. We use f^o_{jl} to denote the cost of increasing the capacity at facility j by l levels and f^c_{jl} to denote the cost to decrease the capacity at facility j by l levels. f^o_{jl} is computed as $f^o_{jl} = f^o_{j(l-1)} + 0.9 \times \left(f^o_{j(l-1)} - f^o_{j(l-2)}\right)$, and $f^c_{jl} = 0.05 \times f^o_{jl}$. We set $f^o_{j0} = 0$ and $f^o_{j1} = 0.1 \times h_i$, where h_i is the average housing price of county i. Similarly, the cost to maintain a facility j at level l is computed as $F_{jl} = F_{j(l-1)} + 0.9 \times \left(F_{j(l-1)} - F_{j(l-2)}\right)$, and we assume $F_{jl} = 0$ and $F_{j1} = 20 \times \mu_{j1}$. Therefore, the fixed cost coefficient $f_{jl_1l_2t}$, is given by:

$$f_{jl_1l_2t} = \begin{cases} f^o_{j(l_2-l_1)} + F_{jl_2}, & \text{if } l_1 < l_2 \\ f^o_{j(l_1-l_2)} + F_{jl_2}, & \text{if } l_1 > l_2 \\ F_{jl_2}, & \text{if } l_1 = l_2 \neq 0. \end{cases}$$

The variable cost c_{ijlt} depends on the transportation cost between two nodes and the production cost at the facility. Transportation cost from facility $j \in V_1$ to demand node $i \in I$ is $d_{ij} = 0.03 \times distance_{ij}$, where $distance_{ij}$ is the spherical distance between nodes i and j. Production cost at facility $j \in V_1$ operating at level $l = 1$ is set to $P_{j1} = 150/(\mu_{j1})^{0.25}$, and for higher levels $l \geq 2$, $P_{jl} = 0.9 \times P_{j(l-1)}$. Therefore, the variable cost is given by $c_{ijlt} = (d_{ij} + P_{jl})$. The expected number of days, $E[ST_{ij}]$, to ship from facility $j \in V_1$ to customer $i \in I$ is computed by dividing the spherical distance between facility j and customer i by the average travel speed of 60 kmph.

Table B.1: Specifics of Test Instances used in the Computational Experiments

Class	Data	Model	Set	$\lvert V_1 \rvert$	$\lvert I \rvert$
A	Amazon	Static and Single capacity	I	30	125, 250, 375, 500, 625
		(L=1, T=1)	II	55	750, 875, 1000, 1125, 1250
			III	80	1375, 1500, 1625, 1750, 1875
			IV	111	2000, 2125, 2250, 2375, 2500
B	Amazon	Dynamic and Modular capacities	I	20	130, 160, 190, 220, 250
		(L=3, T=4)	II	30	280, 310, 340, 370, 400
			III	40	430, 460, 490, 520, 550
			IV	50	580, 610, 640, 670, 700
C	US Counties	Static and Single capacity	I	45	125, 250, 375, 500
		(L=1, T=1)	II	90	625, 750, 875, 1000
			III	135	1125, 1250, 1375, 1500
			IV	180	1625, 1750, 1875, 2000
			V	225	2125, 2250, 2375, 2500
D	US Counties	Dynamic and Modular capacities	I	27	130, 160, 190, 220, 250
		(L=3, T=4)	II	36	280, 310, 340, 370, 400
			III	45	430, 460, 490, 520, 550
			IV	54	580, 610, 640, 670, 700

B.9 Parameter Tuning

In additional to the proposed algorithmic enhancements, we fine-tune some of the CPLEX parameters for all the three variants of our exact algorithm and the ILP1$_{\mathbf{R}}$. We focus mainly on parameters that are helpful in finding feasible solutions quickly. Therefore, the branching is first performed on location and capacity selection variables z and then on allocation variables x. The heuristic (*CPXPARAM_MIP_Strategy_HeuristicFreq*) and relaxation induced neighborhood search (RINS) heuristic (*CPX_PARAM_RINSHEUR*) parameters are set to 5. We also fine-tune *MIP Emphasis* parameter separately for every variant of our algorithm. Based on that *CPX_PARAM_MIPEMPHASIS* is set to 1 (emphasize feasibility over optimality) in B&C-B version and LBP$_{\mathbf{R}}$, and to 2 (emphasize optimality over feasibility) in B&C-R1 and B&C-R2.

For an efficient implementation of *USERC*, we define a separation tolerance (ST) parameter on the violation of inequalities by the fractional solution. In B&C-R1, ST is set to 0.01, whereas in B&C-R2, ST equals to 0.05 and 0.1 for cover and residual service capacity inequalities, respectively. Also, we control the number of cuts to be added at a node in the enumeration tree using parameters (i) *CSN*, which defines the number of times the *USERC* function is called from the same node, and (ii) *EPSOBJ*, which is defined as the minimum improvement in lower bound value after adding cuts. If improvement in lower bound is less than *EPSOBJ* then no further iterations of *USERC* function are performed. We adopt different settings at root node and child nodes to control the number of cuts. For cover inequalities, at the root node we use $EPSOBJ = 0.1\%$, and at child nodes we use $CNS \leq 2$. Whereas for *RSC* inequalities, at the root node *EPSOBJ* is set to 0, and $CNS \leq 20$; and at child nodes we use $CNS \leq 20$. Parameter *Depth* is used to control the frequency in which *USERC* function is called in the enumeration tree. *Depth* is set to 200 and 75 for cover inequalities in B&C-R1 and B&C-R2, respectively, and for RSC constraints, a value of *Depth*=40 is used. Finally, the parameter ϵ used to approximate Δ_{ijlt} is set to 0.001.

B.10 Detailed Results of Branch-and-Cut Algorithms

Table B.2: Average number of CIs, ECIs and RSCs for Sets I and II

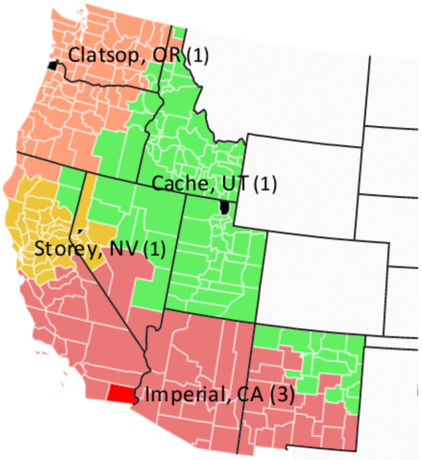
Scenario	Class	B&C-B CIs	B&C-R1 CIs	B&C-R2 ECIs	B&C-R2 RSCs
1	A	6,263	5,071	142	1,361
	B	1,074	463	37	2,767
	C	570	413	58	1,254
	D	511	172	50	3,092
2	A	1,306	130	7	1,307
	B	62	83	34	2,323
	C	660	205	9	1,180
	D	172	167	27	2,331
3	A	9,664	6,053	509	262
	B	1,592	696	222	3,537
	C	4,132	1,985	301	1,121
	D	850	275	92	3,549
4	A	4,204	952	7	322
	B	1,173	952	125	3,144
	C	6,158	2,679	70	1,113
	D	860	263	103	3,320
5	A	9,689	4,125	344	1,709
	B	743	300	83	2,996
	C	4,857	1,676	394	1,284
	D	1,611	739	188	3,914
Average		2,808	1,370	140	2,094

Table B.3: Average number of CIs, ECIs and RSCs for Sets III, IV and V

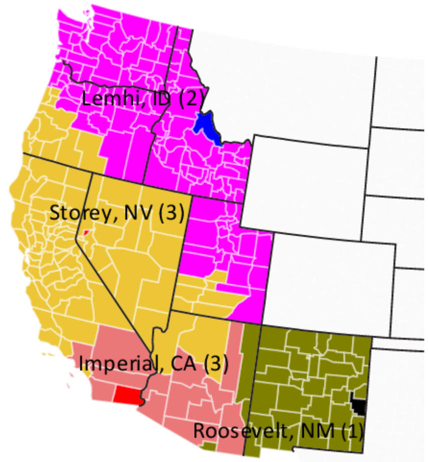
Scenario	Class	B&C-B CIs	B&C-R1 CIs	B&C-R2 ECIs	B&C-R2 RSCs
1	A	7,088	3,008	78	5,747
	B	610	232	40	5,795
	C	4,028	1,477	148	5,910
	D	789	254	49	6,765
2	A	2,136	1,451	39	4,873
	B	128	85	24	5,169
	C	2,505	1,099	173	4,587
	D	205	128	27	5,004
3	A	10,793	4,102	126	4,848
	B	2,252	505	57	6,338
	C	7,705	1,682	213	7,149
	D	868	411	45	7,164
4	A	4,097	1,986	232	4,213
	B	1,372	203	34	7,455
	C	8,107	1,987	273	6,319
	D	1,543	173	58	7,753
5	A	17,222	9,424	584	4,806
	B	1,152	492	113	8,009
	C	4,900	998	505	4,459
	D	1,001	263	191	5,435
Average		3,925	1,498	150	5,890

B.11 Sensitivity Analysis: Effect of Varying Demand in Multiple Periods on Network Configuration

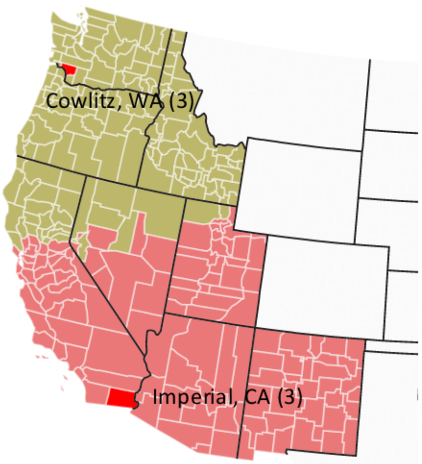
We analyze the effect of varying demand on location and allocation decisions for a problem instance with dynamic and modular capacities ($T = 4$, $L = 3$) and $\theta = 0.9$, $\tau = 1.5$, and $CV = 0.5$. The set of potential facilities, V_1 and customers I are drawn from the following states: Washington, Oregon, California, Nevada, Idaho, Utah, Arizona, and New Mexico. From each state, we randomly choose two counties as potential facility locations (as set V_1), and every county in these states belongs to set I. The demand in multiple periods, λ^F_{it} is generated as follows: $\lambda^F_{i1} = 0.0001\times$ County Population; $\lambda^F_{i2} = U[1.3, 1.4] \times \lambda^F_{i1}$; $\lambda^F_{i3} = U[0.8, 0.9] \times \lambda^F_{i2}$; and $\lambda^F_{i4} = U[1.75, 2.0] \times \lambda^F_{i3}$. Figures B.1c-B.1f represent the location and allocation decisions for the four time periods. In the first time period, the model opens facilities in Cowlitz, WA and Imperial, CA at capacity level 3. In the second time period, with the increase in demand, the model opens an additional facility in Storey, NV at capacity level 1. In the third time period, as the demand decrease, the model prescribes the same facilities and capacity levels, however, the allocations change. In the last scenario, where the demand increase substantially, model opens an additional facility in Roosevelt, NM at capacity level 3 besides changing the capacity of the existing facility at Storey, NV from 1 to 3. These results highlight the importance of dynamic location and capacity planning to meet the varying customer demand and probabilities service level constraints.



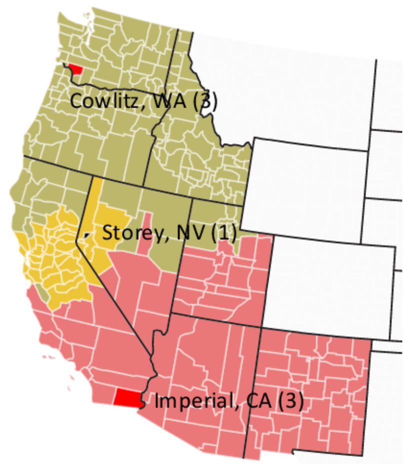
(a) Static Solution *AvgDmd*



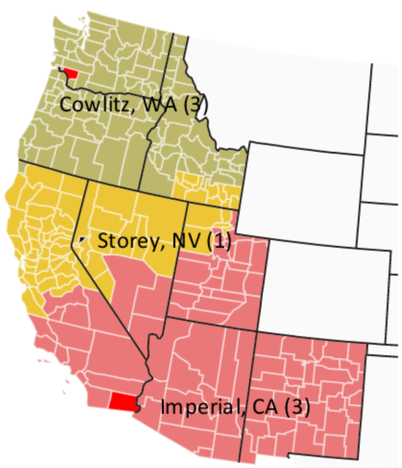
(b) Static Solution *MaxDmd*



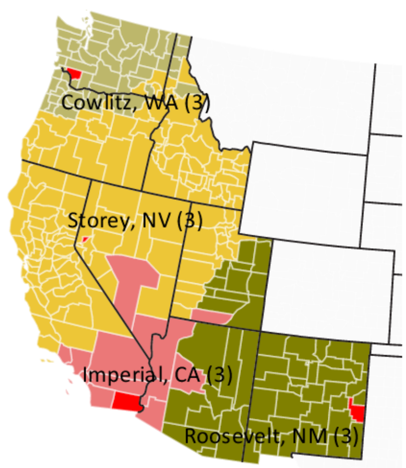
(c) Time Period 1 *RegDmd*



(d) Time Period 2 *RegDmd*



(e) Time Period 3 *RegDmd*



(f) Time Period 4 *RegDmd*

Figure B.1: Solution of Average, Maximum and Regular Demand Scenarios

Appendix C

Detailed Tables of Computational Results: Chapter 4

Table C.1: Comparison of the branch-and-cut algorithm and CPLEX under balanced cost scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	6,479	284	96	950	B&C (time)	26	38	36
	2	0.00	0.00	3,213	2,140	33	2,050	B&C (time)	22	32	46
	3	0.00	0.00	37,483	32,618	13	1,050	B&C (time)	18	35	48
	4	0.00	0.00	37,450	21,838	42	1,400	B&C (time)	17	33	50
II	5	0.00	0.00	2,763	2,217	20	1,375	B&C (time)	25	39	36
	6	0.00	0.00	1,554	808	48	250	B&C (time)	29	43	28
	7	0.00	0.00	51,281	37,939	26	3,000	B&C (time)	16	39	45
	8	0.00	0.00	29,449	12,514	58	2,155	B&C (time)	26	39	35
III	9	0.00	0.00	7,494	8,230	-9	550	CPLEX (time)	28	38	34
	10	0.00	0.00	23,014	17,302	25	3,150	B&C (time)	23	39	37
	11	1.42	0.00	limit[2]	72,325	16	24,143	B&C (time, gap)	25	45	29
	12	26.31	26.31	-	-	-	-	-	-	-	-
IV	13	0.00	0.00	1,883	653	65	1,700	B&C (time)	31	47	22
	14	0.00	0.00	13,247	4,598	65	1,847	B&C (time)	28	47	26
	15	4.06	0.14	limit	limit	-	12,607	B&C (gap)	28	45	27
	16	29.49	29.49	-	-	-	-	-	-	-	-
V	17	0.00	0.00	1,508	1,368	9	2,525	B&C (time)	34	42	24
	18	0.00	0.00	50,456	50,255	0	11,884	B&C (time)	28	47	25
	19	-	-	-	-	-	-	-	-	-	-
	20	33.31	33.31	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	1,508	284	-9	250		16	32	22
	Avg.	0.34	0.01	27,492	21,964	20	4,415		25	41	34
	Max.	4.06	0.14	limit	limit	96	24,143		34	47	50

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value (total cost).
[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.2: Comparison of the branch-and-cut algorithm and CPLEX under dominant fixed cost scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	58	59	-1	1,955	CPLEX (time)	52	32	16
	2	0.00	0.00	223	219	2	2,950	B&C (time)	47	31	22
	3	0.00	0.00	375	377	-1	6,075	CPLEX (time)	52	20	28
	4	0.00	0.00	598	3,812	-84	7,967	CPLEX (time)	46	20	34
II	5	0.00	0.00	83	82	1	2,200	B&C (time)	50	34	16
	6	0.00	0.00	605	115	81	1,350	B&C (time)	67	20	13
	7	0.00	0.00	342	288	16	7,125	B&C (time)	49	23	27
	8	0.00	0.00	1,760	626	64	5,900	B&C (time)	64	19	17
III	9	0.00	0.00	1,118	425	62	2,400	B&C (time)	66	18	16
	10	0.00	0.00	23,463	7,356	69	2,300	B&C (time)	56	23	21
	11	0.00	0.00	9,739	14,642	-33	3,900	CPLEX (time)	63	22	14
	12	0.00	0.02	17,339	limit[2]	-80	7,699	CPLEX (time, gap)	61	23	16
IV	13	0.00	0.00	314	192	39	2,125	B&C (time)	67	21	12
	14	0.00	0.00	4,931	249	95	2,900	B&C (time)	66	22	12
	15	0.00	0.00	1,341	1,131	16	10,248	B&C (time)	65	22	13
	16	0.00	0.00	53,300	3,082	94	2,700	B&C (time)	59	21	20
V	17	0.00	0.00	1,525	613	60	1,894	B&C (time)	68	23	9
	18	0.00	0.00	26,857	641	98	3,850	B&C (time)	62	25	12
	19	0.14	0.00	limit	69,856	19	32,921	B&C (time, gap)	50	30	20
	20	21.12	21.12	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	58	59	-84	1,350		46	18	9
	Avg.	0.01	0.00	12,109	10,005	17	5,708		58	24	18
	Max.	0.14	0.02	limit	limit	98	32,921		68	34	34

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.

[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.3: Comparison of the branch-and-cut algorithm and CPLEX under dominant safety stock cost scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	1,990	563	72	300	B&C (time)	10	31	59
	2	0.00	0.00	249	219	12	2,950	B&C (time)	7	24	69
	3	0.00	0.00	52,575	32,402	38	1,938	B&C (time)	6	26	68
	4	0.86	0.00	limit	48,842	43	4,953	B&C (time, gap)	5	25	70
II	5	0.00	0.00	3,356	1,615	52	650	B&C (time)	12	19	69
	6	0.00	0.00	8,668	1,502	83	300	B&C (time)	16	24	61
	7	0.00	0.00	69,093	53,075	23	1,425	B&C (time)	6	31	64
	8	0.00	0.00	65,232	34,409	47	600	B&C (time)	13	19	68
III	9	0.00	0.00	25,827	15,451	40	1,724	B&C (time)	13	23	64
	10	30.25	30.25	-	-	-	-	-	-	-	-
	11	0.62	0.00	limit[2]	52,324	39	2,400	B&C (time, gap)	14	27	59
	12	-	-	-	-	-	-	-	-	-	-
IV	13	0.00	0.00	7,929	904	89	1,637	B&C (time)	19	29	52
	14	0.00	0.00	23,734	17,694	25	1,550	B&C (time)	17	30	53
	15	-	-	-	-	-	-	-	-	-	-
	16	-	-	-	-	-	-	-	-	-	-
V	17	0.16	0.00	limit	1,895	98	2,163	B&C (time, gap)	18	31	51
	18	0.52	0.00	limit	73,358	15	1,611	B&C (time, gap)	16	27	57
	19	-	-	-	-	-	-	-	-	-	-
	20	-	-	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	249	219	12	300		5	19	51
	Avg.	0.15	0.00	43,129	23,875	45	1,729		12	26	62
	Max.	0.86	0.00	limit	73,358	98	4,953		19	31	70

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.
[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.4: Comparison of the branch-and-cut algorithm and CPLEX under dominant transportation cost scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	585	228	61	1,775	B&C (time)	14	69	17
	2	0.00	0.00	528	465	12	2,050	B&C (time)	11	65	24
	3	0.00	0.00	4,765	1,921	60	3,300	B&C (time)	9	68	23
	4	0.00	0.00	11,504	5,466	52	5,500	B&C (time)	8	67	25
II	5	0.00	0.00	1,059	420	60	1,686	B&C (time)	13	66	20
	6	0.00	0.00	2,023	949	53	1,119	B&C (time)	13	75	12
	7	0.00	0.00	15,112	3,684	76	5,562	B&C (time)	8	68	24
	8	0.00	0.00	4,192	1,910	54	4,627	B&C (time)	12	72	16
III	9	0.00	0.00	1,919	793	59	1,784	B&C (time)	14	69	16
	10	0.00	0.00	3,580	5,593	-36	17,670	CPLEX (time)	12	65	22
	11	0.00	0.00	54,975	23,275	58	6,282	B&C (time)	13	72	15
	12	0.00	0.00	43,118	21,842	49	5,914	B&C (time)	12	70	17
IV	13	0.00	0.00	1,441	797	45	1,575	B&C (time)	16	73	11
	14	0.00	0.00	10,392	2,440	77	4,050	B&C (time)	11	78	11
	15	0.00	0.00	39,752	11,501	71	7,078	B&C (time)	15	67	18
	16	0.00	0.00	41,380	21,227	49	4,300	B&C (time)	12	67	20
V	17	0.00	0.00	838	570	32	1,775	B&C (time)	15	74	11
	18	0.00	0.00	10,406	9,941	4	8,550	B&C (time)	14	71	15
	19	0.36	0.00	limit[2]	34,506	60	55,385	B&C (time, gap)	10	68	22
	20	0.00	0.00	41,008	31,309	24	5,990	B&C (time)	12	68	20
	Min.[2]	0.00	0.00	528	228	-36	1,119		8	65	11
	Avg.	0.02	0.00	18,735	8,942	52	7,299		12	70	18
	Max.	0.36	0.00	limit	34,506	77	55,385		16	78	25

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.
[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.5: Comparison of the branch-and-cut algorithm and CPLEX under service distance scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	837	719	14	1,489	B&C (time)	26	37	37
	2	0.00	0.00	4,140	877	79	1,550	B&C (time)	20	32	48
	3	0.00	0.00	41,070	32,136	22	2,015	B&C (time)	18	37	45
	4	0.00	0.00	18,534	9,808	47	3,880	B&C (time)	17	33	50
II	5	0.00	0.00	5,055	2,607	48	775	B&C (time)	25	39	36
	6	0.00	0.00	2,530	1,533	39	2,060	B&C (time)	26	43	31
	7	0.00	0.00	39,716	36,455	8	3,826	B&C (time)	16	35	50
	8	0.00	0.00	24,969	10,071	60	5,400	B&C (time)	23	36	41
III	9	0.00	0.00	7,279	4,670	36	2,325	B&C (time)	26	43	31
	10	0.00	0.00	26,394	32,983	-20	7,938	CPLEX (time)	22	41	37
	11	0.00	0.00	47,462	36,415	23	2,846	B&C (time)	25	45	29
	12	12.39	0.84	limit[2]	limit	-	2,999	B&C (gap)	24	44	32
IV	13	0.00	0.00	2,231	915	59	1,251	B&C (time)	31	47	22
	14	0.00	0.00	41,864	11,211	73	2,486	B&C (time)	28	49	22
	15	28.04	28.04	-	-	-	-	-	-	-	-
	16	36.06	36.06	-	-	-	-	-	-	-	-
V	17	0.00	0.00	1,416	1,018	28	1,775	B&C (time)	30	45	25
	18	0.27	0.00	limit	40,635	53	5,745	B&C (time, gap)	32	41	27
	19	-	-	-	-	-	-	-	-	-	-
	20	33.39	33.39	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	837	719	-20	775		16	32	22
	Avg.	0.79	0.05	27,258	19,272	29	3,023		24	41	35
	Max.	12.39	0.84	limit	limit	79	7,938		32	49	50

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.

[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.6: Comparison of the branch-and-cut algorithm and CPLEX under under tight capacity scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	587	3,146	-81	1,050	CPLEX (time)	25	37	38
	2	0.00	0.00	15,098	14,658	3	29,028	B&C (time)	19	44	37
	3	0.00	0.00	24,167	8,179	66	22,821	B&C (time)	21	40	40
	4	0.00	1.15	36,596	limit[2]	-58	28,557	CPLEX (time, gap)	19	35	46
II	5	0.00	0.00	2,231	5,223	-57	17,597	CPLEX (time)	23	40	37
	6	0.00	0.00	2,225	1,340	40	1,100	B&C (time)	24	41	35
	7	1.38	0.87	limit	limit	-	136,538	B&C (gap)	23	38	39
	8	0.00	0.00	25,131	26,523	-5	13,612	CPLEX (time)	20	51	29
III	9	1.21	0.04	limit	limit	-	52,475	B&C (gap)	22	43	35
	10	0.00	0.00	46,226	21,581	53	32,918	B&C (time)	17	47	36
	11	0.00	0.00	59,860	43,854	27	18,730	B&C (time)	17	45	39
	12	44.76	44.76	-	-	-	-	-	-	-	-
IV	13	0.00	0.00	26,297	51,812	-49	72,272	CPLEX (time)	20	46	33
	14	2.65	1.35	limit	limit	-	18,230	B&C (gap)	22	38	39
	15	3.15	0.00	limit	70,490	18	56,063	B&C (time, gap)	18	52	31
	16	41.11	41.11	-	-	-	-	-	-	-	-
V	17	2.46	0.06	limit	limit	-	78,282	B&C (gap)	26	36	38
	18	1.78	2.74	limit	limit	-	12,591	CPLEX (gap)	19	46	34
	19	13.69	13.69	-	-	-	-	-	-	-	-
	20	-	-	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	587	1,340	-81	1,050		17	35	29
	Avg.	0.79	0.39	46,876	47,615	-2	36,992		21	42	37
	Max.	3.15	2.74	limit	limit	66	136,538		26	52	46

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.
[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)

Table C.7: Comparison of the branch-and-cut algorithm and CPLEX under loose capacity scenario

		Gap(%)		Tim(sec)					Cost(%)[1]		
Set	Instance	CPLEX	B&C	CPLEX	B&C	%Red	# UserCuts	BestVersion	FC	TrC	SsC
I	1	0.00	0.00	112	109	2	2,475	B&C (time)	39	26	35
	2	0.00	0.00	416	393	5	5,113	B&C (time)	39	28	33
	3	0.00	0.00	567	530	6	6,375	B&C (time)	38	34	28
	4	0.00	0.00	3,095	641	79	10,298	B&C (time)	33	33	34
II	5	0.00	0.00	148	139	6	2,400	B&C (time)	46	28	25
	6	0.00	0.00	540	555	-3	3,100	CPLEX (time)	36	31	33
	7	0.00	0.00	80,445	982	99	7,425	B&C (time)	37	32	32
	8	0.00	0.00	6,712	2,021	70	7,000	B&C (time)	28	38	34
III	9	0.00	0.00	5,578	263	95	2,350	B&C (time)	41	33	26
	10	0.00	0.00	819	805	2	5,144	B&C (time)	40	28	31
	11	0.07	0.00	limit[2]	27,709	68	5,925	B&C (time, gap)	35	34	30
	12	-	-	-	-	-	-	-	-	-	-
IV	13	0.00	0.00	616	470	24	2,375	B&C (time)	30	35	35
	14	0.00	0.00	1,732	1,465	15	4,950	B&C (time)	42	31	27
	15	0.00	0.00	13,689	11,067	19	21,454	B&C (time)	21	57	22
	16	-	-	-	-	-	-	-	-	-	-
V	17	0.00	0.00	1,002	874	13	3,016	B&C (time)	36	34	30
	18	0.00	0.00	81,482	50,995	37	3,347	B&C (time)	33	38	28
	19	0.00	0.00	10,158	54,698	-81	6,750	CPLEX (time)	36	33	31
	20	-	-	-	-	-	-	-	-	-	-
	Min.[2]	0.00	0.00	112	109	-81	2,350		21	26	22
	Avg.	0.01	0.00	17,273	9,042	48	5,853		36	34	30
	Max.	0.07	0.00	limit	54,698	99	21,454		46	57	35

[1] FFC: Facility fixed cost; TrC: Transportation cost; SsC: Safety stock cost. All these costs are expressed as percentage of the objective function value.
[2] Avg., Min. and Max. values are reported over instances that was solved using B&C and CPLEX; limit = 86,400 sec (1 day)