INNOVATIVE OPERATIONS AND NETWORK DESIGNS FOR HIGH-VELOCITY INTRA-CITY COURIER SERVICES

by

OZGUR SATICI

IMAN DAYARIAN, COMMITTEE CHAIR
JOSE DULA
NICKOLAS FREEMAN
XINWU QIAN
TEODOR GABRIEL CRAINIC

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Information Systems,
Statistics and Management Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2024

ABSTRACT

This dissertation introduces a novel network design for intra-city courier services, aiming to improve operational efficiency and service quality of courier companies. Traditional hierarchical network models in courier services mandate all shipments be aggregated at centralized distribution centers for sorting, which can create bottlenecks and delays. In contrast, this study proposes an alternative that utilizes the existing infrastructure of courier package drop-off and pick-up stores dispersed throughout urban areas. This alternative network design reorganizes these stores as mini sorting hubs, enabling the decentralization of the sorting process. The research is presented in three separate articles, each addressing different facets of the proposed network design under various conditions of demand, capacity, and operational constraints.

In the first article, we examine tactical and operational planning for the proposed network structure. Our tactical approach uses a multi-commodity service network design to maximize consolidation and optimize commodity paths, managed through mixed integer programming. We refine these paths in a secondary model to minimize necessary cycles for service guarantees. In operational planning, we adjust our strategies based on short-term demand deviations, enhancing service levels through plans tailored to daily operational specifics.

In the second article, we address the stochastic service network design for an intra-city courier service using a hybrid fleet of contracted and crowdsourced drivers. We strategically acquire capacity at reduced rates considering future demand and adjust dynamically based on actual crowdshipper capacities and spot market conditions. Our modeling approach uses two-stage stochastic programming with advanced decomposition methods, improving efficiency and adaptability to operational data.

In the third article, we focus on service network design challenges for an intra-city express delivery system with specific hub capacity constraints. We model and solve the network design on a time-space framework as an integer program, using commercial solvers for smaller scenarios and a constructive metaheuristic approach for larger cases. This strategy separates the problem into freight routing and vehicle scheduling, iteratively solved to create robust solutions suitable for diverse real-world conditions.

DEDICATION

To my missed mother, father, and my beloved wife, Ayşe.

LIST OF ABBREVIATIONS AND SYMBOLS

B2B	Business-to-Business
B2C	Business-to-Consumer
BBC	Branch-and-Benders-Cut
BD	Benders Decomposition
BDD	Benders Dual Decomposition
C2C	Consumer-to-Consumer
CAP	Cycle-Adding Phase
CD	Contracted Driver
CRP	Cycle Removal Phase
CS	Crowdshipping
DC	Distribution Center
DDD	Dynamic Discretization Discovery
FRP	Freight Routing Problem
FSNDP	Frequency-Based Service Network Design Problem
G	Greedy Approach
ILSM	Integer L-shaped Method
INT-SP	Integer Subproblem
IP	Integer Programming
LP-SP	Linear Programming Subproblem
LTL	Less-than-Truckload
M3	IP-based Alternative Selection Model
M3-G	Hybrid Alternative Selection Approach
MCFND	Multicommodity Capacitated Fixed-Charge Network Design

MIP	Mixed Integer Programming
NDP	Network Design Problem
OD	Origin-Destination
PAM	Partitioning Around Medoids
PASSND	Partially Adaptive Stochastic Service Network Design
PBBC	Branch-and-Benders-Cut with Partial Benders Decomposition approach
PBD	Partial Benders Decomposition
PH	Progressive Hedging
PI	Physical Internet
RMP	Restricted Master Problem
RMP(PBD)	Restricted Master Problem (Partial Benders Decomposition)
SM	Spot Market Driver
SNDP	Service Network Design Problem
SP	Subproblem
SSNDP	Time-Dependent Service Network Design Problem
TS	Tabu Search
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem
VRPOD	Vehicle Routing Problem with Occasional Drivers
VRPTW	Vehicle Routing Problem with Time Windows
VSP	Vehicle Scheduling Problem

ACKNOWLEDGMENTS

Foremost, I would like to thank and express my sincere gratitude to my advisor, Dr. Iman Dayarian, for his uninterrupted support throughout my Ph.D., and for his thoughtfulness, motivation, patience, and immense knowledge. His technical knowledge and guidance had a game-changing effect on my research and technical skills. I will always be grateful for his valuable feedback, patience, understanding, and being an exemplary leader.

I also would like to extend my sincere thanks to my dissertation committee members, Dr. Jose Dula, Dr. Nickolas Freeman, Dr. Xinwu Qian, and Dr. Teodor Gabriel Crainic for their constructive feedback, guidance, and suggestions throughout the dissertation process. I would like to express my appreciation to them for taking the time to understand my research and providing their invaluable expertise in the field.

I would like to thank my former roommate, Emre Kurtoglu, for being a brother to me during my Ph.D. journey. He has been a great company with enormous support and lots of memories.

Finally, I would like to express my deepest gratitude to my parents for their constant love and uninterrupted support and for being patient in my absence over the years.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Since the 1950s, urbanization has significantly reshaped the global demographic landscape, resulting in 33 mega-cities worldwide—a number expected to grow by 2030 (152). This urban shift has not only concentrated populations but also intensified the demand for intra-city courier services, particularly for express deliveries. The increase in global e-commerce has further amplified this demand, nearly doubling the global parcel shipping volume from 87.5 billion in recent years to 159 billion parcels today (148). This growth trend is fueled by the rising popularity of online shopping and the availability of faster delivery services, including business-to-business (B2B), business-to-consumer (B2C), and consumer-to-consumer (C2C) modes. Courier companies are burdened with the increased consumer expectations and competition with other courier companies for fulfilling the most expensive and difficult segment of the e-commerce supply chains: the last mile delivery.

Intra-city courier service operations can be modeled as service network design problems (SNDPs), which specifically focus on designing network structures that efficiently and effectively manage the logistics of services such as courier deliveries. While traditional applications of SNDPs often emphasize long-haul freight operations, the unique demands and constraints of dense urban environments necessitate tailored solutions. This may include services provided by courier companies such as UPS or FedEx, serving as the last-mile delivery operators for large retailers, offering aggressive service guarantees such as same-day or express deliveries, which are the main focus of this dissertation. These short-haul settings require network

designs that accommodate high demand and velocity of services, calling for innovative approaches to manage the intricate logistics of parcel delivery effectively.

Traditional service network design models have relied on hierarchical networks where all shipments are aggregated at centralized distribution centers (DCs) for sorting. Such models, with their reliance on distribution centers typically located on the outskirts of cities, often lead to inefficiencies. This setup is particularly problematic in urban environments, where the close proximity of destinations could otherwise expedite delivery. Moreover, these traditional models struggle to keep pace with the rapid service expectations emerging from the digital economy.

In this research, we propose a decentralized network design for intra-city courier services that utilizes the existing infrastructure of courier package drop-off and pick-up stores as mini hubs. Contrary to the current design of operations, in such a network all hubs serve potentially as origin, destination, sorting, and cross-docking hubs. Based on this proposed network structure, we address the SNDP for intra-city parcel delivery operations in metropolitan areas with a concentrated demand for rapid delivery.

1.1 Contributions

In this dissertation, we proposed innovative solutions to improve the efficiency intra-city courier service operations, particularly focusing on the service network design structures. Our proposed settings differed from traditional hierarchical models by propoisng a decentralized network configuration that utilized existing courier package drop-off and pick-up stores as mini hubs. Throughout our research, we produced three articles, in which we systematically investigated various aspects of intra-city courier service logistics and proposed formulations and solution approaches to address these challenges. These included tactical and operational planning, managing demand uncertainty and crowdshipper availability, and optimizing freight routing and vehicle scheduling decisions while considering practical constraints such as hub capacity limitations.

In the first article, we introduced a rate-based service network design (SND) and proposed a two-level planning framework consisting of a tactical level and an operational level that ensured on-time delivery of the shipments. Shipments with the same origin, destination, and service guarantee were grouped together to form a commodity, while the demand for each commodity was assumed to be given as a rate. For the tactical level planning, we developed a Mixed Integer Programming (MIP) model that assigned each commodity a path and constructed a set of vehicle cycles by minimizing the total fleet size that provided sufficient dispatch frequencies for on-time deliveries. Another MIP then reduced the fleet size by reallocating potential slack times of each commodity path across its arcs. To improve the service levels of each operational period, the operational planning mechanism supplemented the tactical plans according to the specific demand data for the period by designing a set of extra vehicle routes. We conducted an extensive computational study for sensitivity analysis and showed that our results performed better than the state of the art in the literature. Our results indicated that the tactical plan already guaranteed a robust solution with high service levels in most cases, with the operational level adjustments providing improvement in the service levels whenever required.

In the second article, we addressed the stochastic service network design problem of an intra-city courier service provider employing a hybrid fleet of contracted drivers, crowdshippers, and third-party drivers. This study navigated the complexities of uncertain demand and transportation capacity, particularly focusing on crowdshipper availability. We formulated this as a two-stage stochastic model, with the first-stage decisions involving routing of corporate drivers and the second-stage adjusting to the revealed demand and crowdshipper capacity. Our solution approach utilized Benders Decomposition, enhanced by Integer L-Shaped Method and Benders Dual Decomposition techniques, and included innovative strategies like selective subproblems and parallel processing to increase solution efficiency. We also introduced a Partially Adaptive Stochastic Programming approach, allowing for

real-time adjustments to tactical plans based on new information, significantly enhancing operational flexibility and effectiveness.

The third article extended the proposed network design to include specific logistical constraints such as vehicle, hub parking, and storage capacities within an intra-city express delivery network. We developed a comprehensive integer programming model, supported by a metaheuristic that decomposed the problem into freight routing and vehicle scheduling subproblems. These were solved sequentially and iteratively to produce optimized solutions. Our metaheuristic, integrating a memory-based adaptive large neighborhood search for the freight routing and a unified tabu search for vehicle scheduling, demonstrated its capability to handle large instances up to 500 commodities. The methodology's effectiveness was validated through a detailed computational study, confirming the practicality and adaptability of our approach across different urban settings.

1.2 Outline of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, we provide an integrated review of the related works in the literature, for all three studies. We investigate the different variants of SND, crowd-shipping in the parcel delivery services, and the concept of urban logistics.

Chapters 3 through 5 present the three articles that have been studied over the course of these doctoral studies. Chapter 3 presents "Tactical and Operational Planning of Express Intra-city Package Services". Chapter 4 presents "A Branch-and-Benders Cuts Algorithm for a Stochastic Service Network Design with Crowdsourced Capacity". Chapter 5 presents "Service Network Design for Express Intra-city Courier Services". Finally, Chapter 6 provides the "Conclusion and Discussion".

CHAPTER 2

LITERATURE REVIEW

In this section, we provide a of related studies in the literature. We first discuss some of the recent trends in urban parcel delivery systems, and then provide a detailed background to the service network design problem (SNDP).

2.1 Urban Logistics

The conventional urban logistics systems are well studied and have evolved to achieve high service levels. However, with the increasing customer expectations, courier companies struggle with keeping up with those high expectations while maintaining efficiency. Recently, several emerging concepts such as the Physical Internet, hyperconnectivity, and crowdsourced delivery (crowdshipping) are gaining attention, aiming to address some of the challenges in urban parcel delivery. Specifically, the concept of the Physical Internet and its principle of hyperconnectivity aim to revolutionize urban logistics by enabling more collaborative and interconnected logistics operations. Additionally, crowdshipping is becoming popular as a part of urban delivery strategies, utilizing non-professional drivers to enhance flexibility and responsiveness in last-mile delivery.

Physical Internet and Hyperconnectivity The concept of Physical Internet (PI), introduced by (117; 118), and later discussed in an urban logistics setting in (44), describes a framework that creates collaboration among multiple shippers by enabling open asset sharing thereby interconnecting separate transportation modes and nodes (hyperconnectivity) (92).

This concept gained great attention in the literature, summarized in (151) and (126). The network design proposed in this paper is similar to the architecture of a single-tier as well as the interconnected structure of each tier of a two-tier hyperconnected city network (46; 50). A two-tier city logistics system (38; 48) is based on a double layer of city distribution centers, where large distribution centers are located at the outskirts of the urban zone enable the first level of consolidation and coordination activities for long-haul transportation vehicles. Freight at such distribution centers is consolidated into urban vehicles and shipped toward the second set of facilities, located inside the city. The SNDP (35; 162) is the common tool used in these planning processes, by providing the planner with the choice of paths for shipments and the services or resources necessary to execute them. Building such plans involves selecting the services to operate, their schedules, and then executing them by routing shipments through the selected service network.

Crowdshipping Crowdsourcing utilizes the advances in mobile networking, smartphones, and online payment systems to utilize a temporary and task-based workforce, providing an alternative to traditional long-term employment contracts (134; 130; 145; 91; 106). This model has become a new avenue for organizations to capitalize on flexible labor exchanges (146; 59). Specifically, crowdshipping enables ad-hoc drivers to participate in the last-mile delivery of online orders, either partially or fully (8; 11; 57; 109; 147). The growing popularity of this model can be attributed to its economic, environmental, and social benefits (129). Notable real-world implementations include AmazonFlex, Uber Freight, Doordash, Shipt, Instacart, and Postmates.

Despite its advantages, crowdshipping introduces significant challenges, primarily due to the inherent uncertainty in supply, such as the unpredictable availability of transportation capacity. This uncertainty necessitates more complex operations planning (150; 53; 57). To address these challenges and provide a more reliable service, some companies have adopted hybrid delivery fleets. This model combines the low-capital, flexible nature of crowdshipping

with the high-asset, reliable characteristics of owned fleets, as seen in companies like Amazon, Walmart, Veho, and Bringg (57; 17).

Most of the academic research on crowdshipping has been framed within the Vehicle Routing Problem (VRP). Initial studies, such as the one by (8), introduce the concept of the Vehicle Routing Problem with Occasional Drivers (VRPOD), which has been expanded in subsequent research. Papers by (109; 53; 150; 110; 119; 123) explore various extensions like split deliveries, in-store customers, and delivery with time windows using advanced solution approaches such as variable neighborhood search and branch-and-cut algorithms.

A few studies have focused on the SNDP with crowdshipping. These contributions will be discussed in detail in Section 2.3.4.

2.2 Network Design Problem

Network Design Problems (NDPs) are optimization problems focused on configuring the structure and connectivity of networks to achieve specific objectives, such as minimizing costs or maximizing efficiency, while meeting various constraints and service requirements across different domains like transportation, telecommunications, and power systems. NDPs appear in a large number of contexts including transportation, telecommunications, and power systems. In the field of telecommunication, network design models have been used in the design of a local access network with one or two technologies (135; 136), and the design of terminal layout in a centralized computer network (71; 77). In power system applications, network design has been used to plan the transmission system which carries electricity from the generation plants to customer centers (19; 137) and the distribution of energy inside each center (52; 70). Several aspects of transportation planning ranging from strategic capital investments to day-to-day operational scheduling can be represented by NDP models.

2.3 Service Network Design Problem

The SNDP focuses on the tactical planning problem in transportation and involves optimizing the allocation and utilization of resources to meet targeted service levels. The SNDP is a special case of the network design problem (NDP), in which the complexity level of the problem is lower since there is no requirement for balancing the assets at the end of a planning period (14). Early research on SNDP can be traced back to (33) and (64). The SNDP includes tactical plans such as routing decisions, terminal operations, service schedules, resource management, and empty repositioning in various logistics service providers, including railways (32), maritime transportation (27), less-than-truckload (LTL) motor carriers (37), intermodal transportation (43), and freight transportation (36). We are particularly interested in multicommodity service network design, where the goal is to find the least-cost set of routes for a given set of commodities between different pairs of origin and destination nodes.

One of the most common applications of the SNDP is the context of LTL (47; 139; 140). An application of SNDP for LTL carriers is presented in (63). The authors propose a formulation that combines the SNDP and dynamic shipment routing in a multi-commodity context and solves it using a heuristic algorithm. The SNDP has also been applied to air networks, and more specifically for express shipment services (15; 9; 14; 10). The problem includes a fleet of air crafts and a fleet of vehicles such that for the service between an origin and a destination either one of them can be used solely or they can be used together, while a hub-and-spoke network structure is considered. An IP model is formulated in (15) for express shipment service design problem and solved using a column generation-based approach in a reasonable time. (10) create a system to optimize the design of service networks for delivering express packages for UPS. The system simultaneously determines aircraft routes, fleet assignments, and package routings to ensure overnight delivery at a minimal cost. UPS management credits the system with identifying operational changes that have saved several million dollars over the course of two years of implementing the system.

The SNDP is one of the most difficult combinatorial optimization problems (43), due to the complexity of the operations, the high number of decision variables, the relationship among these variables, and the size of the real-life applications, etc. However, there have been considerable advancements in this area in terms of modeling, as well as algorithmic and computational efficiency (84).

The literature on SNDPs is primarily categorized based on the availability of information (deterministic versus stochastic SNDPs) and the approach to modeling the temporal aspect (frequency-based versus time-dependent or scheduled SNDPs). Initially, we will explore frequency-based and time-dependent (scheduled) SNDP literature. Subsequently, we will explore the distinctions between deterministic and stochastic SNDPs, focusing on the various sources of uncertainties that characterize stochastic SNDP.

2.3.1 Frequency-Based SNDP

Frequency-based SNDP (FSNDP) focus on determining the type and frequency of services on freight routes. These models aim to meet traffic flow requirements for a specified service level by optimizing the frequency of vehicle operations. Demand in FSNDP is typically described as flows between various origin/destination pairs, making this variant closer to strategic/tactical planning. Comprehensive reviews of FSNDP can be found in (13; 35; 112; 116). Key formulations in the literature include arc-based (127; 6; 87) and path-based (138; 153) multicommodity capacitated network design.

2.3.2 Time-Dependent (Scheduled) SNDP

Time-dependent service network design problems (SSNDP), also known as scheduled SNDP, explicitly incorporate the temporal dimension by representing demand and activities over time. These models are defined on a time-expanded network, where the time component is discretized, making the problem significantly larger and more complex to solve compared to static models. Time-dependent SNDP is closely aligned with operational planning as it involves scheduling services and coordinating activities within specific time frames.

Examples of time-dependent SNDP include (149), which focuses on large-scale freight transportation problems using a solution approach divided into three parts: network construction, commodity-fleet assignment, and vehicle planning. They employ heuristic approaches, MIP, and constraint programming. (49) assumes deterministic demand and integrates tactical planning with multiple resource allocation at terminals, proposing a metaheuristic that combines column generation, slope scaling, intensification, diversification, and exact optimization methods. (42) integrates strategic resource acquisition and allocation with tactical transportation plans, making strategic and tactical decisions jointly. In (67), resource management is incorporated into tactical planning on a two-tier network structure, with an MILP model solved using Benders decomposition. (82) introduces a novel SSNDP where demand arrival and due times are flexible, proposing a formulation and solution approach based on (21).

Solving larger instances of SSNDP is challenging due to the expanded network size. Network reduction strategies, such as the dynamic discretization discovery (DDD) concept proposed by (21), enable partially time-enabled networks to represent SSNDP models, thus reducing problem size. (81) adapts and enhances the DDD strategy for LTL settings, developing exact and heuristic approaches.

Recent studies in SSNDP, particularly for express shipment services, include (113), which considers the relationship between service commitment prices and customer decisions, aiming to maximize profit by optimizing delivery times, prices, and load plans using a heuristic approach. (144) integrates different planning levels of long-haul and local transportation, applying the dynamic discretization discovery algorithm to solve route-based and arc-based formulations. As an application of SSNDP in city logistics, (143) uses mixed autonomous fleets in the network, solving an IP model with a heuristic approach on a real-world network case study.

2.3.3 Deterministic SNDP

Deterministic service network design problems assume that all parameters are known with certainty. These parameters include demand, travel times, costs, and other operational

variables. Deterministic SNDP models focus on optimizing the allocation and utilization of resources under fixed conditions. Extensive reviews of deterministic SNDPs can be found in (35), (163), and (79).

Deterministic SNDPs are typically used in contexts where demand patterns and other relevant variables can be accurately forecasted or are relatively stable. These models aim to provide optimal solutions for network design, considering fixed inputs. The deterministic SNDP literature includes various formulations and applications, often characterized by their specific focus on long-term strategic and tactical planning.

One of the primary formulations in deterministic SNDP is the multicommodity capacitated network design problem, where the goal is to route multiple commodities between pairs of origin and destination nodes while minimizing costs and respecting capacity constraints. This problem is addressed in works like (127), (6), and (87). These studies develop arc-based and path-based formulations to model the flow of multiple commodities through a network.

Recent studies have focused on integrating asset management with network design. For instance, (4) and (103) propose a branch-and-price framework to solve SNDP with asset management, considering the constraints and requirements of managing assets within the network. (62) addresses the design of balanced service network designs with heterogeneous resources, incorporating the complexities of managing different types of resources within the network. These studies emphasize the importance of considering asset repositioning and resource management in the overall network design.

More recent studies have explored various aspects of deterministic SNDP, including express shipment services. For instance, (113) considers the relationship between service commitment prices and customer decisions, aiming to maximize profit by optimizing delivery times, prices, and load plans using a heuristic approach. (144) integrates different planning levels of long-haul and local transportation, applying the dynamic discretization discovery algorithm to solve route-based and arc-based formulations. In city logistics, (143) uses mixed

autonomous fleets in the network, solving an integer programming model with a heuristic approach on a real-world network case study.

2.3.4 Stochastic SNDP

Stochastic service network design problems incorporate uncertainty in various parameters such as demand, travel times, costs, and vehicle breakdowns. Unlike deterministic SNDPs, where all inputs are known with certainty, stochastic SNDPs aim to enhance the robustness, reliability, and efficiency of network designs by considering probabilistic information and addressing uncertainties. This approach is crucial for real-world applications, where variability and unpredictability are inherent.

The literature on stochastic SNDPs is extensive and covers a wide range of applications and methodologies. Stochastic models often use scenarios to represent different possible states of the world, incorporating these scenarios into optimization problems to find solutions that perform well across various potential future conditions. The goal is to develop plans that are not only cost-effective but also resilient to uncertainties. Various sources of stochasticity are studied in the literature, including demand, travel time, and supply uncertainties. Below, we will discuss the main sources of uncertainty addressed in stochastic SNDP research.

Demand Uncertainty

Demand uncertainty is one of the most extensively studied aspects of stochastic SNDP. Early works such as (108) introduce demand stochasticity using a scenario tree and propose an expected cost-minimizing MIP model solved with a two-stage stochastic programming approach. This model constructs a service network in the first stage and determines the flow using the constructed network and outsourcing options in the second stage. (84) extend this work by developing a variable neighborhood search (VNS) metaheuristic to solve larger, real-life instances, demonstrating significant cost reductions.

More recent studies have focused on integrating additional complexities into the demand uncertainty framework. For example, (159) propose a two-stage robust optimization method

and develop a column-and-constraint generation approach to solve the introduced robust models exactly. This approach outperforms traditional Benders Decomposition in terms of computational efficiency and solution quality. Additionally, (88) present a method for bundling scenarios in a progressive hedging heuristic, using Fuzzy C-Means and Gaussian Mixture Models to calculate the membership score of scenarios to bundle centers, further enhancing the robustness of the solutions.

Travel Time Uncertainty

Travel time uncertainty has also gained significant attention in recent years. This type of uncertainty is critical in transportation networks, where delays and variability in travel times can significantly impact service levels and operational efficiency. (58) address this issue by developing a two-stage model that minimizes both cost and carbon emissions for multimodal transportation networks with fixed rail and maritime schedules. The first stage selects motor-carrier services, while the second stage adjusts costs when delays to upcoming shipments are observed.

Another notable example is the work by (97), which focuses on service network design with quality targets. These targets are defined for the on-time operation of services and the delivery of demand loads to destinations. The authors propose a two-stage model and introduce a progressive hedging-based metaheuristic to solve the problem, emphasizing the importance of meeting service quality targets despite variability in travel times.

Supply Uncertainty

The introduction of supply uncertainty into SNDP models primarily stems from the utilization of crowdshipping. The variability in the availability and capacity of crowdshippers significantly complicates network design, as it introduces a level of unpredictability typically absent in more traditional logistics models. Despite its critical importance, only a few studies have explicitly focused on SNDP under supply uncertainty conditions.

One such study is by (11), which explores dynamic routing to manage the unpredictable availability of crowdshippers. This study integrates real-time adjustments into the SNDP to respond to sudden changes in crowdshipper availability, using advanced probabilistic models to predict and mitigate potential disruptions in service. Another study is from (57), which proposes the use of in-store customers for deliveries from store inventories. This study addresses both supply and demand uncertainties by considering stochastic arrivals of online orders and the availability of in-store customers as potential delivery agents, thereby creating a more flexible and responsive SNDP model. (93) presents a novel approach by incorporating public transportation and crowdshipping into the last-mile delivery process. Their model allows for multiple stopovers and utilizes parcel lockers as hubs, thus integrating the public transport system to enhance the reliability and efficiency of crowdshipping under supply uncertainty. This approach not only addresses the variability in crowdshipper availability but also leverages existing infrastructure to optimize the delivery network.

2.4 Solution Approaches

The solution approaches for SNDPs can be broadly categorized into exact and heuristic methods. Given the NP-hard nature of the MIP formulation of SNDP, research efforts have predominantly focused on crafting heuristics and metaheuristics. This section provides an overview of both types of approaches, exploring their methodologies and relevant studies.

2.4.1 Exact Approaches

Exact approaches aim to find optimal solutions to SNDPs by exploring all possible configurations within the problem's constraints. These methods are typically based on mathematical programming and optimization techniques and are suitable for small to medium-sized problem instances due to their computational complexity. One common exact method is the branch-and-price framework, which dynamically generates paths for commodities and cycles for resources, addressing the complexities of heterogeneous resources and asset repositioning,

as proposed by (4) and (103). (93) address stochastic demand and capacity, developing a path-based two-stage stochastic programming formulation and a branch-and-price algorithm.

Benders Decomposition is another optimization technique that decomposes a problem into a master problem and subproblems, solving them iteratively. (132) provide a comprehensive review of Benders Decomposition, highlighting its application in SNDPs. The Branch-and-Benders-Cut approach generates valid cuts within a single search tree, improving computational efficiency. Recent advancements in Benders Decomposition include the introduction of Partial Benders Decomposition (PBD) by (45), which incorporates explicit information from the scenario subproblems directly into the master problem, and the Benders Dual Decomposition (BDD) method by (131), which reformulates the subproblems by incorporating local copies of master variables.

The dynamic discretization discovery (DDD) algorithm addresses the challenges of time-dependent SNDP by enabling a partially time-enabled network to represent the model, thereby reducing problem size. This method is explored by (21) and further enhanced by (81) for less-than-truckload (LTL) settings.

2.4.2 Heuristic Approaches

Heuristic approaches provide near-optimal solutions to SNDPs by using approximate methods to explore the solution space. These methods are particularly useful for large-scale problems where exact approaches become computationally infeasible. Various metaheuristic methods have been developed to address SNDPs. For example, (74) introduce cycle-based neighborhood structures, while (127) propose a two-phase tabu search metaheuristic for arc-based SNDP. (157) present a three-phase metaheuristic combining path relinking and tabu search. Additionally, (26) introduce a metaheuristic with cutting-plane and variable fixing procedures, and (41) combine column generation and slope-scaling for SNDP with resource constraints. (102) propose a metaheuristic based on tabu search for SNDP with heterogeneous assets, and (160) present a hybrid algorithm with pricing, cutting techniques, and local search for large-scale SNDP scenarios with a heterogeneous fleet. (164) focus on

SF Express's intra-city same-day operations, developing an IP model and three heuristic approaches: an IP-based heuristic, a metaheuristic, and a hybrid metaheuristic. These methods are designed to handle hub capacity constraints and limited loading/unloading capacity.

The incorporation of crowdshipping into SNDP introduces additional complexities related to supply uncertainty. Studies like (8), (109), and (57) explore these aspects, proposing various heuristic methods to integrate crowdshipping into traditional delivery models.

CHAPTER 3

TACTICAL AND OPERATIONAL PLANNING OF EXPRESS INTRA-CITY PACKAGE SERVICES

3.1 Introduction

We address the service network design for intra-city high-velocity parcel delivery operations in metropolitan areas with high demand. Urbanization, the global population shift from rural to urban areas, brought about many highly densely populated cities, especially after the 1950s (161). According to the United Nations (152), the current urban population of 4.2 billion will rise to approximately 2.5 billion by 2050, and the number of mega-cities (cities with 10+ million inhabitants) will increase from the current level of 33 to 43 in 2030. Besides that, the boom in global e-commerce sales boosted the demand for package delivery and also increased the expectations of service quality. Additionally, the effects of the COVID-19 pandemic caused worldwide retail e-commerce sales to grow 20.2% in 2019 and 27.6% in 2020 (51). The demand and the increased speed of delivery services promote each other, thereby increasing the competition among courier companies. This demand is both in the business-to-consumer (B2C) and consumer-to-consumer (C2C) forms. Currently, most major courier companies offer same-day delivery services in areas that have sufficient population density. As the demand and desired speed increase, any improvement in the performance of the delivery operations increases the potential gain.

We focus on the route planning problem in the context of intra-city express courier services. Main courier companies own several stores scattered around big cities (see, for

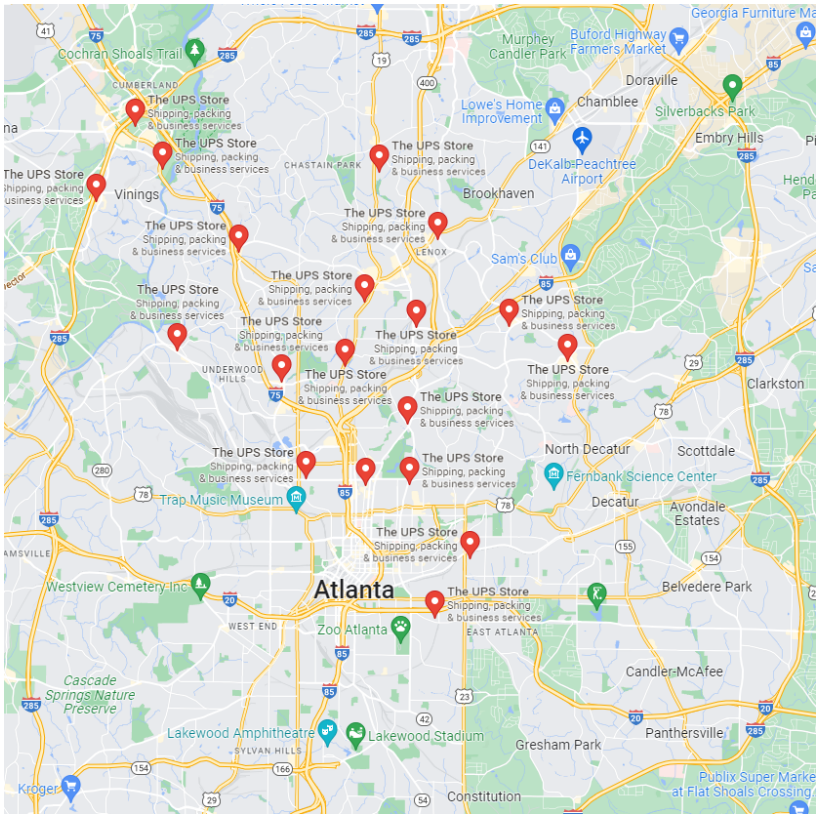
example, the distribution of the UPS stores in Atlanta in Figure 3.2). These stores act as drop-off and pick-up locations, each serving the geographical region around it. The traditional operational network structure of the courier companies in big cities is a hierarchical system with a hub-and-spoke underlying network (124), which requires all shipments to be transferred to distribution centers (DC) for the sorting process (see Figure 3.1a) before being redistributed to their ultimate destination hubs. Often DCs are located on the outskirts of the cities, and therefore, such an organizational structure may lead to longer routes and more time spent at terminals, resulting in potential delivery delays in consolidation-based transportation systems (40).

Figure 3.1: Hierarchical Model versus the proposed Horizontal Model

(a) Hierarchical Model (b) Proposed Horizontal Model

We propose an alternative horizontal network design (see Figure 3.1b) that removes the need to transit intra-city shipments to DCs. Assuming that pick-up and drop-off locations have some basic sorting capabilities, they can be seen as consolidation hubs in the intra-city network. That is, each shipment with its ultimate origin and destination can be mapped into an origin hub and a destination hub in this network. The focus of this paper is on the transportation of the shipments within this network of hubs, which has a strong resemblance to the classical less-than-truckload (LTL) long-haul networks. Although we will present the problem in the context of an intra-city problem, the insights drawn from our study can be extended to applicable long-haul LTL settings. The main complexity in intra-city express delivery services can be attributed to its pressing delivery timelines, resulting in limited opportunities for shipment consolidations, which elevates the need for creative operating strategies that are both cost-efficient and operationally simple to execute. Conversely, in settings with relatively looser timelines (e.g., classical long-haul LTL or next-day delivery intra-city services) the planner enjoys a higher degree of freedom in designing routes relying

Figure 3.2: The distribution of UPS stores in Atlanta [Credit: Google Maps]

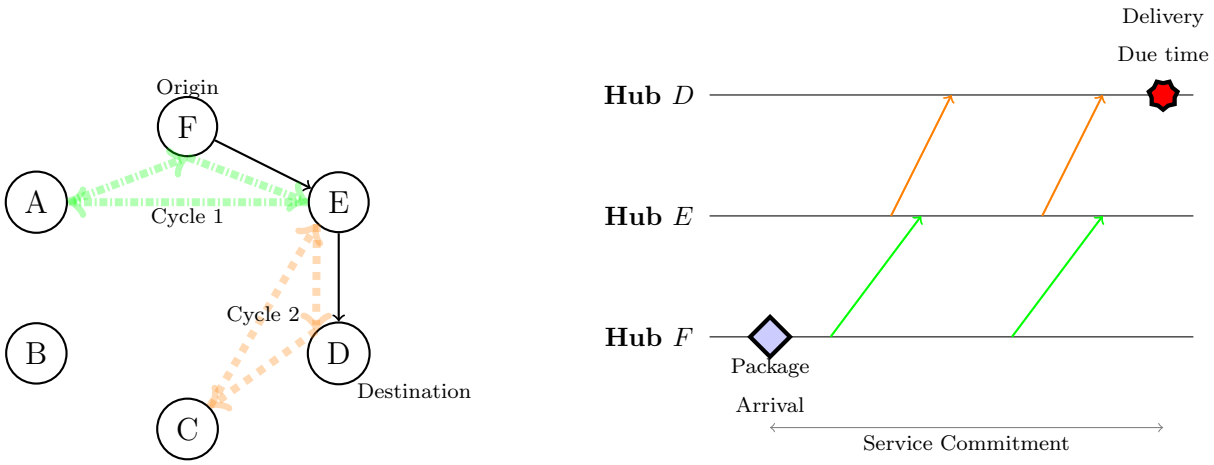


on a high level of aggregating shipments over time or consolidating shipments with different origins-destinations along their routes.

In the proposed horizontal setting, each hub collects the demand of all customers in its covered region, we face a high density and almost stable demand rates between hubs. Therefore, we assume that we are given a *demand rates* between each pair of hubs, representing the average number/volume of shipments received at an origin, destined for a destination in a given time unit, say an hour. The high demand density coupled with the constant arrival of demand over time justifies the adoption of vehicle routes taking the form of cycles operated repetitively and continuously throughout the day. Cyclic routes also regulate the repositioning of empty vehicles, which is a common concern in LTL. The proposed approach aims to determine how the shipments should be routed through the network and how the vehicle cycles should be designed to facilitate these movements and guarantee on-time delivery of the shipments.

Figure 3.3 illustrates a network with six hubs. Our goal is to concurrently identify one path associated with each hub pair with a non-zero demand rate through the network and a set of vehicle cycles that can guarantee on-time delivery of all shipments. For the sake of the example, we focus on the demand originating at hub F with a destination at hub D. One feasible solution would be using the path F-E-D executed using two cycles: Cycle 1 and Cycle 2 (See Figure 3.3a). Continuous execution of each cycle would create fixed-frequency dispatches along its underlying arcs. Figure 3.3b illustrates the consecutive dispatches along arcs (F, E) and (E, D) generated by Cycles 1 and 2, respectively. Shipments associated with an origin-destination pair arrive at a specific rate. Figure 3.3a shows the arrival of one shipment at hub F with a destination at hub D. Given a set service commitment, the arrival time of each individual shipment to the system determines its individual due time at its destination.

Figure 3.3: Illustration of package and vehicle movements in the network



(a) The package movement on the physical network

(b) Scheduled dispatches along the arcs

The decision-making process of the design and operation of the network is executed in two levels: *tactical* and *operational*. The tactical level aims to generate a set of routes (cycles) by taking aggregate (average) demand rate data over a long period of time (e.g., a week or a month) as input. These routes serve as vehicle assignments and long-term vehicle allocation. Depending on the level of variations and seasonality in demand, courier companies may

consider different time periods for their tactical planning. While the aggregate data give a high-level understanding of demand patterns, it may smooth out peaks and valleys that would be observed at a more granular level. Therefore, the operational level planning concerns adjusting the baseline plans obtained from the tactical planning level to the day-to-day fluctuations of demand. The proposed two-level planning approach has several advantages over the existing approaches in terms of computational efforts, practicality, and operational simplicity. As opposed to most existing approaches in the literature, our tactical planning approach does not rely on a time-expanded network. Further, our approach eliminates the need to re-optimizing the whole system from scratch each day, allowing us to address problems of larger sizes in smaller times. Additionally, the tactical plans serve as a guideline to the decision maker for medium-term decisions such as resource acquisition and allocation, fleet sizing, and capacity planning. The operational plans, on the other hand, allow the company to increase the service levels to the desired level. Our contributions in this paper are as follows:

• We propose a novel design of network structure and operations for express intra-city package transportation services. The proposed approach allows the design of tactical and operational plans for vehicle routes in the network ensuring on-time delivery of the shipments. Specifically, our paper contributes to the literature by exploring the dynamics and interplay between these two planning stages in a comprehensive way, from the receipt of demand data to the execution of plans. By presenting both planning stages in a single paper, we can illustrate how upstream decisions (tactical level) affect downstream decisions and operations and showcase the holistic perspective needed for successful logistics management.

• We propose tactical planning by developing a novel mixed integer program (MIP) to assign each commodity a path, and to determine a set of vehicle cycles and dispatch frequencies to provide sufficient capacity in the network for on-time delivery of shipments. Dispatch frequencies are further refined with the goal of reducing the required fleet size in a separate MIP formulation.

• We propose operational planning to adjust the baseline plans obtained in the tactical planning to the specificity of each operational period, by identifying a set of optimal extra vehicle dispatches in the time-space network, and by designing a set of extra vehicle routes in an innovative manner to improve the service level (on-time delivery) of the plans.
• We conduct an extensive computational study with multiple sensitivity analyses to gauge the performance of our proposed approach. We compare the proposed horizontal network configuration with the hub-and-spoke model, which represents a particular type of conventional hierarchical network structure. We also compare our results to those of an existing approach in the literature to prove the superiority of the performance of our algorithm. Our results indicate that the tactical plan already guarantees a robust solution with high service levels in most cases, with the operational level adjustments providing improvement in the service levels whenever required. We test the robustness of our operational plans in case of travel and handling time uncertainty, and show that our solutions provide promising service levels even in those conditions.

The paper is organized as follows. In Section 3.3, we describe the problem setting. In Section 3.4.1, the tactical planning is discussed. Next, in Section 3.4.2, we discuss operational planning. In Section 3.5, a case study is described and the results are analyzed. Finally, in Section 3.6, we present some conclusions and discuss future research.

3.2 Literature Review

We divide the review of the literature into two main topics; (1) the service network design problem, and (2) the urban logistics.

Service network design problem (SNDP) The SNDP focuses on the tactical level of the logistic operations including the routing selections, terminal operations, service schedules, and empty repositioning that exist in the operations of less-than-truckload carriers (34). Particularly, in multi-commodity service network design, the problem includes identifying

a set of minimum-cost routes associated with the commodities with different origin and destination pairs. The earliest studies on the SNDP are (33) and (64). In terms of the assumptions on the information availability, SNDP has deterministic and stochastic variants, while our focus will be on the deterministic variant in this paper.

Excellent reviews of SNDP models and applications appeared in (35) and (162). Also, (79) review the SNDP papers that focus on intermediate facilities. The two main formulations in the multi-commodity capacitated network design literature are the arc-based (127; 6; 87) and path-based (138; 153) formulations.

Recently, studies are focusing more on service network design with asset management. An extensive review of studies implementing design-balance constraints can be found in (5). Also, later (86) and (62) included models explicitly enforcing the empty repositioning. (102) study the design of balanced SND with heterogeneous resources. (104) consider a version of the SND problem with finite heterogeneous resources allocated to the terminals, and the resources have to return back to their home terminals at the end of their service. The authors propose a branch-and-price approach where cycles for resources and paths for commodities are dynamically constructed. In this context, the other recent works in the asset management literature are (26; 101).

(39) consider two types of service network design problems: static SNDP and time-dependent SNDP. Time-dependent SNDPs are usually defined on a time-expanded network to explicitly address the time component in a discretized form. In that case, the problem is called a scheduled service network design problem (SSNDP). (149) focus on a large-scale freight transportation problem. The solution approach is divided into three parts, network construction, commodity-fleet assignment, and vehicle planning. Heuristic approaches, MIP, and Constraint Programming are used in the solution of the decomposed parts. (49) incorporate the strategic decision of resource acquisition and the tactical network design decisions considering the allocation of multiple types of resources to the terminals. The authors propose a metaheuristic that combines column generation, slope scaling, intensification, diversification,

and exact optimization methods. The strategic decisions of resource acquisition and allocation, and the tactical decisions of designing and executing scheduled transportation plans are combined in (42). Their proposed model assumes that the demands are known and makes strategic and tactical decisions jointly. (3) incorporate the hub location and allocation decisions to the problem, jointly considering the transportation costs as well as the travel times. (61) study the logistics service network design for United Nations Humanitarian Response Depot, creating an efficient inventory prepositioning plan for operations in East Africa.

(67) address the tactical planning problem in SSND with resource management in a deterministic setting. In their two-tier network structure, they construct plans for a heterogeneous fleet as well as consolidation decisions considering capacity limitations. An MILP model is proposed for this extended SSNDP, which is then solved using Benders decomposition. (165) focus on optimizing the city parcel delivery operations of a courier company operating in China. They tackle various challenging aspects, including split deliveries and pickups, maintaining cross-trip consistency requirements, and coping with limited unloading capacity at the main hubs. To achieve this, they propose a heuristic solution approach with the goal of minimizing the number of vehicles required for the operations. (82) introduce a novel type of SSNDP in which the arrival and due times of the demand is flexible. A formulation along with a solution approach based on the algorithm presented in (21) is developed.

Considering the fact that the SSNDP models are defined on time-space networks, solving larger instances becomes challenging. Network reduction strategies have been implemented for those types of instances. An example of this is studied in (21). In this paper, the dynamic discretization discovery (DDD) concept is proposed. The DDD enables a partially time-enabled network to represent the SSNDP model, therefore reducing the problem size. (81) adapts and enhances the DDD strategy to develop exact and heuristic approaches for the LTL setting.

The popularity of the SNDP in recent years is visible in the literature. A more recent study on SNDP, specifically for express shipment services appeared in (113), in which the proposed model accounts for the relationship between the price of service commitment and customer decisions. In their model, the maximization of profit is achieved by deciding on the set of optimal delivery times and corresponding prices as well as the load plan. The formulated model is then solved using a heuristic approach. (144) focus on integrating the different planning levels of long-haul and local transportation and propose a route-based and an arc-based formulation. Then, the dynamic discretization discovery algorithm, developed by (21), is applied to solve both of the models. As an application of SNDP in city logistics, (143) utilize mixed autonomous fleets in the network and solve an IP model using a heuristic approach on a real-world network case study.

Urban logistics The conventional urban logistics systems are well studied and evolved for achieving high service levels. With the increasing customer expectations, the efficiency and effectiveness of those systems decline as the services become customized and individualized and the systems become fragmented. The concept of physical internet (PI), introduced by (117; 118), and later discussed in an urban logistics setting in (44), describes a framework that creates collaboration among multiple shippers by enabling open asset sharing thereby interconnecting separate transportation modes and nodes (hyperconnectivity) (92). This paradigm-shifting concept garnered great attention in the literature, summarized in (151) and (126). The network design proposed in this paper is similar to the architecture of a single-tier as well as the interconnected structure of each tier of a two-tier hyperconnected city network (46; 50), which typically involves a much simpler configuration (e.g., no cross-docking) than our setting. A two-tier city logistics system (38; 48) is based on a double layer of city distribution centers, where large distribution centers located on the outskirts of the urban zone enable the first level of consolidation and coordination activities for long-haul transportation vehicles. Freight at such distribution centers is consolidated into urban

vehicles and shipped toward the second set of facilities, located inside the city. The service network design problem (35; 162) is the common tool used in these planning processes, by providing the planner with the choice of paths for shipments and the services or resources necessary to execute them. Building such plans involves selecting the services to operate, and their schedules, and then executing them by routing shipments through the selected service network.

A somewhat related problem setting is the liner shipping network design (27), with the time-constrained liner shipping network design (89) being the most closely related setting to ours due to the strict time limitations considered. Similar to our approach, as will be discussed in Section 4, (89) consider a set of potential paths for each commodity and a set of potential cyclic routes. They make the path and cycle selection in the same model. Contrary to our design, their model includes the time restriction in their pre-generated potential paths via transshipment arcs, while we enforce such restrictions in the model. Different from our problem, they consider a fixed fleet size, and operate on a hub-and-spoke network with the objective of minimizing the various cost components, while the underlying network of our design is not hub-and-spoke and we aim at determining the smallest fleet size required.

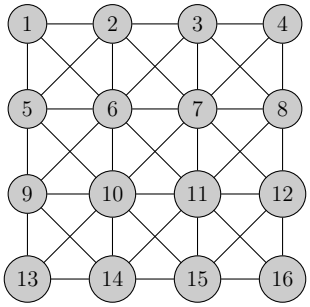
To the best of our knowledge, the only paper addressing a similar intra-city problem with demand rates is (56), in which two separate models are proposed for commodity path selection and vehicle cycle selection in a sequential manner. In this paper, we integrate these two decisions into one model which can prevent potential sub-optimality due to the sequential approach. In Section 3.5, we will compare the performance of our proposed approach to that of (56) to illustrate the potential gain of an integrated decision-making framework. Although we present the algorithmic design and findings of this paper for the type of applications characterized by high-pace demand, short service guarantees, and a large number of OD pairs, they can also be applied to other settings such as long-haul transportation with a longer time horizon.

3.3 Problem Statement

We define our intra-city service network on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, with $\mathcal{V}$ being the set of nodes (hubs) and $\mathcal{A}$ being the set of directed arcs between hubs of the network. This network could be part of a larger network connecting the city to other cities, however, our focus is on the inter- or intra-city freight transportation activities happening within the boundaries of the city. For the sake of presentation, we initially focus on a stylized network, taking the form of a sandbox (see Figure 3.4). Within this network, there is potentially a demand between each pair of nodes: each node can be the origin for shipments and at the same time the destination for other shipments. We define a commodity $w \in \mathcal{W}$ as all the shipments that share the same origin, destination, and service commitment attributes. Parameter d_w represents the average number of shipments associated with a commodity arrived at the commodity's origin per unit of time (e.g., per hour), referred to as the demand rate of commodity w. The deterministic travel time along each arc $a \in \mathcal{A}$ is denoted by t_a. We assume that t_a includes the handling time at the origin of arc a. In the most general setting, there are potentially multiple service offerings for each commodity at different prices. In our setting, since the intra-city service focuses on a relatively limited geographical region, it is assumed that the service provider offers a common service to all origin-destination pairs (note that the model can easily adapt to the case with customized service commitments). Thus, let S be the common service commitment for all commodities offered by the company. That is, all activities including inter-hub transits and intra-hub handling and wait times required from the receipt of a shipment to delivering it to its destination hub cannot take longer than S, otherwise the shipment is considered delayed.

We assume that the hubs in the network are grouped into a set of overlapping clusters $\mathcal{B} = \{1, \ldots, B\}$. The subgraph associated with a hub cluster $b \in \mathcal{B}$ is considered to be a complete graph, while hubs in two different hub clusters are only accessible via paths going through overlapped hubs. In the stylized network in Figure 3.4, each hub cluster takes the form of a square box, such that each group of 4 neighboring hubs with a direct

Figure 3.4: A stylized service network



arc constitutes a hub cluster. However, in a more general setting, these clusters might not have specific shapes or patterns. An algorithmic procedure must be applied to form the hub clusters, as explained further under Hub Clustering part in Section 3.4.1.

Exact commodity volumes (demand rates) may be unavailable much in advance; the actual demand volume information often becomes available only for the near future, (e.g., same or next day). Instead, using historical data, forecast demand under the form of probability distributions may be available. In such a setting, a computationally expensive approach would be to re-optimize the network from scratch upon the reveal of accurate information on a daily basis. This might not be reasonable both from the required computational effort and also from a practical perspective. Further, it is also known that simplicity and consistency of operations (for instance the set of routes drivers need to execute over different days of operations) are usually preferred by companies (95). Additionally, for contractual purposes, the companies may want to avoid plans that are drastically different from one day to the next (54; 55). Therefore, our goal is to design an algorithm that allows us to construct a common skeleton (a base plan) for all daily operations, potentially several days ahead of execution, and minimally adjust it in response to specific demand realizations of each day of operation.

The base plan includes one path for each commodity through the network $\mathcal{G}$ and a series of vehicle routes that guarantee the time- and capacity-feasibility requirements of the commodity paths. Specifically, a path $p \in \mathcal{P}^w$ associated with a commodity $w \in \mathcal{W}$ is a sequence of arcs connecting the commodity's origin to its destination through the network,

where $\mathcal{A}^p$ is the set of arcs in path p, and $\mathcal{P}^w$ is a set of potential "admissible" paths that commodity w can take. In different contexts, admissibility can be defined differently to address the potential limitations of the problem such as the path length, the number of handovers for each commodity, etc.

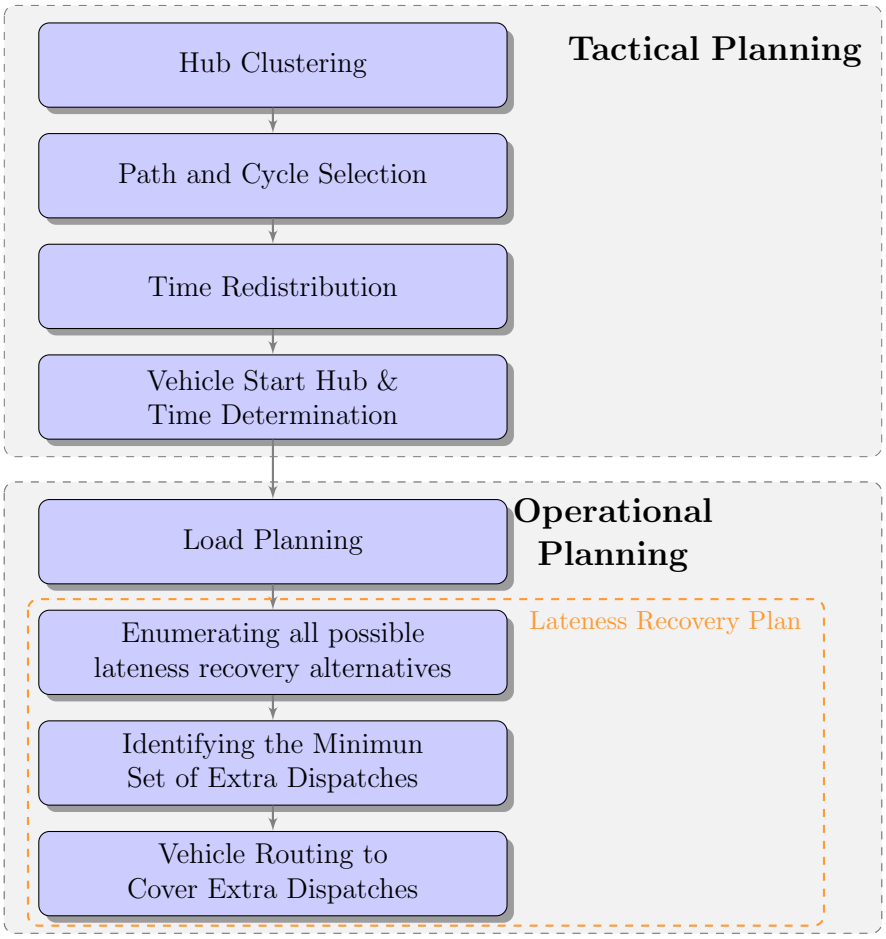
We aim at generating vehicle routes that take the form of a cycle, operated on a continuous basis. To transport commodities through the arcs on their paths, a set of potential cycles $\mathcal{C}^b$ are considered within each hub cluster $b \in \mathcal{B}$. A cycle $c \in \mathcal{C}^b$ is a sequence of arcs in the sub-graph associated with the hubs in b forming a loop and is operated by a vehicle of fixed capacity q that moves continuously along the arcs of the cycle in a pre-specified direction. The length of cycle $c \in \mathcal{C}^b$ is denoted by l_c and corresponds to the sum of all arc lengths constituting the cycle. Binary parameter β_a^c equals 1 if arc a is part of cycle c.

The performance of the delivery network is evaluated using service level metrics, which measure the percentage of packages delivered before their promised delivery times. The base plan aims to achieve high service levels based on average demands, while the daily adjustments are aimed at achieving full-on-time delivery given the observed demand variations compared to the average demand.

3.4 Solution Methodology

To simplify the process of planning and to promote operations consistency, we address the planning of the network organization in two phases. First, a tactical plan is designed based on the aggregate demand data over a relatively long time horizon. By observing demand patterns between each pair of hubs over a long enough time period, one can fit a probability distribution to each hub pair's demand. Our tactical level planning takes the mean of such a distribution as input, and the resulting plan will be the basis of daily operations. Each day of operation can be seen as one potential realization of the ensemble of the hub pair demand distributions with possible deviations from the aggregate demand rates used to generate the tactical plan. Next, at the operational level, the tactical plan is minimally adjusted in

Figure 3.5: Tactical & Operational Planning Diagram



response to the variations of the observed demand versus the aggregate demand. While a tactical plan remains valid as long as the aggregate demand data does not exhibit significant changes, the operational plan will be updated frequently, potentially, on a daily basis. Based on the managerial preferences and the performance of the tactical plans under the specific conditions of the system, operational planning may be omitted. However, it is important to carry out operational planning to ensure 100% service levels are maintained. The sequence of decision-making in tactical and operational planning are illustrated in Figure B.1.

3.4.1 Tactical Planning

The tactical plan can be used for medium-term contractual purposes and serve decisions such as fleet sizing and capacity planning. Additionally, the routes generated through the

tactical planning phase are used as a baseline for daily operations. In general settings, the scope of tactical-level planning is usually less than a year, and decisions are updated based on a weekly, daily, or as-needed basis. These update intervals should be short enough to capture the significant system changes, however, they should be long enough for operational simplicity for the personnel. As shown in Algorithm 1 and Figure B.1, as the first step of our tactical planning, the network is partitioned into a series of overlapping hub clusters, while the sub-network in each cluster is assumed to be a complete graph and inter-cluster movements are limited to paths through the overlapped hubs. The main purpose of hub clustering is to remove arcs that are deemed unpromising from the network while keeping the most important connections among the nodes. The next step consists of identifying one path for each commodity and constructing vehicle routes. Each commodity path may potentially include one or more arcs and may go through multiple hubs before reaching its destination hub. Each vehicle route takes the form of a cycle, operated on a continuous basis. The time feasibility of a commodity path is guaranteed by securing a high enough frequency of vehicle dispatches along the arcs of the path, while such a dispatch frequency is a function of the number and length of vehicle cycles traversing an arc. We introduce a mixed integer programming (MIP) formulation for commodity path and vehicle cycle selection that allows the user to make these decisions concurrently with the objective of minimizing the fleet size under time and capacity constraints (Model 3.1.) In this MIP, the available time (service commitment) is pre-distributed among different arcs of each commodity path to avoid the nonlinearity of the model, which results in reducing the complexity of the problem and consequently accelerating the solution process. Each arc of the network may potentially be covered by multiple cycles. There are two main advantages in vehicles operating routes of type cycles; (1) simplicity of operations, and (2) reducing empty or low utilization movements, while not having to deal with empty repositioning decisions. In order to exploit opportunities of reducing the fleet size by reallocating the potential slack times along the commodity path, in the next step, we introduce a new integer programming (IP) formulation

(Model 3.4.) Finally, in the last step, we develop an algorithm that determines the vehicle start hubs and times, such that the non-homogeneity in the inter-dispatch times due to different cycle lengths are minimized.

Algorithm 1: Tactical Planning

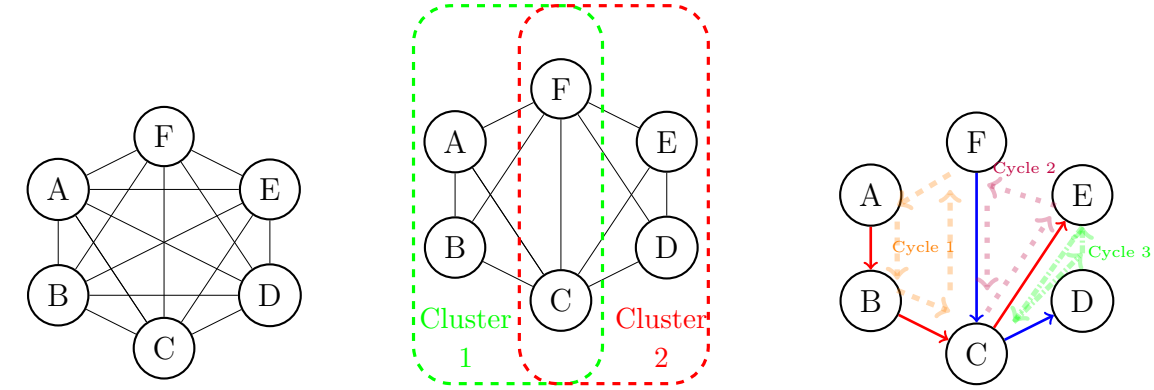
Data: Network and aggregate demand data
Result: Tactical plan
1 **Step 1:** Hub Clustering;
`// Network reduction`
2 **Step 2:** Path and Cycle Selection;
`// Identifying one path per commodity and the required number of vehicle cycles`
3 **Step 3:** Time Redistribution;
`// Reducing the fleet size by reallocating available time to the arcs of commodity paths`
4 **Step 4:** Vehicle Start Hub & Start Time Determination;
`// Set the vehicles' start times and hubs to mitigate inter-dispatch time variability`

The process of hub clustering and selection of both paths and cycles is demonstrated using an example in Figure 3.6. The physical network with 6 hubs is shown in Figure 3.6a. This network is then divided into 2 overlapping clusters in Figure 3.6b (STEP 1 in Algorithm 1). Then we generate a path for each commodity from its origin hub to its destination hub, as well as a set of vehicle cycles conducting commodity movements (STEP 2 in Algorithm 1). Figure 3.6c depicts 3 cycles $(A, B, C, F), (F, C, E), (C, D, E)$ and two commodity paths $(A, B, C, E), (F, C, D)$. The commodity along the path (A, B, C, E) is handed over from Cycle 1 to Cycle 2 before finally being delivered to destination hub E. Similarly, the commodity along path (F, C, D) is transported on Cycle 1 and then passed on to Cycle 3 before reaching its destination.

Hub Clustering

In the stylized network in Figure 3.4, hubs are pre-clustered into boxes. In a generic network, however, such clustering decisions are critical and could significantly affect the efficiency of results. The main purpose of hub clustering is network reduction without significantly sacrificing efficiency. The clusters also serve as boundaries of operating regions by vehicles:

Figure 3.6: The overview of the Path and Cycle Selection procedure



(a) The physical network (b) Partitioning into overlapping clusters (c) Concurrent commodity path and vehicle cycle selections

movements of each vehicle are limited to the cluster of hubs it is assigned to. A hub shared between two or more clusters is considered the channel for inter-cluster freight movements. The premise is that hubs within the same cluster have the opportunity to be connected through direct arcs, consequently minimizing the transfer times between them. We discuss our approach to creating a series of overlapping hub clusters in A.1. The outcome of the hub clustering phase is a set $\mathcal{B}$ of hub clusters.

Path and Cycle Selection Model

We model the problem as a path-based mixed integer programming (MIP) formulation which assigns commodities to one of their potential paths and identifies a subset of candidate cycles in each hub cluster in order to carry out the traversal operations, ensuring the time feasibility of the paths and required capacity along the arcs. For the sake of computational simplicity, among all possible paths, $\mathcal{P}^w$ may include the P shortest paths commodity w can take.

Let X_p^w be a binary decision variable that equals 1 if candidate path $p \in \mathcal{P}^w$ is selected for the commodity w, and 0 otherwise. We assume that the transportation of commodities through their paths is conducted separately on each arc. That is, at the start node of each arc, commodities are loaded into the first arriving vehicle and then unloaded at the end node of that arc. In practice, if two or more consecutive arcs of a commodity path are part of

the same vehicle cycle, the shipments associated with that commodity remain on board the vehicle. However, at the intermediate hub shipments associated with all other commodities onboard the vehicle are unloaded, and potentially new shipments are loaded. This extra loading/unloading time is added to the length of the arc (and consequently the paths and cycles incorporating it.)

Each arc $a \in \mathcal{A}$ can potentially be part of multiple commodity paths. Thus, the flow rate F_a, the total number of shipments going through arc a per time unit, can be calculated as the sum of demand rates of commodities that traverse that arc along their selected paths. The flow rate can be used to determine a lower bound on the number of vehicle dispatches per time unit. The frequency of vehicle dispatches along an arc is also affected by the "tightness" of commodities using that arc along their paths; the more time-constrained a commodity, the larger the required vehicle dispatch frequencies along its path's arcs. Let the decision variable Y_c denote the number of vehicles assigned to cycle $c \in \mathcal{C}^b,\ \forall b \in \mathcal{B}$. The model aims to select the best combination of cycles to operate and the number of vehicles needed on each cycle to conform to both capacity and time constraints.

Shipments move towards their destination one arc at a time according to their commodity paths and wait for the first vehicle dispatch along the next arc. We initially distribute the service commitment time S among the arcs of each commodity path proportional to the arc travel times. That is, the total time that shipments of commodity w according to candidate path $p \in \mathcal{P}^w$ can spend to move along an arc $a \in \mathcal{A}^p$, including the travel time and waiting at the origin hub of the arc is capped at S_a^p, where $S_a^p = St_a / \sum_{a' \in \mathcal{A}^p} t_{a'}$. Thereby, the frequency of vehicle dispatches along an arc $a \in \mathcal{A}^p$ is set such that the sum of waiting time before the next dispatch and the travel time along arc a does not exceed S_a^p. Obviously, if $S_a^p < t_a$, then path p is not considered time-feasible. The available waiting time or *slack time* at each arc can be expressed as $S_a^p - t_a$: The inter-dispatch times of vehicles along arc a must be less than the arc's slack time to ensure time feasibility. A more relaxed variant corresponds to setting the inter-dispatch times of vehicles not to exceed $\alpha(S_a^p - t_a)$ when α is a constant.

We refer to α as the *frequency parameter* and set $\alpha > 1$. Notice that, since a commodity path consists of multiple arcs, while setting $\alpha > 1$ may cause a delay along one arc of the path, this delay may be recouped along some other arcs of the path. The proposed path and cycle selection model takes the form of the following MIP.

$$\text{Min} \quad \sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b} Y_c \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}^w} X_p^w = 1, \qquad w \in \mathcal{W}, \tag{3.1b}$$

$$F_a = \sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}^w} d_w X_p^w, \qquad w \in \mathcal{W}, p \in \mathcal{P}^w, a \in \mathcal{A}^p, \tag{3.1c}$$

$$F_a \leq \sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b} \frac{\beta_a^c Y_c q}{l_c}, \qquad a \in \mathcal{A}, \tag{3.1d}$$

$$\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b} \frac{Y_c \beta_a^c}{l_c} \geq \frac{1}{\alpha(S_a^p - t_a)}, \qquad p \in \mathcal{P}^w, a \in \mathcal{A}^p, \tag{3.1e}$$

$$X_p^w \in \{0,1\}, \qquad p \in \mathcal{P}^w, w \in \mathcal{W}, \tag{3.1f}$$

$$Y_c \in \mathcal{Z}_+, \qquad c \in \mathcal{C}, \tag{3.1g}$$

$$F_a \geq 0, \qquad a \in \mathcal{A}. \tag{3.1h}$$

In Model (3.1), the objective function (3.1a) aims to minimize the total number of vehicles needed to fulfill on-time delivery of all commodities. Since the vehicles are assumed to be running throughout the day, the cost minimization objective is reduced to minimizing the fleet size. Constraints (3.1b) guarantee that exactly one path among the admissible paths of each commodity is selected. Constraints (3.1c) define F_a, the flow rate on each arc a of the network given commodity path selections. The flow rate on a given arc is the sum of the demand rates of the commodities whose assigned path contains that arc. Constraints (3.1d) are the capacity constraints and ensure that the flow rate on each arc is not greater than the capacity per time unit made available by vehicles traversing that arc. Note that $1/l_c$ represents the frequency of dispatches along arcs of cycle c, and therefore, Y_c/l_c and $Y_c q/l_c$

represent the number of vehicle dispatches per unit time along the arcs of cycle c provided by vehicles operating that cycle, and the capacity made available along such arcs by vehicles operating cycle c, respectively. Constraints (3.1e) are the time constraints, which guarantee that the commodities do not wait at nodes along their path more than their slack time times the frequency parameter $\alpha \geq 1$. In other words, if $\alpha = 1$, the vehicle dispatch frequency along an arc must be high enough to ensure on-time delivery of the commodity with the tightest slack time that uses that arc. Finally, constraints (3.1f)-(3.1h) define the type and domain of the model's variables. Model (3.1) is a tool to generate a baseline tactical plan that determines one path per commodity and a minimal fleet of vehicle cycles to execute movements of shipments along the arcs of the network to ensure time and capacity feasibility. However, note that constraints (3.1d) and (3.1e) are based on the simplifying assumption that once the number of vehicle dispatches along an arc is determined, such dispatches are performed in a way that the inter-dispatch times are distributed evenly. As we will discuss in §4, this is an ideal situation, which in a setting in which an arc is covered by multiple cycles of different lengths, such homogeneity of inter-dispatch times may not be achievable. We elaborate on this issue in the subsequent sections and propose an approach to alleviate the potential shortcomings.

As such, Model (3.1) relies on two main simplifying assumptions: (a) service commitment time S is distributed among the arcs of commodity paths proportionally w.r.t. their lengths, and (b) inter-dispatch times along each arc of the network are distributed homogeneously given the number of cycles covering the arc. Next, we discuss how starting from a baseline solution obtained from Model (3.1), we explicitly relax such assumptions and alter the solution accordingly.

Time Redistribution

Model (3.1) relies on a proportional partitioning of the service commitment S among arcs of each commodity path. This simplifying assumption allows us to avoid nonlinearity in Model

(3.1) due to making path selection and time allocation decisions concurrently, and potentially accelerates the solution process. However, this assumption may result in sub-optimality of the solutions. Note that an arc a is possibly used by multiple commodity paths, while the vehicle dispatch frequency along a is derived from the most time-constrained commodity path among those. For all other commodity paths using a, arc a may have a higher dispatch frequency than the one they require. This creates an opportunity to reallocate the time saved along a among the other arcs of such commodity paths, potentially allowing fleet size reductions. Therefore, we develop a second IP model which tries to identify such opportunities given a solution of Model (3.1). Let $\hat{S}_a$ be the maximum time taken for shipments arriving at the start point of arc a to be dropped off at the end point of arc a, given the current vehicle dispatch frequency along a. Derived from constraints (3.1e) of Model (3.1), $\hat{S}_a$ can be calculated as

$$\hat{S}_a = \frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}} + t_a \tag{3.2}$$

where y_c is the optimal solution of Model (3.1). The calculation of $\hat{S}_a$ is as follows: $\frac{y_c\beta_a^c}{l_c}$ is the added dispatch frequency along arc a if y_c vehicles operate on cycle c. Thereby, $\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}$ is the total frequency of all vehicles covering arc a, and consequently, $\frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}}$ is the expected waiting time at the origin of arc a, resulting in $\frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}} + t_a$ being the total (waiting + travel) time of a commodity along arc a. Let Δ^p be the total slack time on path p, i.e., $\Delta^p = S - \sum_{a\in\mathcal{A}^p}\hat{S}_a$. The dispatch frequency on arc a can be reduced only if the total slack time Δ^p of all commodity paths $p : a \in \mathcal{A}^p$ can compensate for the reduction. Considering that elimination of a cycle drops the frequency on all of the arcs of the cycle, the same condition must hold for all of the arcs of a cycle to be considered as a candidate for removal.

Let $\mathcal{A}^p$ denote the set of arcs of the commodity path p that is selected in Model (3.1). Also, let $\mathcal{A}^*$ denote the set of arcs of the network that are used by at least one of the selected commodity paths, i.e., $\mathcal{A}^* = \cup_{p\in\mathcal{P}}\mathcal{A}^p$. Set $\mathcal{U}_a^p$ contains all cycle combinations currently covering arc $a \in \mathcal{A}^*$, and $n_{c,u}$ indicates the number of cycles of type c included in combination $u \in \mathcal{U}_a^p$ (each cycle combination is a set of one or more cycles among those selected in Model (3.1), i.e., $y_c \geq 1$). Parameter $\theta_{a,u}^p$ denotes the minimum extra time required to be allocated to arc $a \in \mathcal{A}^p$ so that cycles in combination $u \in \mathcal{U}_a^p$ can be removed without causing lateness and is calculated as

$$\theta_{a,u}^p = \frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c} - \sum_{c\in u}\frac{n_{cu}}{l_c}} - \frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}} \tag{3.3}$$

The calculation of $\theta_{a,u}^p$ is as follows: $\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}$ is the total frequency of all vehicles visiting arc a, and $\sum_{c\in u}\frac{n_{cu}}{l_c}$ is the reduction in frequency on arc a, if all of the cycles in combination u are removed. Therefore, $\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c} - \sum_{c\in u}\frac{n_{cu}}{l_c}$ is the reduced dispatch frequency along arc a, and its inverse is the new inter-dispatch time, which is longer than the original inter-dispatch time $\frac{1}{\sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b}\frac{y_c\beta_a^c}{l_c}}$. As a result, the difference between the two terms is the minimum amount of extra slack time required on arc a for all cycles in combination c to be removed.

Then, an IP model can be constructed and solved to redistribute Δ^p across the arcs in $\mathcal{A}^p$, maximizing the opportunities for removing cycles. Let $Z_{a,u}^p$ be a binary variable taking the value 1 if cycle combination $u \in \mathcal{U}_a^p$ is selected for removal. Integer variable $\bar{Y}_c$ indicates the number of vehicles on cycle c removed. Parameter f_a is the output of the decision variable F_a in Model (3.1). The time redistribution model takes the following form.

$$\text{Max} \quad \sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b} \bar{Y}_c \tag{3.4a}$$

$$\text{s.t.} \quad f_a \leq \sum_{b\in\mathcal{B}}\sum_{c\in\mathcal{C}^b} \frac{\beta_a^c q(y_c - \bar{Y}_c)}{l_c}, \qquad \forall a \in \mathcal{A}, \tag{3.4b}$$

$$\sum_{a\in\mathcal{A}^p}\sum_{u\in\mathcal{U}_a^p} \theta_{a,u}^p Z_{a,u}^p \leq \Delta^p, \qquad \forall w \in \mathcal{W}, \quad p \in \mathcal{P}^w, \tag{3.4c}$$

$$\sum_{u\in\mathcal{U}_a^p} Z_{a,u}^p \leq 1, \qquad \forall w \in \mathcal{W}, \quad p \in \mathcal{P}^w, \quad a \in \mathcal{A}^p, \tag{3.4d}$$

$$\bar{Y}_c \leq \sum_{u\in\mathcal{U}_e^p} \beta_a^c n_{c,u} Z_{a,u}^p, \quad \forall b \in \mathcal{B}, c \in \mathcal{C}^b, p \in \mathcal{P}, a \in \mathcal{A}^p \cap \mathcal{A}^c, \tag{3.4e}$$

$$\bar{Y}_c \in \mathcal{Z}_+, \quad \forall c \in \mathcal{C}, \tag{3.4f}$$

$$Z_{a,c}^p \in \{0,1\}, \quad \forall b \in \mathcal{B}, \quad c \in \mathcal{C}^b, \quad p \in \mathcal{P}, \quad a \in \mathcal{A}^* \tag{3.4g}$$

In Model (3.4), the objective function (4.2a) maximizes the total number of cycles removed. Constraints (4.2b) enforce a lower bound on the cycle frequencies after the removal of the selected cycle combination. Constraints (4.2c) ensure that the total slack time required for the removal of the selected cycle combination in each path is less than the total available slack time in that path. Constraints (4.2d) ensure that at most one cycle combination for each arc $a \in \mathcal{A}^p$ for each path $p \in \mathcal{P}$ is removed. Constraints (4.2e) guarantee that the number of cycles of type c removed is capped by the number of such cycles included in the selected combinations to remove associated with the arcs of cycle c. If solved to optimality, Model (3.4) allows us to redistribute the available time S along the arcs of commodity paths, given a set of such paths. An effective time redistribution maximizes the opportunity to reduce the required fleet size while maintaining the time feasibility of the schedules. Depending on the network size, the size of set $\mathcal{U}_a^p$ might be quite large, affecting the tractability of Model (3.4). To address this issue, an upper bound on the number of cycles per cycle combination is used to limit the number of considered cycle combinations. Let κ be the maximum number of cycles per combination that is included in the set $\mathcal{U}_a^p$. For example, if $\kappa = 2$, then only

the cycle combinations that contain one or two cycles are passed to Model 3.4. Although this limitation might restrain some potential improvements in the model, it decreases the problem size significantly. We study the effect of κ on the improvement results in a sensitivity analysis in Section 3.5.

Vehicle Start Hub and Start Time Determination

The time constraints (3.1e) in Model (3.1) are structured upon the average number of vehicle dispatches per time unit (say per hour), and assuming that the inter-dispatch times along an arc are distributed evenly. However, in a setting where the minimum required number of dispatches per time unit is guaranteed by vehicles operating on cycles of different lengths, such homogeneity of inter-dispatch times may not be perfectly achieved.

This is mainly problematic in situations where the actual inter-dispatch times along an arc become longer than the allowable slack time of a commodity path that uses that arc, consequently resulting in late deliveries of shipments received during certain time periods of a day. Thereby, relying on this assumption might decrease the operational service level.

We address this phenomenon both in tactical and operational planning. In the tactical planning, given the set of vehicle cycles prescribed by Model (3.1), we try to minimize the frequency of occurrences of inter-dispatch times exceeding the maximum allowable length. This is done by greedily identifying the start time and start hub of each cycle at the beginning of the day. As we will discuss in Section 5, the remaining cases of violation of the allowable inter-dispatch times are tackled at the operational planning by potentially adding extra dispatches at strategically chosen locations/times, given the specificity of each day of operation.

In an environment where the required vehicle dispatch frequencies on arcs are generated using vehicles that operate on cycles of different lengths, inter-dispatch time variability is inevitable. Nevertheless, the impact of such variability can be minimized through a series of well-chosen start times and start hubs for the vehicles. Each vehicle assigned to a cycle

can start its daily operations after the business hours start time, at a certain hub of its cycle. To determine the vehicles' start hubs and start times, we first work out the maximum allowable inter-dispatch times for each arc of the network, denoted by $\delta_a, a \in \mathcal{A}$, with $\delta_a = \min_{p|a\in\mathcal{A}^p} \{S_a^p - t_a\}$, where S_a^p is the allotted time to arc a with respect to path $p : a \in \mathcal{A}^p$ post Model (3.4). δ_a is a metric to measure the level of "tightness" of an arc: a smaller δ_a represents a tighter arc. We process the arcs of the network in increasing order of δ_a. For each arc a, we set the dispatch times of vehicles that cover a, if they are not already fixed for another arc processed before. We schedule the departure times of these vehicles such that they all occur after the start of business hours and the first inter-dispatch times equal δ_a. If for a given arc, some of the vehicle dispatches have been scheduled (when processing another arc with a higher priority), we find all of the fixed vehicle dispatches along the arc throughout the considered time horizon and insert an unscheduled vehicle whenever an inter-dispatch time exceeds δ_a. If we do not encounter any violation of the required inter-dispatch for the arc being processed, we leave these vehicles unscheduled so that they can be prioritized for another arc. We continue this scheduling process until all of the used arcs of the network are considered.

Once the dispatch time of a vehicle is set based on one of its cycle's arcs, the start time and start hub of the vehicle are determined. This is done by identifying the earliest time during the business hours and its corresponding hub along the cycle at which the start of operation would allow the vehicle to arrive on time at the arc it was fixed to.

3.4.2 Operational Planning

Operational planning aims at adjusting the tactical plan to the specific needs of each day of operation, assuming that accurate demand information becomes available ahead of the full execution period. The operational planning mechanism receives the tactical plan as well as the demand data for a given day of operation as input and prescribes a minimal set of additional vehicles to the plan to guarantee full-on-time delivery. In fact, due to the assumption of homogeneity in the inter-dispatch times, and variations in demand realizations

versus the expected demand, the baseline plan may fail to guarantee full on-time delivery of all shipments. Algorithm 2 provides a high-level description of operational planning. At the operational level, as shown in Figure B.1, we first solve the load planning problem given the baseline plan and the observed demand. Next, we identify time intervals associated with each commodity during which received shipments will reach their destination late, and determine a minimal set of extra dispatches to recover such lateness. We then formulate the problem of covering the chosen extra dispatches as a vehicle routing problem with time windows (VRPTW) (Section 5).

Algorithm 2: Operational Planning

Data: Tactical plan and actual demand data
Result: Operational plan
1 **Step 1:** Load Planning;
`// Assigning shipments to available vehicle dispatches and identifying late intervals.`
2 **Step 2:** Lateness Recovery Plan;
3 **Step 2.1:** Enumerating all possible lateness recovery alternatives;
`// Identifying all alternative paths to recover each late interval.`
4 **Step 2.2:** Identifying the Minimum Set of Extra Dispatches;
`// Selecting a subset of alternative paths associated with all late intervals.`
5 **Step 2.3:** Vehicle Routing to Cover Extra Dispatches;
`// Formulating and solving the problem of covering the extra dispatches as a VRPTW.`

Load Planning

Once the vehicle cycles, their start hubs, and start times are determined, one can construct a time-space network representing all temporal movements in the system. Figure 3.7a depicts a schematic time-space network illustrating a snapshot of dispatches along the commodity path from hub A to hub E, via hubs B, C, and D.

Using such a time-space network, associated with each commodity, one can identify time intervals during which shipments received at the origin hub of the commodity will not reach their destination on time. A late delivery could occur due to one of the two reasons, both being byproducts of inter-dispatch time variability: (1) The alignment of next dispatches along the arcs of the commodity is in such a way that the total travel time and waiting time

at the hubs exceeds S; or (2) The required capacity on one or more vehicle dispatches along the arcs of the commodity path exceeds the vehicle's capacity. Note that each dispatch along an arc may potentially serve multiple commodities. Thus, in cases where the capacity of a vehicle is not sufficient to accommodate the entire outstanding quantity, load planning rules involving capacity allocation are implemented.

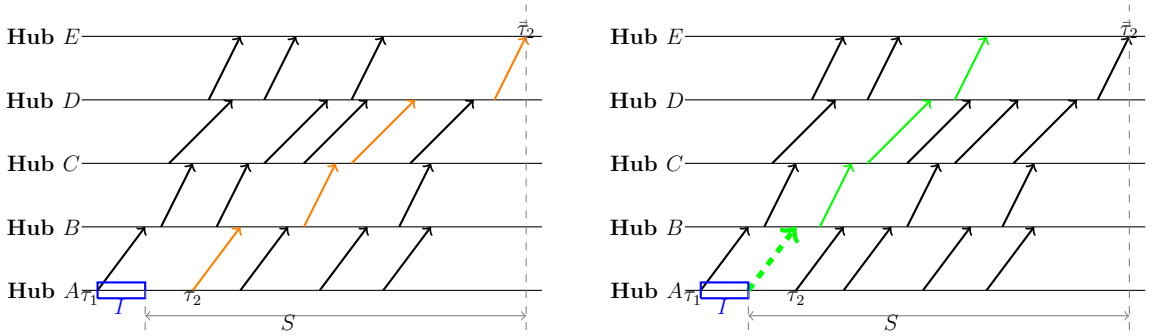
We calculate a capacity usage percentage cap for each commodity on vehicle dispatches that comes into effect if the outstanding quantity to be transferred along an arc exceeds the vehicle capacity. The capacity cap associated with a commodity w on vehicles traversing arc a is calculated as d_w/f_a, where d_w and f_a correspond to the demand rate of commodity w that uses arc a, and the total flow rate along arc a, respectively. Note that f_a is an output of Model (3.1). To minimize the chance of lateness, shipments arriving at the origin hub of each commodity are loaded on the vehicles in a first-in-first-out fashion until the commodity cap is reached.

Late time intervals, periods during which shipments arrived at the origin hub of a commodity will reach their destination late, are identified as follows. Let τ_1 and τ_2 be two consecutive dispatch times along the first arc of a given commodity path, and $\bar{\tau}_2$ be the arrival time of the fastest existing path at the commodity destination with the first arc dispatch at τ_2 (the orange path in Figure 3.7a). Shipments associated with the considered commodity that arrive within interval $I = [\tau_1, \min\{\bar{\tau}_2 - S, \tau_2\}]$ will reach the commodity destination late. Note that if $\bar{\tau}_2 - S \leq \tau_1$, all shipments received within $[\tau_1, \tau_2]$ will be on time unless the vehicle capacity cap constraint prevents it. Using this procedure, all late time intervals associated with all commodities can be identified.

Lateness Recovery Plan

The outline of the lateness recovery plan is presented in **Step 2** of Algorithm 2, as well as in Figure B.1. Late time intervals identified in the load planning phase can be prevented by adding extra dispatches along one or more arcs of the corresponding commodity paths at

Figure 3.7: Lateness recovery plan



(a) Identifying late time intervals

(b) Additional dispatches for lateness recovery

specific times. For example, the additional dispatch between hubs $(A - B)$ in Figure 3.7b, shown as a dashed arrow, allows the shipments arriving during time interval I to be shipped on earlier dispatches on hubs $(B - C - D)$ (shown as the green route), thereby recovering the lateness. In **Step 2.1** of Algorithm 2, all of the alternative paths for each late time interval are enumerated. The detailed description of our proposed algorithm to efficiently enumerate all possible alternatives associated with a late time interval is provided in A.2. Then In **Step 2.2** of Algorithm 2, we aim at identifying the minimum number of extra dispatches that can be added to recover late time intervals. Given a late time interval associated with a commodity, there may be multiple alternative sets of extra dispatches that can be considered to recover all the associated lateness. At least one of the alternatives associated with each late time interval should be selected in order to prevent lateness. Since an arc in the network is potentially used by more than one commodity path, an extra dispatch along a given arc may potentially contribute to recovering multiple time intervals with late shipments, i.e., a potential extra dispatch may be part of several alternative sets associated with different commodities. Moreover, associated with each extra dispatch, there is a time window during which the dispatch can occur to help recover the corresponding lateness. Once we have identified the additional dispatches and their corresponding time windows, in **Step 2.3** of Algorithm 2, we approach the task of constructing vehicle routes to cover them as a vehicle

routing problem with time windows (VRPTW), aiming to minimize the number of additional vehicles needed.

Identifying the minimal set of extra dispatches

For each late time interval, satisfying at least one of the alternatives enumerated would be required to recover the lateness. Satisfying an alternative is performed by executing its extra dispatches. It is conceivable that a set of alternatives belonging to multiple different late time intervals necessitate adding an extra dispatch on the same arc at almost the same time. In such a case, adding one extra dispatch on a specific time window covers multiple late time intervals. As a result, late time intervals can be recovered by a smaller number of extra dispatches and thereby a smaller number of vehicles.

In this section, we develop one heuristic approach for selecting one alternative for each late time interval in a greedy fashion to decrease the total length of extra dispatches needed. We also develop an IP formulation to solve the same problem. Note that the greedy approach can be used as a standalone approach to identify the set of alternatives selected to recover all late time intervals. Alternatively, the IP model can be warmstarted with the solution of the greedy approach for potential further improvements.

Greedy Alternative Selection Approach In the greedy approach, the idea is to prioritize the alternatives with less number of extra dispatches as well as extra dispatches on shorter arcs. Associated with each late interval, we identify the alternative whose total length of its extra dispatches is the minimum.

Let $\mathcal{I}$ be the set of late time intervals, and the set Λ^i be the set of alternatives for late time interval i in the baseline plan. Each $\lambda \in \Lambda^i$ is an alternative that recovers late time interval $i \in \mathcal{I}$, and $\mathcal{D}^\lambda$ is the set of extra dispatch arcs in alternative λ. Each extra dispatch $d \in \mathcal{D}^\lambda$ has a time window associated with alternative λ. For each late time interval $i \in \mathcal{I}$, the greedy approach selects alternative $\hat{\lambda} \in \Lambda^i$ where $\hat{\lambda} = \arg\min_{\lambda \in \Lambda^i}(\sum_{d \in \mathcal{D}^\lambda} t_d)$.

IP-based Alternative Selection Approach In the IP-based alternative selection approach, the time is discretized by collecting all of the extra dispatch time windows start and end time points associated with alternatives in $\cup_{i\in\mathcal{I}}\Lambda^i$. Let $\mathcal{T}$ be the set of such time points. Let binary parameter θ^t_{da} be 1 if adding a dispatch along arc at time point $t \in \mathcal{T}$ would satisfy the requirement of extra dispatch d for $i \in \mathcal{I}, \lambda \in \Lambda^i, d \in \mathcal{D}^\lambda$.

Let D^t_a be a binary decision variable that equals 1 if a dispatch is added to the time window $[t, t+1]$ along arc a, and $C_d = 1$ if extra dispatch $d \in \lambda \in \Lambda^i$ is covered. Also, let the binary variable $B_\lambda = 1$, if all extra dispatches in $\mathcal{D}^\lambda$ for a given $\lambda \in \Lambda^i$ are covered.

$$\text{Min} \quad \sum_{a\in\mathcal{A}^*}\sum_{t\in\mathcal{T}} t_a D^t_a \tag{3.5a}$$

$$\text{s.t.} \quad C_d \le \sum_{t\in\mathcal{T}} \theta^t_{da} D^t_a, \qquad \forall i \in \mathcal{I}, \quad \lambda \in \Lambda^i, \quad d \in \mathcal{D}^\lambda \quad a \in \mathcal{A}^*, \tag{3.5b}$$

$$B_\lambda \le C_d, \qquad \forall i \in \mathcal{I}, \quad \lambda \in \Lambda^i, \quad d \in \mathcal{D}^\lambda, \tag{3.5c}$$

$$\sum_{\lambda\in\Lambda^i} B_\lambda \ge 1, \qquad \forall i \in \mathcal{I}, \tag{3.5d}$$

$$D^t_a, C_d, B_\lambda \in \{0,1\} \tag{3.5e}$$

The objective function (3.5a) minimizes the total length of chosen extra dispatches. Constraints (3.5b) allow us to identify all extra dispatches that are covered if a dispatch along arc a at a given time point is added. Constraints (3.5c) ensure that an alternative is selected only if all of its dispatches are covered. Constraints (3.5d) enforce the need to satisfy at least one alternative per late time interval. Constraints (3.5e) define the domain and type of the variables of the model. If an extra dispatch along an arc is included in multiple selected alternatives, the timing of the extra dispatch would correspond to the intersection of the time windows associated with such alternatives.

Vehicle Routing to Cover Extra Dispatches

After determining the extra dispatches and their time windows, we formulate the problem of

constructing vehicle routes to cover such dispatches as a vehicle routing problem with time windows (VRPTW), with the goal of finding the minimum number of vehicles required. In such a setting, a fleet of vehicles departs from a single depot to cover the required extra dispatches within their time windows. In this VRPTW setting, the extra dispatches represent the nodes to visit, while the length of an extra dispatch represents the service time of such a node. The time window of an extra dispatch represents the node's time window. The location of the depot with the maximum traffic is selected as the depot location. The distance matrix is asymmetric, where the distance between two nodes v and v' corresponds to the travel time between the end hub of dispatch v and the start hub of dispatch v'. Also, the capacity of the vehicles in the VRPTW is assumed to match the rest of the fleet.

We solve this VRPTW using a Tabu Search Algorithm (75; 30; 23). Tabu Search is a metaheuristic neighborhood search method mainly used in combinatorial optimization. It starts with a potential solution and explores the search space by moving to the best solution in the neighborhood of the current solution iteratively. Anti-cycling rules are applied using a tabu list that keeps the moves in the latest iterations in memory and prohibits them from repetition. Implementing the tabu search algorithm to the VRWTW problem allows us to efficiently design routes that cover potentially multiple extra dispatches throughout a day. Covering all the required extra dispatches within their specified time windows guarantees a %100 service level, i.e., %100 on-time delivery across the network.

3.5 Computational Study

To assess the performance of our proposed methodology, we perform a series of computational tests on two series of instances. We first introduce the set of instances considered. Next, we run a comprehensive set of analyses on one set of instances to evaluate the strength and limitations of our proposed approach. Next, we compare the performance of our proposed algorithm to the closest setting from the literature on the second set of instances. Finally, we compare the performance of our horizontal network design with that of hub-and-spoke as

a special case of hierarchical network design to show where our design can outperform the traditional hierarchical network design.

3.5.1 Instances

Two groups of instance classes are considered. The first group focuses on a case study of the city of Chicago, a highly populated metropolitan area. We examine the network of 48 FedEx stores in the city, treating each store as a hub. All our sensitivity analyses are run over these instances. The second group uses the same data set and parameters from (56) that we use to compare the performance of our approach against.

Chicago Case Study Data and Instance Generation

The FedEx store locations are filtered from the list of stores given on the FedEx website: `https://local.fedex.com/en-us/il/chicago/`. Only the 48 stores that currently can offer pick-up and drop-off services and are located in the Chicago city boundaries are considered. Google API service is used to get the real road distances between nodes and an approximate 30 km/h vehicle speed in the city limits independent of the time of the day is assumed for travel times. Additionally, to account for the handling time at each node, 0.2 hours is added to the travel time of each arc. The handling time is considered fixed for each node partly because of the fixed time spent on paperwork, parking, etc., and partly for simplification.

Demand Generation The generation of demand instances is a two-phase process. First, a function that generates the demand rate between hub pairs is defined. The function considers the population, median income level, number of hubs, number of FedEx warehouses, and number of distribution centers of major retailers such as Amazon, Walmart, etc. The population per hub in the region and the proximity to the distribution centers increase the demand rate proportionally. In the demand rate function, we look at the population on a zip code area level and we consider 20 major distribution centers consisting of 4 Amazon

fulfillment centers, 8 FedEx warehouses, and 8 other retailer distribution centers in the Chicago area. For each origin and destination hub, the DCs in a 10-mile radius are counted, then, each DC is assigned a weight based on its type: Amazon, FedEx, etc. The DC's around the origin hub are assigned with higher weights than the DCs around the destination, since the former is proportional to the number of online orders, whereas the latter is proportional to the number of order returns. Thereby, the demand rate function captures the population per hub in the area, online orders, and order returns between hubs. Also, the median income in a region affects the weights: the weights are higher for regions with higher income levels and economic activity.

In the second stage of generating the demand instances, the pairwise demand rates generated in the first stage are considered to be the mean demand rates of a Poisson distribution that generates daily demand rates. We generate 10 realizations of demand rates by drawing 10 sets of random variables from the aforementioned Poisson distribution.

Instances from (56)

In the work of (56), the authors conduct computational experiments using real-world data from a major Chinese courier company. The courier's physical network includes 49 local hubs and 3 gateway hubs. The average arrival rates are determined from historical data that encompasses the origin, destination, pickup request time, and service commitment from July 2017.

Note that this is the only existing paper in the literature where shipment arrivals are assumed to happen continuously based on given rates. In (56), the authors focus on a two-layer network. The first layer consists of a network of riders, and the second layer consists of a fleet of shuttles that have repeatable routes. The aim of this study is to find the most efficient shuttle routes for operations during the day, considering the service commitment and vehicle capacity constraints.

3.5.2 Performance Metrics

The performance metrics used in this study are:

- **Service levels:** Calculated as the percentage of the packages that are delivered before their service commitments;
- **Earliness per early package:** Calculated as the average difference between the delivery time and service commitment for on-time delivered packages;
- **Lateness per late package:** Calculated as the average difference between the delivery time and service commitment for late delivered packages.

3.5.3 Numerical Analyses on the Chicago Case

Design of Experiments

In the process of clustering the 48 nodes in the Chicago area, we use the clustering framework explained in the solution approach section under Hub Clustering. We generate $k = 16$ overlapping clusters using two different maximum membership parameter values $m = 5, 10$, where the membership parameter indicates the maximum number of clusters a hub can be a member of (See A.1). We run experiments with both of those clustered networks.

We also analyze the sensitivity of the solutions vis-a-vis the vehicle capacity, q, and service commitment, S, parameters. Thus, our instances are run with vehicle capacities of 50, 100, 150, 200, and 250 shipments and service commitments of 8, 9, 10, 11, and 12 hours (although in the tables we only report results for $S \in \{8, 10, 12\}$). Also, κ, the maximum number of cycles per combination is tested with values $\kappa = 1, 2$ and 3. Therefore, the tests are characterized by parameters m, k, q, S.

The tactical plans are established based on mean demand rates (Section 3.5.3). The paths and cycles determined through the tactical planning stage are examined from an operational planning perspective (Section 3.5.4). Based on the actual demand volume of each commodity observed at the operational level (the 10 realizations discussed in Section

3.5.3) and the developed tactical plan, the load planning is executed, late time intervals are tracked, and all possible alternatives with extra dispatches are identified. Then, to achieve 100% service level, one alternative for each late time interval is selected through the use of either the greedy approach, the IP-based alternative selection model, or a hybrid approach such that the total length of extra dispatches needed is minimized. Next, we solve the corresponding VRPTW using a tabu search algorithm to find the minimum number of additional vehicles needed to cover all extra dispatches. In the operational planning of this case study, we generate 10 demand distribution realizations with the existing demand rate averages, then solve for each instance and report the average values for sensitivity analysis.

In this computational study, we run several tests with different numbers of hub clusters, k, and maximum membership limits, m. As k decreases, the hub clusters contain larger numbers of hubs, and therefore, more direct arcs among hubs, increasing opportunities for achieving higher efficiency in the network. However, as a result of the larger problem size, the required computational power to solve the problem would also increase. In an environment where the potential number of cycles per cluster is limited, the real benefit of having a lower k might not be as evident. That is why in the remainder of the paper we only focus on the case of $k = 16$. All of the models are coded in Python language and solved by Gurobi software. Models 3.1 and 3.4 are given a maximum of 12,000 seconds while Model 3 is given 12,000 seconds in the approach with no warmstart, and it is given 1200 seconds in the approach the model is warmstarted with the greedy approach.

Tactical Planning Analyses

In this section, the results of the tactical planning part of the case study are presented.

Path and Cycle Selection Model Results The results of Model (3.1) are summarized in Table 3.1. In this table, for each combination of m, k, q, and S, two values are reported. The first values are the total number of vehicles needed in the network, Y and the second values

Table 3.1: Results of Model 3.1: Fleet size and optimality gap
(Y: Fleet size, q: Vehicle capacity, S: Service commitment, m: Membership limit)

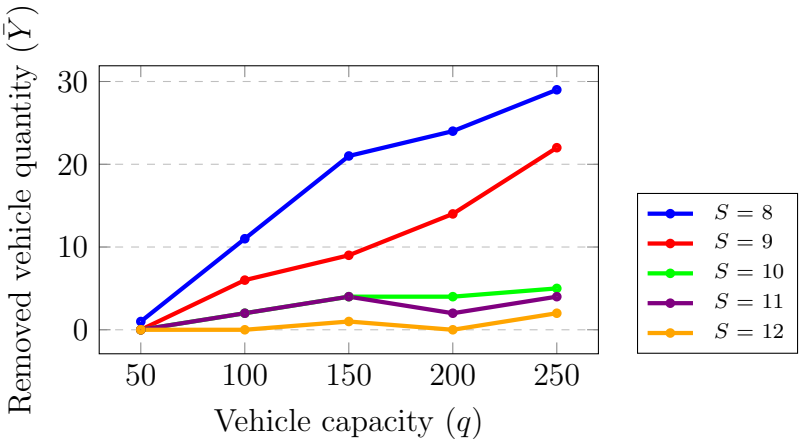
m	q	50		100		150		200		250	
	S	Y	Gap	Y	Gap	Y	Gap	Y	Gap	Y	Gap
	8	560	2.9%	344	3.8%	284	3.9%	267	4.1%	258	3.9%
5	10	534	2.1%	307	2.9%	237	3.8%	204	3.8%	199	3.0%
	12	522	1.9%	293	3.4%	220	2.7%	189	3.2%	173	2.3%
	8	557	2.3%	344	4.4%	284	3.9%	267	4.1%	259	4.2%
10	10	535	2.2%	306	2.6%	239	4.9%	204	3.8%	199	3.0%
	12	522	1.9%	293	3.4%	221	3.2%	189	3.2%	173	2.3%

are the optimality gaps of the Model 3.1 given the time limit of 12,000 seconds. According to the table, in separate instances, Y_c variable gets the values between 560 vehicles, which is obtained in instances with the tightest capacity and service commitment, and 173 vehicles, which is obtained from the instance with the loosest capacity and service commitment parameters. The optimality gap percentages take the values between 1.9% and 4.9%.

From this table, we make the following observations: (a) The tactical plan fleet size Y is inversely proportional to vehicle capacity q and service commitment S parameters; (b) Looking at the capacity in particular, the required fleet size dramatically decreases between $q = 50$ and $q = 100$, then the decrease continues with a smaller rate as the capacity increases; (c) Comparing the service commitment parameter values, their effect on the result is respectively higher as the vehicle capacity constraint loosens up. In fact, in situations where the constraint of the vehicle capacity is a loose constraint, it is more likely that the timing constraint becomes active and sets the required vehicle dispatch frequencies. In such a circumstance, any increase in the available time, S, would have a more significant impact. Also, as it can be seen in Table 3.1, the results of Model (3.1) with $m = 10$ show that although almost always the total number of required vehicles is smaller compared to when $m = 5$, such improvements are modest. Therefore, for the remainder of our analysis, we solely focus on cases with $m = 5$.

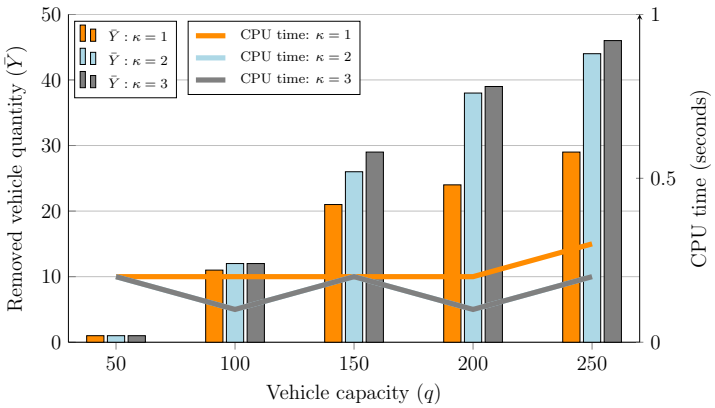
Time Redistribution The detailed results of the Time Redistribution Model (3.4) are presented in Table A.1 in the Appendix. The values for each instance represent the number of vehicles removed and the required CPU time. In this table, we break down the reported results as a function of κ, the maximum number of cycles per combination. In Figure 3.8, the results of the time redistribution Model (3.4) for $\kappa = 1$ are shown. It can be seen that the number of cycles removed from the network is the highest when the service commitment is at its lowest level. This observation shows that when service commitments are more aggressive, the value of an efficient time distribution along commodity paths is much higher. When service commitments are not binding, arcs along commodity paths already receive ample time even when S is proportionally distributed among the arcs. Specifically, when S is large, vehicle frequencies along the arcs of commodity paths are likely derived from the vehicle capacity constraint and a finer distribution of S among the arcs would have little impact. The interaction between service commitment and vehicle capacity constraints is better illustrated when we trace the curve associated with a given S with respect to different vehicle capacities: increasing the vehicle capacity results in a larger number of removed vehicles. This can be intuitively justifiable since the value of time in the Model 3.1 is highest with ample capacity and low service commitment. Although the graph only shows the instances of $\kappa = 1$, the same pattern is observed for other values of these parameters.

Figure 3.8: Number of vehicles removed in Model 3.4
(S: Service commitment)



In Figure 3.9, the results of Model (3.4) are compared for different values of κ and q, when $S = 8$. It can be observed that with a higher κ value, the number of cycles removed is always higher. This can be explained since a higher κ already includes all cycle combinations that a lower κ has. However, although this parameter does not have a significant effect on the optimization run time of Model (3.4), creating a higher number of combinations takes dramatically longer time and memory. More specifically, the higher CPU time is not because of the time the IP solver requires, but instead, the time required for such combinations to be identified and constructed and the IP model to be built. All the remaining tests are run using $\kappa = 2$.

Figure 3.9: Impact of Maximum number of cycles per combination ($\boldsymbol{\kappa}$) values in terms total run time of Model 3.4 vs the number of vehicles removed ($\bar{Y}$)



Evaluating the Quality of the Tactical Plans

The performance of the tactical plans can be evaluated by considering a case in which those plans are implemented without any extra dispatches. In that case, the tactical plans that are made based on the aggregate demand rates are tested for 10 different realizations of the demand rates by implementing load planning. The performance of the tactical plans is represented by the average of service levels, earliness per early package, and lateness per late package metrics of the tests. In Table 3.2, the service levels associated with different combinations of parameters q and S for $\kappa = 2,\ m = 5$ are reported, where each reported

value corresponds to the average service level over the 10 realizations of the demands. These results show that the average service levels achieved from the tactical plan when applied to different realizations of demand distributions are already high, advocating the robustness of the designed methods. Examining the service levels as shown in Table 3.2, we observe that with an aggressive service commitment (e.g., $S = 8$), the service levels increase with vehicle capacity. In the case of a moderate service commitment ($S = 10$), the lowest service levels are observed with moderate vehicle capacity values ($q \in \{150, 200\}$), and finally in the longest service commitment case ($S = 12$), the service levels decrease with increasing vehicle capacity. When both the vehicle capacity and the service commitment constraints are equally tight or loose (e.g., ($S = 8$, $q = 50$) or ($S = 10$, $q = 150$) or ($S = 12$, $q = 250$)), the two constraints compete for setting the vehicle dispatch frequencies. Since both constraints are almost active at the base demand rates (demand rates used in the tactical planning phase), small fluctuations in demands from the base rates will likely result in violation of both constraints and hence the creation of late time intervals and lower service levels. However, when one of the two constraints is binding while the other one is loose, the loose constraint will absorb small fluctuations in demand, even when the tight one becomes active.

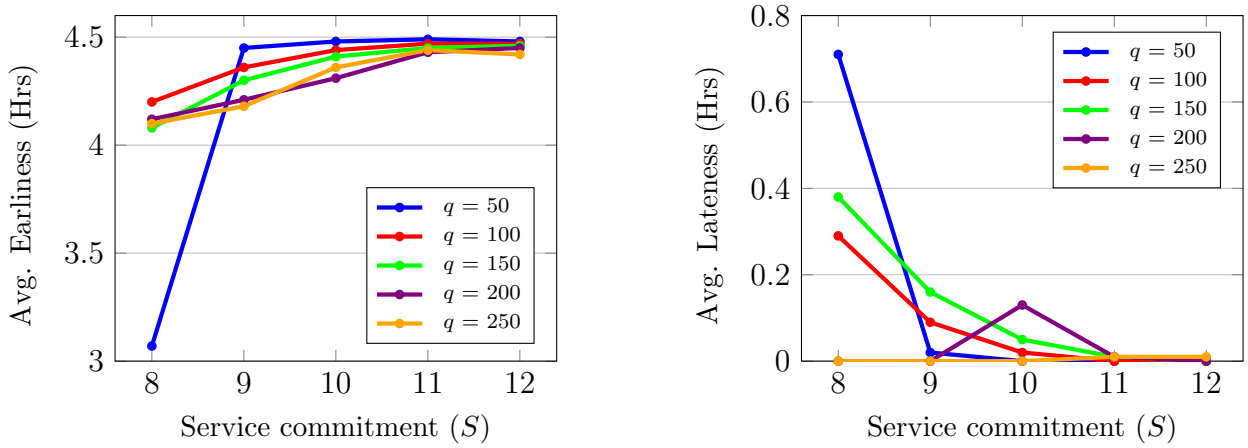
Table 3.2: Average service levels when the tactical plan applied to 10 different realizations of demand
(q: Vehicle capacity, S: Service commitment)

q/S	**50**	**100**	**150**	**200**	**250**
8	87.2%	95.1%	93.6%	100.0%	100.0%
10	100.0%	99.9%	99.3%	99.2%	100.0%
12	100.0%	100.0%	99.9%	99.9%	99.7%

Average earliness per early shipment and average lateness per late shipment are two other metrics that we use to evaluate the tactical plan. The results and detailed analysis of those metrics are given in A.5.1. While the reported service levels indicate the absolute percentage of on-time deliveries, these two new metrics add more depth to the analysis. The results show that apart from $S = 8$ for which the average earliness is slightly above 3 hours, for all other values of S considered in this study, the average earliness is between 4 to 4.5 hours.

The amount of earliness increases with S, however, it reaches a plateau as typically the fleet size shrinks when S increases as shown in Figure 3.10a. Following the same trend, the average lateness of late packages decreases with S, with that average being at its maximum for $S = 8$ at slightly above 0.7 hours. Therefore, an increase of the service commitment by one-hour results in a sharp decline in the average lateness as in Figure 3.10b.

Figure 3.10: Average earliness per early shipment and average lateness per late shipment in hours w.r.t. Service commitment ($\boldsymbol{S}$), (q: Vehicle capacity)



(a) Average earliness per early package

(b) Average lateness per late package

3.5.4 Operational Planning Analyses

In the operational planning, the dispatch schedules created in the tactical planning are customized to the needs of the 10 randomly drawn demand realizations, and the average values of the results are reported.

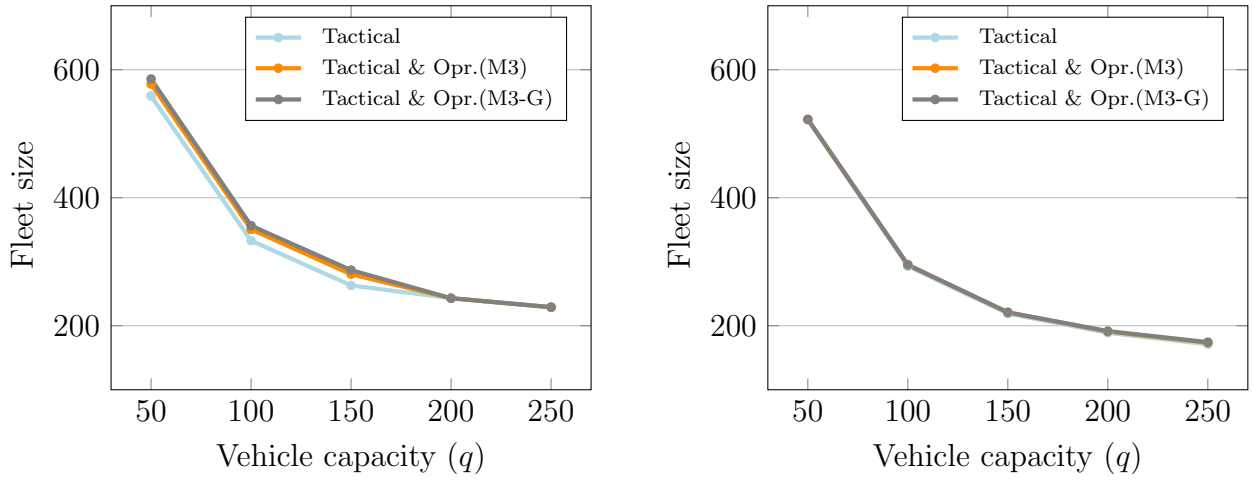
The alternative selection process for identifying the minimal set of extra dispatches is performed by using the greedy approach (G), the IP-based alternative selection model (M3), or the hybrid approach (M3-G) in which the IP model is warmstarted with the solution of the greedy. The greedy approach is fast in terms of CPU time with an average of 5 seconds in each instance. The M3 approach is capable of offering significant improvement compared to the greedy approach, however, in the majority of the instances, the walltime of 12,000 seconds is reached with an optimality gap in the order of 20%. The hybrid approach, M3-G,

on the other hand, runs only for 1200 seconds and produces solutions that are significantly better than the greedy approach and comparable to those of M3, in a fraction of the time required by M3.

The minimal set of extra dispatches obtained from the three approaches G, M3, and M3-G with or without warmstart is covered by a fleet of vehicles, where its size and vehicle routes are obtained by modeling and solving a variant of VRPTW using tabu search. The size of the fleet covering the extra dispatches heavily depends on the number of extra dispatches. Therefore, the extra dispatches obtained from G are covered by more vehicles than the extra dispatches obtained from M3 and M3-G. The detailed results are reported in A.5.3.

Finally, the total necessary fleet sizes can be calculated as the sum of vehicles operating the cycles determined in the tactical planning stage and the additional vehicles to cover extra dispatches determined in the operational planning stage. The total fleet sizes for all instances are reported in Table A.2. Similar to Table A.5, the three values for each instance correspond to the three alternative selection approaches G, M3, and M3-G. While the fleet sizes using the M3 approach have values between 173.5 and 577.7, these numbers are between 182.3 and 642.5 for the greedy approach, representing an increase of 5.1% to 11.2% in the fleet sizes.

Figure 3.11: Comparison of fleet sizes for two extreme cases of Service commitment (S)



(a) Service commitment = 8

(b) Service commitment = 12

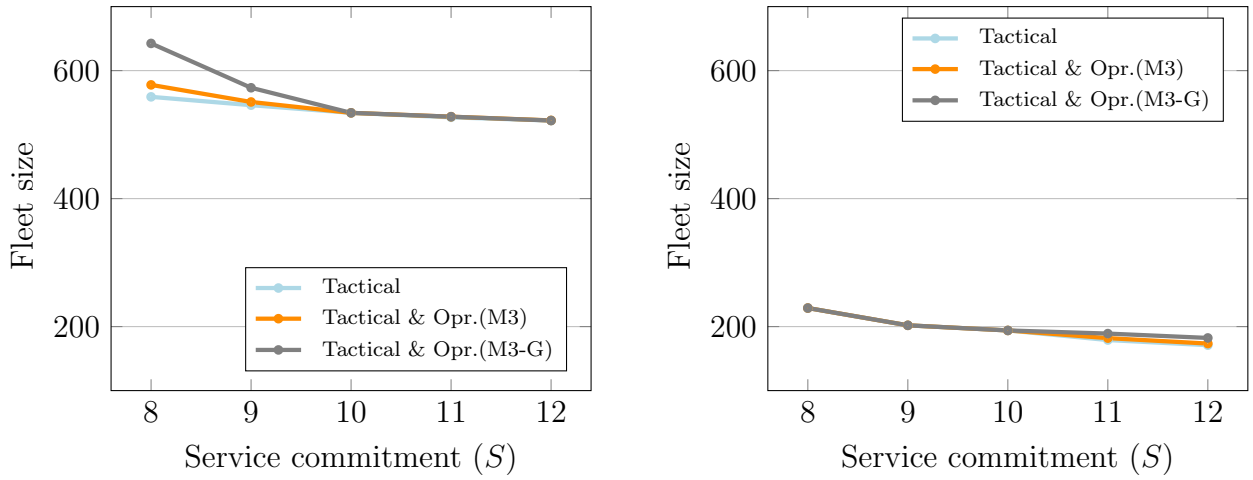
Figure 4.7 provides a comparison of the fleet size necessary for the tactical plan and the total fleet sizes after the addition of vehicles to cover the extra dispatches for the two extreme cases of S, namely $S = \{8, 12\}$. Also, the comparisons with respect to the service commitments are shown for $q = \{50, 250\}$ in Figure 3.12. When service commitments are more aggressive, for example, $S = 8$, as can be seen in Figure 3.11a, the differences between the fleet size necessary for the tactical plan and the total fleet sizes are significant. This finding is in line with the earlier observations about the service level of the tactical plan if used with no customization to demand realizations. The extra fleet size in the operational planning becomes significantly smaller when $S = 12$. This is mainly due to the high service level achieved by the tactical plan, even if no customization to the demand realizations is performed. The same relationship can be observed in Figure 3.12 as well. The instances that have visible differences between fleet sizes of only tactical plans and both tactical and operational plans are those for which the tactical plan does not guarantee a 100% service level at the operational level.

The pattern of total fleet size with respect to the vehicle capacity is similar to that of the results of Model 3.1 as it decreases with the increasing capacity. According to Figure 4.7, the effect of capacity on the total fleet size is strong, while, as it can be seen in Figure 3.12, the service commitments have a softer effect on the total fleet size.

3.5.5 Analysis of the Effect of the Frequency Parameter α

In this section, we study the effect of the frequency parameter α, the parameter that sets the vehicle dispatch frequency along an arc as a function of the maximum waiting time at the start node of the arc. We assess the effect of $\alpha \in \{1, 2, 3\}$ on the total number of vehicles (tactical and operational) when $\kappa = 2$ and M3-G approach is used. The percentage changes in the fleet sizes are presented in Table 3.3. Generally, increasing α decreases the dispatch frequencies along the arcs, which results in increasing the average wait times at the hubs. Consequently, at the tactical level, we require a smaller fleet size that guarantees a potentially lower service level. The service level can be improved by adding extra vehicles

Figure 3.12: Comparison of fleet sizes for two extreme cases of Vehicle capacity (q)



(a) Vehicle capacity = 50

(b) Vehicle capacity = 250

Table 3.3: Decrease in fleet size (Tactical & Operational) for $\alpha = 1$ vs. $\alpha = 2$ and $\alpha = 1$ vs. $\alpha = 3$.
(S: Service commitment, q: Vehicle capacity)

q	50	100	150	200	250
$\mathbf{S}/\alpha$	$1 \to 2 \mid 1 \to 3$	$1 \to 2 \mid 1 \to 3$	$1 \to 2 \mid 1 \to 3$	$1 \to 2 \mid 1 \to 3$	$1 \to 2 \mid 1 \to 3$
8	8.1% | 4.6%	10.6% | 7.9%	14.6% | 11.1%	11.8% | 5.2%	15.8% | 5.2%
10	7.2% | 3.1%	7.4% | 3.8%	7.6% | 7.3%	9.3% | 3.2%	9.5% | 2.9%
12	7.6% | 9.1%	11.5% | 11.5%	9.9% | 11.6%	9.8% | 11.3%	7.0% | 7.1%

at the operational level. Specifically, increasing α from 1 to 2 decreases the total number of vehicles for the instance $S = 8, q = 250$ by 15.8% (40.9 vehicles), and for the instance $S = 8, q = 50$ by 8.1% (47.3 vehicles). The minimum decrease is observed for the instance $S = 12, q = 250$, with 7.0% (12.3 vehicles). On the other hand, increasing α from 2 to 3 increases the total number of vehicles in most of the instances, while resulting in insignificant decreases (less than 3%) in the remaining instances. These results show that a less conservative tactical plan ($\alpha = 2$ instead of $\alpha = 1$) and taking care of occasional lateness at the operational level could lower the required fleet size. However, it is often the case that the capacities committed to ahead of time are cheaper than last-minute additions, therefore, given such costs, the decision-maker will be able to identify the most cost-efficient solutions.

3.5.6 Comparison with the state-of-the-art in the literature

In this section, with the purpose of validating our approach, we compare the performance of our solution framework with (56). We apply our proposed tactical and operational planning procedures to the instances solved by (56) and compare our results. We modified our model to accommodate multiple vehicle types so that we can solve their instances. Our hub clustering procedure is implemented to reduce the network size with $k = 16$ and $m = 10$.

We follow the exact same rules to generate travel times and demand rates. We also consider a service commitment $S = 4$ and set the frequency parameter α in Constraints (3.1e) to 2 to be consistent with the implementation in (56). $\kappa = 2$ was used for the time redistribution model. The comparison is made for only Regime 1 instance of (56), and all of the other parameters are kept the same for both studies.

The results of the comparison are shown in Table 3.4. Our tactical planning approach proposed achieved an average on-time delivery (service level) of 99.89% and a minimum on-time service of 68.69% among all commodities using 251 vehicles. In comparison, the proposed model in (56) achieved an average on-time metric of 99.96% and a minimum on-time metric of 57.53% using 307 vehicles in their Base case. Even after their cycle removal phase (CRP) which is meant to reduce the fleet size, their design still requires approximately 10% more vehicles compared to our tactical plan with a larger portion of the fleet consisting of larger vehicles, while the difference in the average on-time metric is negligible and the difference on the minimum on-time metric is 16%. This comparison with (56) shows that our approach on average guarantees almost the same level of on-time metric, and performs significantly better on the minimum on-time metric.

Our proposed combined tactical and operational planning procedures achieved a 100% score for both average on-time and minimum on-time metrics using 258 vehicles. Meanwhile, the model proposed in (56) achieved an average on-time metric of 99.97% and a minimum on-time metric of 85.11% using 278 vehicles after their cycle-adding phase (CAP) and CRP, with a larger portion of the fleet consisting of higher capacity vehicles. In summary, our

approach can generate commodity paths and vehicle cycles that result in a higher level of service using a smaller fleet size.

Table 3.4: Comparing the performance of our approach to that of (56)

		This paper		**(56)**		
		Tactical plan	**Operational plan**	**Base**	**CAP**	**CRP**
# of vehicles	**Vehicle type: 1T**	143	150	130	173	101
	Vehicle type: 3.5T	108	108	177	177	177
	Total	251	258	307	350	278
Metrics	**Average on-time**	99.89%	100.00%	99.96%	99.99%	99.97%
	Minimum on-time	68.69%	100.00%	57.53%	86.13%	85.11%

3.5.7 Comparison to the Hub-and-Spoke Network

In this section, we evaluate the performance of our proposed horizontal network design and solution approach by comparing it to the classical hub-and-spoke network system. The comparison is conducted for two instance classes: (1) instances from (56) and (2) the Chicago case study instances.

The hub-and-spoke network operates with either one central distribution center (DC) or multiple DCs. In the single DC setup, all commodities are first transferred to the DC where they are sorted before being sent to their final destinations. The transfers are accomplished through cycles between the hubs and the DC, with the frequency of a cycle between a DC and a given hub being determined by the most restrictive of (1) the minimum time required to deliver all commodities from a hub before their deadlines, (2) the capacity needed to transport the flow leaving the hub toward the DC, and (3) the capacity needed to transport the flow leaving the DC entering the hub. In a network with multiple DCs, each hub is assigned to the nearest DC. There are cycles between each hub and its assigned DC, as well as cycles between each pair of DCs. Commodities are first transferred to their origin hub's assigned DC, then to the assigned DC of their destination hub, and finally delivered to their final destination hub.

The comparison of the fleet size between the proposed operational and tactical planning of the horizontal network ($\alpha = 1$ and M3-G) and the hub-and-spoke network was not possible

for the instances of (56) that have 3 DCs since the hub-and-spoke network turned out to be infeasible even with a very low handling time of 10 minutes at the hubs. Specifically, the considered service commitment does not allow some commodities to go through the intermediate DCs before reaching their destinations.

We were able to conduct the comparison in the Chicago case for both 1DC and 2DC with 1 hour sorting times at the DCs. The locations of DCs are selected to be at the two airports of Chicago, which is a common area for the courier companies to establish their sorting centers. In the 1DC case, we chose the closest airport and in the 2DC case, we selected the 2 closest airports to the downtown area.

Table 3.5 contrasts the fleet size necessary for full-on-time delivery using our proposed horizontal approach (labeled as "No DC") and the traditional hub-and-spoke network with either 1 or 2 DCs for varying vehicle capacities and service commitments. The optimal solution for each parameter combination is highlighted in bold. The results indicate that our proposed approach generally leads to (often significantly) smaller fleet sizes than both the 1DC and 2 DC cases. In some rare cases, specifically when the vehicle capacity is large and the service commitment is tighter, the hub-and-spoke network with 1DC marginally outperformed the horizontal network design.

Table 3.5: A comparison of fleet sizes between the horizontal network design and the hub-and-spoke network design with 1DC and 2DCs. (q: Vehicle capacity, S: Service commitment)

q	50	100	150	200	250
S	NO DC | 1DC | 2DC	NO DC | 1DC | 2DC	NO DC | 1DC | 2DC	NO DC | 1DC | 2DC	NO DC | 1DC | 2DC
8	**586** | 832 | 1153	**356** | 428 | 587	**282** | 293 | 411	229 | **228** | 336	214 | **188** | 308
10	**534** | 832 | 1153	**308** | 428 | 587	**242** | 293 | 405	**210** | 228 | 308	191 | **188** | 253
12	**523** | 832 | 1153	**295** | 428 | 587	**220** | 293 | 405	**191** | 228 | 308	**174** | 188 | 252

3.5.8 Time-Feasibility Robustness

In the express delivery setting, possible increases in travel times may lead to potential late deliveries. Although travel time uncertainty is not accounted for explicitly in our algorithm, we conduct a robustness analysis to assess the resiliency of the ultimate plans. Recall that

in our implementation, the handling times at the hubs are considered to be part of the travel times. Therefore, any variations in travel times may have roots in actual travel times or handling times at the hubs.

Suppose that at the operational level, $\gamma\%$ of vehicle dispatches experience some delay. For a vehicle dispatch d_a along arc a, one may observe a delay $\xi_{d_a}\%$ where $\xi_{d_a} \sim U(0, \xi^{max})$. That is, the actual travel time of vehicle dispatch d_a becomes $t_{d_a} = t_a(1 + \xi_{d_a})$. Whenever a vehicle dispatch is selected and its travel time is increased, the subsequent dispatches of that vehicle (either in a cycle or route) are pushed forward to account for the domino effect of the delay. With the updated set of vehicle dispatches, the load planning procedure is repeated for the same set of commodities without adding any new vehicles. Given the described setting, we evaluate the on-time performance of our operational plans when $\gamma = \xi^{max} = 25\%$. Specifically, we randomly select 25% of the vehicle dispatches obtained during the operational planning and increase their travel times by a percentage uniformly selected over the interval $(0, 25\%)$. The resulting service levels are reported in Table 3.6. Each value in the table represents the average service level of 10 instances, for different values of S, q, and α.

As can be seen from Table 3.6, following random increases in travel times, the operational plans demonstrate service levels ranging from 93.6% to 99.6% (for $\alpha = 1$), indicating a high level of robustness. Specifically, the lowest service level observed is 92.1% when both capacity and service commitment constraints are tight, while the highest robustness is observed in the presence of longer service commitments and smaller vehicle capacities. The latter can be explained by higher dispatch frequencies along the arcs, which reduce the effect of the delays. In summary, our robustness analysis demonstrates the effectiveness of our operational plans in dealing with travel time uncertainties.

3.6 Discussion and Managerial Insights

We considered the problem of an intra-city courier service provider offering high-velocity delivery services in densely populated metro areas. The provider handles high package

Table 3.6: Average service levels after random travel time increases (q: Vehicle capacity, S: Service commitment, α: Frequency Parameter)

q	50	100	150	200	250
S	$\alpha = 1 \vert \alpha = 2$	$\alpha = 1 \vert \alpha = 2$	$\alpha = 1 \vert \alpha = 2$	$\alpha = 1 \vert \alpha = 2$	$\alpha = 1 \vert \alpha = 2$
8	%93.6 | %92.1	%96.5 | %94.8	%96.8 | %94.5	%97.4 | %94.2	%97.7 | %94.1
10	%96.4 | %95.6	%97.3 | %94.9	%96.8 | %95.3	%96.9 | %94.7	%97.7 | %94.4
12	%99.6 | %99.5	%99.2 | %98.4	%96.7 | %95.3	%97.2 | %95.3	%97.3 | %95.6

volumes while committing to a tight service guarantee. Based on the idea of utilizing the pick-up and drop-off stores of the courier company as sorting facilities, a network can be structured with such stores as hubs. Our proposed methodology relies on tactical-level planning followed by operational planning. The tactical planning takes the form of a multi-commodity service network design. The tactical plan is based on aggregate demand rates over a relatively long period of time such as a week, a month, or a season. We developed a MIP formulation that identifies one path for each commodity and allocates a set of vehicles to cyclic routes to operate continuously to execute the traversal of the shipments along arcs according to the determined commodity paths. Then, a second model is introduced to decrease the number of cycles needed by rearranging the time allocations along different arcs of a commodity path while still meeting the service guarantee. Then, the start hub and start time of each vehicle are determined in such a way that the effect of inter-dispatch time variability is minimized. In the operational planning phase, the time horizon is shortened, e.g. a day and the predetermined tactical plan is supplemented with extra dispatches on specific arcs and during specific time windows to alleviate potential deviations of the observed demand from the aggregate demand patterns. This two-phase planning framework is applied to instances adopting the topology of a major US city in an extensive computational study. Notice that although we presented the tactical and operational planning stages in an integral way, the insights gained from each part can be applicable to relevant planning stages of other logistics problems, even in the absence of the other part. For example, the operational

planning problem may assume a tactical problem to be exogenous. The key findings of the computational experiments show that:

- The results show that the service levels achieved from the tactical plan are already high and further improved through the operational level planning.
- When both service commitment and vehicle capacity constraints are binding, the service levels are relatively lower due to limited flexibility to accommodate demand fluctuations.
- Our approach demonstrates a significant reduction in fleet size compared to the current state-of-the-art, with a higher proportion of smaller vehicles.
- Our proposed horizontal network structure significantly reduces the fleet size compared to the classical hub-and-spoke network, in the majority of 1DC cases, and in all 2DC cases. Additionally, the hub-and-spoke network was infeasible in a real-world dataset (56), while the horizontal approach produced high-quality results.
- Our solutions approach generates solutions that are robust in the face of moderate travel time variations, even when uncertainty is not accounted for explicitly.

Our approach is based on some simplifying assumptions such as constant commodity demand rates over a day and constant handling time at the hubs. The future work will include relaxing these assumptions by considering time-dependent demand and load-dependent handling time at the hubs. Additionally, other aspects of the problem such as allowing overtime and quantifying the cost of the operations as well as the environmental effects of the operations design, considering a hybrid setting of the proposed horizontal network and the hub-and-spoke network can be some future directions. Developing solution methodologies based on metaheuristic approaches may be required for both tactical and operational planning in high pace settings.

CHAPTER 4

A BRANCH-AND-BENDERS CUTS ALGORITHM FOR A STOCHASTIC SERVICE NETWORK DESIGN WITH CROWDSOURCED CAPACITY

4.1 Introduction

There are 2,573 urban areas in the United States, embodying a population of 249M, representing 83% of the entire country's population (US 24). This is the result of a phenomenon often referred to as urbanization. Such a concentration of population in urban areas brought about, among others, a significant increase in demand for express delivery services. A major part of the demand is due to the growth of online shopping. E-commerce represented 13.2% of all retail sales in 2021 in the US. With the catalytic effect of COVID-19, courier services in all forms of B2B, B2C, and C2C garnered great attention. Courier companies, as the pillar of the last-mile delivery of e-commerce, try to keep their service levels and consumer satisfaction high while competing with each other in terms of offering higher standards. Currently, major courier companies such as UPS, FedEx, DHL, SF Express, and JD.com, offer express and same-day delivery services in several major cities. The cost and speed of delivery are the main differentiating factors among courier services. With such slim lead times, companies have limited opportunities for consolidation to improve vehicle utilization, intending to reduce per-shipment costs. Moreover, short service guarantees amplify the effect of uncertainties in demand volumes over time in terms of system performance even further. Specifically, uncertainties stemming from the demand side, such as the arrival time and volume of packages, as well as their destination have received substantial attention in

the literature (97). Also, the uncertainties regarding travel in large cities such as traffic conditions, parking limitations, etc. have been studied (94).

The advances in mobile networking technologies, smartphones, and online payment systems have made it possible for organizations to tap into the temporary and task-based workforce as opposed to permanent workers on long-term contracts (134; 130; 145; 91; 106). The "crowd" has become a new source for organizations to capitalize on flexible labor exchange (146; 59). Crowdshipping consists of settings that allow a group of ad-hoc drivers to take over the entire or part of the last-mile delivery of online orders (8; 11; 57; 109; 147). Over the past few years, crowdshipping has gained growing attention, mainly due to its economic, environmental, and social benefits (129). AmazonFlex, Uber Freight, Doordash, Shipt, Instacart, and Postmates are only a few of the successful applications of employing the crowd in logistics applications. However, those benefits also come with challenges. The inherent supply (transportation capacity) uncertainty, i.e., crowd availability, adds to the complexity of crowd-based operations planning. In recent years, studies addressing those uncertainties such as supply availability, compensation, and demand-supply matching have appeared (150; 53; 57). To provide a service guarantee that is both reliable and robust, the industry has witnessed the emergence of hybrid delivery fleets, as demonstrated by leading companies such as Amazon, Walmart, Veho, and Bringg. These fleets strike a balance between the low-capital and uncertain nature of crowdshipping and the high-asset and reliable nature of an owned fleet. Thus, there are some recent studies in the literature (57; 17).

In this study, we consider a service network design as the underlying mechanism of intra-city express courier service. The network consists of a set of hubs representing the pick-up/drop-off stores/stations throughout the city, and each shipment is associated with one hub as its origin and one as its destination. The courier company is committed to transferring the shipments from their origins to their destinations within a service guarantee. The considered horizontal network differs from the traditional hierarchical networks which

require all shipments to be first transferred to distribution centers to be sorted before being redistributed to their destinations. In a horizontal network design, we assume the courier hubs are equipped with a moderate sorting capability, removing the need for a sorting process at a distribution center. This network setting has been investigated in various studies on service network design, such as those by (56; 141; 93; 164) and (80). Furthermore, (80) highlight that their partner company SF Express in China utilizes this type of network structure. The network potentially involves a link between each hub pair. From its origin to its destination, a shipment may go through multiple hubs and traverse their linking arcs. The considered intra-city service network has a strong resemblance to the classical less-than-truckload (LTL) long-haul networks, and therefore, the insights gained from our study may be applicable to long-haul LTL settings. While we adopt such a network structure, the scope of our study does not extend to examining the limitations of the hubs within these networks.

The design of operations of such a system consists of concurrent decision-making regarding freight routing in the network and resource allocation in accordance with such routes. We consider the employment of a hybrid fleet, consisting of three possible channels through which the courier company can acquire transportation capacity. The first channel consists of the drivers that the company commits to using based on a given set of routes well ahead of the actual operation date. Examples of these drivers are corporate drivers or contractual drivers hired through a forward market. The advantage of using such a channel is its relatively low fee. The second channel consists of a spot market, where the company can acquire the required transportation capacity on specific links of the network at specific times with short notice. This option, while being flexible is the most expensive option due to its on-demand nature. The third channel is by employing crowdshippers if their availability (the time and the routes they are willing to operate) can be exploited by the system. Crowdshippers are generally cheaper than the drivers hired through the spot market, however, their availability is uncertain and out of the control of the courier company.

The main purpose of this paper is the design of operations of such a network both at the tactical and operational levels. In the proposed setting, at the tactical level and given some probabilistic information about future demand and crowdshipper availability across the network, a set of vehicle routes are scheduled for corporate (or forward contractual) drivers. At the operational level, once more accurate information about the demand and crowdsourced capacity is revealed, additional drivers in the form of crowdshippers (if available) or on-demand third-party drivers from the spot market can be added at different times and locations to the network to compensate for any lack of transport capacity.

4.1.1 Related Literature

The service network design problem (SNDP) is a transportation planning problem that aims to optimize resource allocation while meeting service level targets, involving direct service decisions, routing, terminal operations, scheduling, resource management, and empty repositioning, and has been well-reviewed in (35; 163; 79; 39).

SNDP Classification. (39) differentiate between two classes of service network design problems known as static SNDP and time-dependent (scheduled) SNDP (SSNDP), based on how the temporal dimension of the problem is approached. SSNDP explicitly represents the demand and activities in time while the static SNDP models the frequency of executed services. Our problem falls into the category of SSNDP; the most recent studies being (49; 42; 67; 82).

Additionally, taking a different angle, the SNDP can be classified into two categories: deterministic, where all parameters are assumed to be known, and stochastic, which takes into account the uncertainty in various parameters such as demand, capacity, travel times, costs, and breakdowns. The uncertainties may lead to less efficient plans if the parameters are reduced to estimated average values (107; 12), and incorporating uncertainty into the problem can enhance the plans' robustness, reliability, and efficiency (88).

Stochastic SNDP. Most of the studies on stochastic SSND focus on demand uncertainty. (108) study a setting with fixed capacity and resource-management constraints combined

with uncertain demand. For small-sized instances, they show that cost reductions can be achieved when stochastic demand as well as the correlation of demand is explicitly considered. (107) address real life-size instances of the problem, proposing a Variable Neighborhood Search-based metaheuristics to solve large instances in a reasonable time. (158) consider both fixed and variable capacity, taking the temporal dimension of the demand into account. Recently, (105) study a modified version where certain service vehicles remain fixed throughout the planning horizon, while others possess the flexibility to adjust their routes based on daily customer demands, for which the authors propose a two-stage stochastic mixed-integer linear program.

(83) propose a model that simultaneously addresses strategic decisions regarding fleet sizing and allocation, as well as tactical decisions regarding a repeatable transportation plan and schedule. The authors incorporate resource acquisition decisions into the stochastic SNDP with demand uncertainty and propose a column-generation-based matheuristics scheme to solve the problem. In another recent work, (159) develop a two-stage robust optimization method and introduced a column-and-constraint generation approach to solve the introduced robust models exactly. (88) propose a method for bundling scenarios in a progressive hedging heuristic to consider an SNDP with uncertain demand. The authors use Fuzzy C-Means and Gaussian Mixture model methods to calculate the membership score of a scenario to each bundle center.

Meanwhile, the consideration of travel time uncertainty has been gaining attention in recent years. (58) develop a two-stage model to minimize both the cost and carbon emissions of a multimodal transportation network with fixed rail and maritime schedules. The first stage selects motor-carrier services, and the second stage adjusts costs when delays to upcoming shipments are observed. Other examples are (96; 97), which focus on SSND with quality targets. The authors define quality targets for the on-time operation of services and delivery of demand loads to destinations. They propose a two-stage model and introduce a progressive hedging (PH) based metaheuristic to solve the problem.

Our paper differs from the existing studies in the literature of stochastic SND as we address both demand and capacity uncertainty simultaneously, which results in a highly complex problem. Next, we discuss supply uncertainty in the context of crowdsourcing and its relevant literature.

Crowdsourced Logistics. "Crowd" has been employed in different parts of logistics and there is a growing body of literature on crowdsourced transportation (2). However, most existing applications studied in the literature take the form of Vehicle Routing Problem (VRP) settings. One of the first studies on the use of crowd for last-mile delivery, often referred to as crowdshipping is (8), in which the authors introduce the vehicle routing problem with occasional drivers (VRPOD). Subsequent studies by (109; 53; 150; 110; 119; 123) explored various extensions, such as split deliveries, in-store customers, pickup and delivery with time windows, intermediate depots, and last-mile delivery networks, with solution approaches such as variable neighborhood search, branch-and-cut, and Benders Decomposition algorithms. It should be noted that the aforementioned works mainly focused on incorporating crowdshipping into delivery operations with the assumption of supply uncertainty, without considering demand uncertainty. (11) and (57) consider the uncertainty in both supply and demand in their crowdshipping models, where (11) focuses on dynamic routing, while (57) proposes employing in-store customers for delivery of online orders from the store inventory, in a setting where both online order placements and in-store customer arrivals are stochastic.

None of these studies, however, consider transferring packages between different vehicles (stopovers/cross-docking), but in this paper, a package may be transported with the help of multiple vehicles and crowdshippers. Our paper falls into the category of a time-dependent stochastic SNDP addressing both demand and supply uncertainties allowing the transfer of packages in their routes. In the literature, we are aware of only one other relevant study that considers a comparable problem. (93) focus on a last-mile delivery within-region model that incorporates public transportation, crowdshipping, and backup transfers utilizing

parcel lockers as hubs. Similar to ours, this paper considers stochastic demand and capacity, allowing multiple stopovers in package routes. The authors propose a path-based two-stage stochastic programming formulation and develop a branch-and-price algorithm to solve the problem. Our work differentiates itself from (93) in two main aspects. First, (93) use the existing public transportation capacity with pre-determined schedules as a means of transportation. They focus on identifying public transport transshipment points and generating a set of backup transfer routes. In our setting, at the tactical level, no existing routes are considered and the decision maker needs to determine a set of routes for the corporate drivers in the absence of full information about the future demand and potential crowdsourced capacity in the future. Second, we assume that crowdshippers are available at specific times on certain links of the network, which are unknown to the decision-maker at the time of designing tactical plans (probabilistic information is assumed to be available), and the actual information about their availability is revealed only shortly prior to employing them, whereas (93) assume that crowdshippers are available for the entire scheduling period and willing to travel to any node.

Benders Decomposition (BD). A widely employed approach in modeling stochastic programming is the utilization of the two-stage formulation. Within these formulations, the decision variables are partitioned according to the information accessible during decision-making, and the uncertain parameters are typically represented by a set of scenarios. Initially introduced by (18) and subsequently applied to stochastic models in (154) (referred to as L-shaped method).

The Branch-and-Benders-Cut strategy adopts a distinctive approach by generating valid cuts within a single search tree, obviating the need to establish a new branch-and-bound tree each time a new set of cuts is generated. A comprehensive review of these studies, as well as Benders Decomposition in general, can be found in (132).

Most recently, two primary methodologies have gained considerable attention for enhancing the efficiency of Benders Decomposition. First, in the work of (45), the introduction of Partial

Benders Decomposition (PBD), which involves the incorporation of explicit information from the scenario subproblems directly into the master problem. The second approach focuses on enhancing the quality of the cuts. (20) and (133) apply valid inequalities to reinforce classical Benders cuts. (131) introduce the Benders Dual Decomposition (BDD) method, which entails reformulating the subproblems by incorporating local copies of master variables. This method utilizes Lagrangian duality to assess coupling constraints and determines the pricing of connections between the local copies and the master variables. Lagrangian techniques have also been applied to generate alternative optimality cuts, especially when integrality requirements are present in the subproblem (25; 169).

The majority of applications of BD on two-stage stochastic problems predominantly concentrate on scenarios with integer or continuous first-stage variables and continuous second-stage variables. However, if similar to our problem, some or all of the second-stage variables are integer, the duality of the subproblems can not be applied, and the classical BD cuts can not be generated. The existing literature addressing this challenge is succinctly summarized in (132). While a prevalent strategy involves solving the linear relaxation of the subproblems, (98) introduce the integer L-shaped method (ILSM), which utilizes lower-bounding functions as opposed to classical optimality cuts. Importantly, this method imposes the condition that the variables in the master problem must be binary. Subsequently, (7) propose a framework employing a cut-generating linear program that makes use of previously visited solutions, thereby enhancing the methodology.

4.1.2 Contributions

Our contributions in this paper are summarized as follows.

- We introduce the concept of service network design with the hybrid fleet, in which the fleet of corporate and/or contractual drivers is assumed to be augmented by a group of ad-hoc crowdshippers who can perform the transport of shipments between hubs. The considered setting is characterized by a high level of uncertainty due to stochastic demand as well as the inherently stochastic nature of crowdshipper availability.

- We formulate the problem as a two-stage stochastic programming with recourse, representing the decision-making at tactical and operational planning phases. This allows us to explicitly account for the stochastic demand as well as the uncertain availability of crowdshippers. In such a formulation, the corporate drivers are scheduled in the first stage based on estimations of future demand and crowdshipper availability. At the operational level, if the corporate drivers' routes and crowdshipper availability do not fully match transportation capacity needs on the network, a recourse action is taken by employing third-party drivers through a spot market.

- We develop a tailored, state-of-the-art Branch-and-Benders-Cut (BBC) approach to solve the proposed two-stage stochastic program with integer/binary variables both in the first and second stage. Our proposed approach combines techniques from the classical Benders Decomposition, integer L-shaped method, Benders Dual Decomposition, and partial Benders Decomposition within a branch-and-cut framework. Our method enforces integrality of both the first-stage and the second-stage variables. Our search is built on a single tree incorporating selective subproblems, parallelism, and ϵ-optimality techniques.

- We introduce the concept of partially adaptive stochastic service network design (PASSND): Once a solution is obtained to the stochastic program, the decision maker gets the chance to partially re-optimize the system and update the plans based on the additional information gained at specific times as the plans are being executed. Through a computational study, we determine the sweet spot to schedule a plan update assuming that the number of allowed updates is limited.

- We conduct an extensive computational study to assess the performance of the proposed approach on a large set of instances and provide insights through sensitivity analyses. Further, we quantify the added value of crowdshippers, and determine where employing them and encouraging their participation will be helpful.

4.1.3 Paper organization

The paper is organized as follows. In Section 4.2, we describe the problem setting, and in Section 4.3, we formulate a two-stage stochastic model. In Section 4.4, we discuss our proposed solution approach. In Section 4.5, we define PASSND and discuss its implementation. Next, in Section 4.6, we describe the computational experiments and analyze the results. Finally, in Section 4.7, we present some concluding remarks and discuss future research.

4.2 Problem Definition

We consider a *planning horizon* (e.g., a month) consisting of a sequence of *operational periods* (e.g., days). Suppose we can identify subsets of operational periods that exhibit similar demand and capacity patterns. Each subset of operational periods can have a tactical plan designed specifically for it, which serves as the basis for operations during those periods. For instance, one can consider a month as the planning horizon noticing that the same days of the week over that month exhibit similar demand and capacity patterns (e.g., all Mondays have more or less similar behaviors). In such a case, one tactical plan per day of the week is designed and will be the basis of operations for those days during the targeted month (planning horizon). The length of the planning horizon depends on a series of factors such as the precision of the forecast, the change of parameters over time, and the level of flexibility in the resource allocation. Each representative operational period $\mathcal{T}$ is discretized into a set of equal-length time epochs $t \in \mathcal{T}$, where all events are mapped.

We define an intra-city service network on a graph $\mathcal{G} = \big(\mathcal{H}, \mathcal{A}\big)$ where the set of nodes $\mathcal{H}$ corresponds to the physical hubs. In an urban area, such hubs can be local stores or drop-off and pick-up stations of a courier service provider. Each hub represents the potential origin or destination of a shipment. The set of directed arcs $\mathcal{A}$ represents the links between the nodes in the graph. For an arc $a = (i, j) \in \mathcal{A}$, i is called the tail node and j is the head node of the arc. The length of an arc $a \in \mathcal{A}$, denoted l_a, corresponds to the travel time between

its tail and head nodes. Sets $\delta(i)^+$ and $\delta(i)^-$ represent the set of outgoing and incoming arcs of node i in network $\mathcal{G}$, respectively.

The shipping demand is expressed as a set of commodities $\mathcal{K}$, where a commodity is a group of shipments sharing the same attributes such as origin and destination hubs, arrival time to the system, and the latest allowed delivery time at their destination hubs. Each commodity k is encoded as a tuple $\langle o_k, d_k, t_k, q_k \rangle$, with $o_k \in \mathcal{H}, d_k \in \mathcal{H}, t_k \in \mathcal{T}$, and q_k being the origin hub, destination hub, the arrival time to the system, and the volume of commodity k, respectively. We assume that the demand is stochastic, that is the set of commodities over operational periods is unknown. We consider a common service guarantee, R (a multiple of the length of a time epoch), for all commodities. That is, commodity k is due at its destination by $\sigma_k = t_k + R$. Notice that the extension of the proposed approaches to a setting where the service guarantees are customized to commodities is straightforward.

The transportation capacity to move shipments in the network of hubs is provided using a hybrid fleet consisting of three types of drivers. The first type is a fleet of drivers that the courier company commits to employing based on specific sets of routes designed at the tactical level. These could be company drivers or contractual drivers hired through the forward market. We denote such drivers as CD. CD routes take the form of partial tours. That is, from the start hub of a vehicle to its final hub, route continuity is preserved and the entire vehicle movements along that route are paid for. At the end of each day, CDs return to the central parking location G from their final hub, although the movements from and back to G are not subject to payment. Once the routes of CDs are identified at the tactical level, there are zero or limited opportunities to alter them at a later stage. The second type of driver consists of crowdshippers, denoted by CS drivers. In contrast with CDs, CSs announce their availability shortly (at most a few hours) before being employed, and therefore, the courier company cannot count on their capacity with certitude ahead of time. A CS availability consists of their willingness to transport a certain number of shipments (based on their vehicle capacity) on a specific link of the network at a specific time. CSs can

announce their availability in the format of a movement between a pair of hubs at a specific time during an operational period. A CS availability must become known soon enough for the dispatcher to be able to employ them on a task. This can be thought of as somebody willing to operate a transfer between a pair of hubs close to their home and work locations on their way to work at a specific time of the day. The third type of drivers are those hired through the spot market (e.g., Uber Freight) on an on-demand basis, denoted SMs. Similar to CSs, SM routes do not necessarily take the form of a tour and can be as simple as one movement between a pair of hubs of the network at a specific time. A movement (i.e., the transfer between a hub pair) can be assigned to a CS only if at least one CS is available along the corresponding arc of the network at that specific time. SMs, on the other hand, can be called in at any time with a negligible lead time. From a capacity perspective, while vehicle capacities across different driver categories could be different, we assume that all vehicles within a category offer the same capacity. Specifically, let u^{CD}, u^{CS}, and u^{SM} be the vehicle capacities of CDs, CSs, and SMs, respectively.

From a cost perspective, CSs are the cheapest option in terms of per-mile cost. Crowdshippers typically perform simple movements between two hubs on their way to their next destinations, which would often involve a short detour from their personal route. Consequently, moderate compensation would be sufficient to incentivize their participation. The second cheapest type of driver in terms of per-mile cost is CD, as they are either owned by the company or are hired on longer contracts and well in advance. Finally, the most expensive resources are SMs. The cost of each type of driver is correlated to the flexibility it offers, CSs are the least flexible option as the company has no control over their availability and learns about their willingness to participate with short notice. CDs are fully flexible at the tactical level, however, once the routes are fixed, no flexibility is available at the operational level. SMs are the most flexible resource as they can be called in as they are needed at the operational level. The cost on each arc corresponding to each vehicle type is represented by a fixed cost

plus a variable cost proportional to the traveled distance. The total costs associated with each arc $a \in \mathcal{A}$ are denoted c_a^{CD}, c_a^{SM}, and c_a^{CS} for CD, SM, and CS drivers respectively.

Along its route from its origin to its destination, a shipment associated with a commodity may go through multiple hubs, while the inter-hub movements are executed by CD, SM, and CS drivers. That is, for part of its route, a shipment may be loaded on a CD vehicle, and for some other part on an SM or CS vehicle.

The design of operations of such a system is challenging and the main challenge stems from the inherent uncertainty both in terms of demand (what commodities, at what quantities, and when will actually occur during a given operational period) and CS capacity (on which arcs of the network and at what times there will be CS drivers willing to serve). Given probabilistic information on demand and CS availability, our goal is to design a mechanism that allows scheduling *a priori* routes for CDs at the tactical level, keeping in mind the potential cost savings of employing CSs and the additional cost of hiring SMs at the operational level. Also, we would like to investigate how the additional and more accurate information revealed with regard to stochastic elements (demand and CS availability) as we go through an operational period can be exploited to update the plans with the goal of minimizing operational costs.

In the next section, we propose a mathematical formulation for the problem. The model relies on the fact that tactical decisions (CD routing) must be made well before the actual demand and CS availability are revealed and that such plans could be augmented at the operational stage by adding SM and CS drivers whenever needed.

4.3 A Two-stage Formulation

We formulate the defined stochastic service network design with a hybrid fleet as a two-stage stochastic problem with recourse, where the first stage concerns the planning of CD drivers at the tactical level while the second stage supplements the first-stage decisions according to the observed demand and capacity, with $\mathcal{K}$ and $\mathcal{B}$ being the support of them,

respectively. In that sense, the recourse consists of adding SM or CS drivers in the second stage. We choose to formulate uncertainty in terms of commodity and CS availability through a set of scenarios, $\mathcal{S}$. Each scenario $s \in \mathcal{S}$, with an occurrence probability p^s, represents one possible realization of demand and crowdshipper availability over a given operational period. We denote by $\mathcal{K}^s$ and $\mathcal{B}^s$ the set of commodities and crowdshippers in scenario s. Parameter $b_a^{st} \in \mathcal{B}^s$ indicates the number of available crowdshippers along arc a at time epoch t according to scenario $s \in \mathcal{S}$.

4.3.1 First-stage Formulation

Tactical decisions, i.e., fleet sizing and route scheduling of CDs for contractual and scheduling purposes, are made in the first stage. Let integer variables X_a^{t-} and X_a^t be the first-stage decision variables, indicating the number of CD vehicles dispatched along arc a, with an arrival time t at the head node of a and departure time t from the tail node of a, respectively. The first-stage formulation takes the following form.

$$\text{Min} \quad \sum_t \sum_a c_a^{CD} X_a^t + \mathcal{Q}(\boldsymbol{X}, \mathcal{K}, \mathcal{B}) \tag{4.1a}$$

$$\text{s.t.} \quad \sum_{a \in \delta(i)^-} X_a^{t-} = \sum_{a \in \delta(i)^+} X_a^t, \qquad \forall i \in \mathcal{H}, t \in \mathcal{T} \tag{4.1b}$$

$$X_a^t \in \mathbb{Z}_+ \qquad \forall a \in \mathcal{A}, t \in \mathcal{T}. \tag{4.1c}$$

The objective function (4.1a) minimizes the total cost, which is the sum of the first-stage cost (the first term) and the expected cost of the second stage (the second term) given a first-stage solution. Specifically, $\mathcal{Q}(\boldsymbol{X}, \mathcal{K}, \mathcal{B}) \approx \sum_s p^s Q_s(\boldsymbol{X}, \mathcal{K}^s, \mathcal{B}^s)$, where $Q_s(\boldsymbol{X}, \mathcal{K}^s, \mathcal{B}^s)$ is the second-stage cost of scenario s given a first-stage solution $\boldsymbol{X}$. The function $Q_s(\boldsymbol{X}, \mathcal{K}^s, \mathcal{B}^s)$ is called the *recourse function*, and $\mathcal{Q}(\boldsymbol{X}, \mathcal{K}, \mathcal{B})$ therefore the *expected recourse function*. The second-stage formulation will be introduced in Section 4.3.2. Constraints (4.1b) are CD vehicle balance constraints, while Constraints (4.1c) define the domain and type of the decision variables. It should be noted that the solution of this model creates a set of balanced

CD movements. Creating a set of CD routes from the balanced set of vehicle movements is straightforward.

4.3.2 Second-stage Formulation

The second stage formulation makes scenario-specific recourse decisions given a first-stage plan $\boldsymbol{x}$ (hence a vector of parameters), with each element x_a^t being the number of CD vehicles scheduled along arc a in time epoch t. For the sake of simplicity of notation, we drop superscript s in the notation used for a given scenario s with its commodities and crowdshippers. Let integer variable V_a^t be the number of SM drivers assigned to arc a at time epoch t. Similarly, let integer variable Z_a^t be the number of CS drivers assigned to arc a at time epoch t. Also, continuous variable $Y_{a,k}^t$ denotes the flow of commodity k sent along arc a at time epoch t. The second-stage formulation, representing the subproblem associated with a generic scenario s takes the following form.

$$Q_s(\boldsymbol{X}, \mathcal{K}^s, \mathcal{B}^s) = \text{Min} \quad \sum_{a\in\mathcal{A}}\sum_{t\in\mathcal{T}}(c_a^{SM}V_a^t + c_a^{CS}Z_a^t) \tag{4.2a}$$

$$\text{s.t.} \quad Z_a^t \le b_a^t, \qquad a \in \mathcal{A}, t \in \mathcal{T} \tag{4.2b}$$

$$\sum_{k\in\mathcal{K}} Y_{a,k}^t \le u^{CD}x_a^t + u^{SM}V_a^t + u^{CS}Z_a^t, \qquad a \in \mathcal{A}, t \in \mathcal{T} \tag{4.2c}$$

$$\sum_{a\in\delta(i)^+} Y_{a,k}^t - \sum_{a\in\delta(i)^-} Y_{a,k}^{t-} = \begin{cases} -q_k & \text{if } (i,t) = (o_k, t_k) \\ q_k & \text{if } (i,t) = (d_k, \sigma_k) \\ 0 & \text{otherwise} \end{cases}, i \in \mathcal{H}, t \in \mathcal{T}, k \in \mathcal{K} \tag{4.2d}$$

$$V_a^t, Z_a^t \in \mathbb{Z}_+, \qquad a \in \mathcal{A}, t \in \mathcal{T}. \tag{4.2e}$$

The objective function (4.2a) minimizes the total cost of the second stage, which is the sum of the costs of added CS and SM drivers. Constraints (4.2b) guarantee that the number of employed crowdshippers along an arc in a given time epoch is capped by the number of available ones according to the scenario. Constraints (4.2c) restrict the flow assigned to an

arc at a time epoch to the capacity along that arc provided by CD, SM, and CS vehicles. Constraints (4.2d) are flow conservation constraints. Finally, Constraints (4.2e) indicate the type and domain of the variables. Notice that this recourse function (4.2a) represents a *complete recourse* since for every first-stage solution $\boldsymbol{X}$, there exists at least one second-stage solution $(\boldsymbol{V}, \boldsymbol{Z})$.

4.4 Solution Approach

In this section, we discuss our proposed solution approach, PBBC, to solve the two-stage stochastic problem defined in Section 4.3, with integer variables in both the first and second stages. In the proposed PBBC approach, first, we adopt a partial Benders Decomposition (PBD) approach to improve the bounds of the master problem. Next, within a single branch-and-bound tree, we add different types of optimality cuts in two phases: In the initial phase, the linear relaxations of both the master problem and subproblems are solved to generate classical Benders Decomposition (BD) cuts. Subsequently, in the second phase, binary constraints are enforced on the decision variables of the master problem. At each node of the branch-and-bound tree, contingent upon the integrality of that solution, optimality cuts are generated using either the BD method, Integer L-shaped method (ILSM), or Benders Dual Decomposition(BDD) method. Additionally, our approach incorporates several acceleration approaches such as selective subproblems, parallelism, and ϵ-optimality techniques.

It is worth mentioning that the ILSM is mainly designed for problems with binary variables and BDD with integer subproblems requires the master-problem variables to be binary for convergence. Therefore, we convert the integer master problem variables, X_a^t, to binary variables assuming that X_a^t can take values from the set $\{0, 1, \ldots, M_a^t\}$, where M_a^t is the maximum number of CD vehicles that can be stationed at the tail hub of arc a at any time. Then, binary variable $\Gamma_a^{m,t}$ that gets value of 1 if X_a^t equals $m \in \{0, 1, \ldots, M_a^t\}$ is introduced. Next, X_a^t is substituted with $\sum_{m=0}^{M_a^t} m\Gamma_a^{m,t}$, and a new constraint $\sum_{m=0}^{M_a^t} \Gamma_a^{m,t} = 1$ is

added to the master problem. In the remainder of the paper, we work with this formulation, involving binary variables in the master problem.

In the following sections, we will first introduce the restricted master problem formulation of our PBBC approach in Section 4.4.1, then describe the different types of cut generation mechanisms in Section 4.4.2. Finally, in Section 4.4.3, we describe the high-level structure of the solution algorithm.

4.4.1 Branch-and-Benders-Cut (BBC) and Partial Benders Decomposition (PBD)

The process of Benders Decomposition involves three main steps: projection, dualization, and relaxation (72; 73). First, the model is projected onto the subspace defined by the first-stage decision variables. Then, the projected term is dualized to produce an equivalent model expressed as a set of valid inequalities (cuts) defining the feasibility requirements (feasibility cuts) and projected costs (optimality cuts) for the first-stage decision variables. Finally, a relaxation step is performed where a master problem and subproblems are iteratively solved to guide the search process and generate violated cuts, respectively.

Despite its promise to simplify the resolution of large-scale stochastic programs, the basic version of Benders Decomposition also presents significant drawbacks. The main disadvantage lies in the initial relaxation step, which produces a weak initial bound out of the master problem formulation. To improve such initial bounds, one needs to repeatedly solve the relaxed master problem to incorporate newly generated cuts, requiring significant computational efforts. Therefore, while Benders Decomposition offers a structured approach to address stochastic programs, its effectiveness is hindered by the limitations and complexities associated with the iterative process of generating and reintroducing cuts during the solution procedure.

Partial Benders Decomposition (PBD) methodology is one of the most efficient techniques to alleviate those drawbacks in the literature. As its efficiency shown in (45) for decreasing the number of optimality cuts, we artificially create an artificial scenario s' by taking the

convex combination of the scenarios such that $s' = \sum_{s \in \mathcal{S}} p^s s$. Let $\hat{V}_a^t, \hat{Z}_a^t, \hat{Y}_{a,k}^t$ denote the variables representing $V_a^t, Z_a^t, Y_{a,k}^t$ for scenario s' in the master problem. The problem with the scenario s' included in the restricted master problem, denoted RMP(PBD), takes the following form.

$$Z^{RMP(PBD)} = \min \sum_t \sum_a c_a^{CD} \sum_{m=0}^{M_a^t} m\Gamma_a^{m,t} + \sum_{s \in \mathcal{S}} p^s \theta^s \tag{4.3a}$$

$$\text{s.t.} \sum_{a \in \delta(i)^-} \sum_{m=0}^{M_a^t} m\Gamma_a^{m,t-} = \sum_{a \in \delta(i)^+} \sum_{m=0}^{M_a^t} m\Gamma_a^{m,t}, \qquad \forall i \in \mathcal{H}, t \in \mathcal{T} \tag{4.3b}$$

$$\hat{Z}_a^t \le b_a^t, \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{4.3c}$$

$$\sum_{k \in \mathcal{K}} \hat{Y}_{a,k}^t \le u^{CD} \sum_{m=0}^{M_a^t} m\Gamma_a^{m,t} + u^{SM} \hat{V}_a^t + u^{CD} \hat{Z}_a^t, \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{4.3d}$$

$$\sum_{a \in \delta(i)^+} \hat{Y}_{a,k}^t - \sum_{a \in \delta(i)^-} \hat{Y}_{a,k}^{t-} = \begin{cases} -q_k, & \text{if } (i,t) = (o_k, t_k) \\ +q_k, & \text{if } (i,t) = (d_k, \sigma_k) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{H}, t \in \mathcal{T}, k \in \mathcal{K} \tag{4.3e}$$

$$\sum_{t \in \mathcal{T}} \sum_{a \in \mathcal{A}} (c_a^{SM} \hat{V}_a^t + c_a^{CS} \hat{Z}_a^t) \le \sum_{s \in \mathcal{S}} p^s \theta^s \tag{4.3f}$$

$$\sum_{m=0}^{M_a^t} \Gamma_a^{m,t} = 1 \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{4.3g}$$

$$\Gamma_a^{m,t} \in \{0,1\} \qquad \forall m \in \{0,..,M_a^t\}, a \in \mathcal{A}, t \in \mathcal{T} \tag{4.3h}$$

$$\hat{V}_a^t, \hat{Z}_a^t, \hat{Y}_{a,k}^t \in \mathbb{R}_+ \qquad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{4.3i}$$

In model (4.3), θ^s represents an underestimator of the $Q_s(\boldsymbol{X}, \mathcal{K}^s, \mathcal{B}^s)$, which is the objective function of the subproblem, associated with each scenario s, given a first stage solution $\boldsymbol{x}$. Note that $\boldsymbol{x}$ is a fixed parameter in the subproblems rather than a variable. The objective function (4.3a) minimizes the total first stage cost and the expected recourse cost of the set of actual scenarios, $\mathcal{S}$. Constraints (4.3b) enforce vehicle flow balance.

Constraints (4.3c), (4.3d), and (4.3e) ensure the feasibility of the artificial scenario s'. Through Constraint (4.3f), the cost of the artificial scenario is linked to the underestimators θ^s, improving the quality of the approximation. Constraints (4.3g) link the binary $\Gamma_a^{m,t}$ variables to integer X_a^t variables. Constraints (4.3h) and (4.3i) define the variable domains.

We remark that in a setting with continuous second-stage variables, similar to (45), Constraint (4.3f) would be an equality. However, in our setting, due to the existence of integer second-stage variables V_a^t, Z_a^t, the equality does not hold. Therefore, we adapt the linking constraints to the setting with integer second-stage variables, by relaxing the equality requirement. This constraint holds since the recourse cost on linearly relaxed variables $\hat{V}_a^t, \hat{Z}_a^t$ constitute a lower bound on the convex combination of the subproblem costs with integer variables V_a^t, Z_a^t.

4.4.2 Cut Generation

Given the assumption of ample SM driver availability, the primal subproblems are never deemed infeasible. Hence, in this context, only optimality cuts are generated. To accelerate the branch-and-cut process, we implement three accelerating strategies that will be embedded in the cutting generation mechanisms explained in this section. These strategies are the followings. (1) Inspired by (111), our method exclusively integrates nondominated cuts into the master problem. We proceed as follows. For the first μ iterations of back and forth between the master problem and subproblems, all scenario subproblems are solved (generating BD cuts for each subproblem). Then, scenarios are ranked based on their cuts' average historical violation levels. For the subsequent iterations, only the top $\sigma\%$ of subproblems based on their rankings are solved, and the rankings are potentially updated. If the top $\sigma\%$ subproblems do not produce any violated cuts, we resort to a sequential approach, addressing subproblems individually until a valid cut is identified. (2) To expedite the process of evaluating subproblems, parallel computing is utilized to concurrently solve multiple subproblems, assigning each subproblem to a distinct thread. (3) Akin to the

approach described in (131), we incorporate ϵ-optimality, ensuring that subproblems are solved within an optimality gap of $\epsilon\%$.

In our methodology, three types of optimality cuts with distinct structures are integrated: classical BD cuts, ILSM cuts, and BDD cuts. Whenever a solution is attained within the Branch-and-Bound (B&B) tree, the subproblems are solved, generating cuts from one of these categories. These cut-generating methods are discussed next.

Classical Benders Decomposition

In the classical BD method, the dual of the LP relaxation of each subproblem is solved and the dual variables are used to create the cut. Let $\pi_a^{(4.2b),t}, \pi_a^{(4.2c),t}, \pi_i^{(4.2d),t_k}$ for all i, a, t, k denote the dual variables associated with Constraints (4.2b), (4.2c), and (4.2d), respectively. Then, the objective function of the dual of the subproblem is given by

$$\text{Max} \sum_a \sum_t b_a^t \pi_a^{(4.2b),t} + \sum_a \sum_t u^{CD} x_a^t \pi_a^{(4.2c),t} + \sum_k q_k (\pi_i^{(4.2d),t_k}). \tag{4.4}$$

In the classical BD, if this dual polyhedron is not empty, then the dual is either unbounded or feasible. If it is unbounded, then the primal subproblem is infeasible (which does not happen in our setting). If the dual polyhedron is feasible, then the optimal value of θ^s should not be less than the value of objective function (4.4) of the dual of scenario s subproblem, forming an optimality cut. Therefore, the optimality cut takes the form of

$$\theta^s \geq \sum_a \sum_t b_a^t \pi_a^{(4.2b),t} + \sum_a \sum_t u^{CD} (\sum_{m=0}^{M_a^t} m \Gamma_a^{m,t}) \pi_a^{(4.2c),t} + \sum_k q_k \pi_{o_k}^{(4.2d),i}. \tag{4.5}$$

Algorithm 3 details the steps taken to generate classical BD cuts given a master problem solution and an ordered (if available) list of subproblems.

Algorithm 3: Classical BD cut generation

```
   // Inputs:
   // - Scored scenario list (list of scenarios with associated scores)
   // - Current MP solution (current master problem solution)
   // - σ% (percentage of scenarios to consider)
   // Outputs:
   // - Cuts list (list of violated cuts added)
   // - Updated scored scenario list (scenario list with updated scores)
 1 Function Main()
 2     Sort scenarios in descending order of scores
 3     Solve LP-SP for σ% · |S| scenarios                          // solve LP subproblems
 4     Generate σ · |S|% BD cuts                                   // generate classical BD cuts
 5     Add violated cuts to cut list
 6     while cut list is empty do
 7         Solve LP-SP for next scenario in list
 8         Generate a BD cut
 9         if cut is violated then
10             Add cut to cut list
11     end
12     Update scenario scores
```

Integer L-shaped Method

Since classical BD optimality cuts (4.5) rely on dual variables of subproblem constraints, the integrality requirements of the second-stage variables are relaxed to be able to construct the dual problem. As a result, the BD explained above does not guarantee integrality in the second stage. Integer L-shaped method (ILSM) (99) is able to generate optimality cuts without requiring dual solutions to the subproblems. Thus, the integrality in the second stage can be enforced. Let $\gamma_a^{m,t}$ denote the value of variable $\Gamma_a^{m,t}$ according to the solution of the master problem, which will be fixed in the subproblems. Also, let Ω_1 be the set of index combinations (a,t,m) such that $\gamma_a^{m,t}=1$, and similarly Ω_0 be the set of index combinations (a,t,m) such that $\gamma_a^{m,t}=0$. Also, let L be a lower-bound on $Q_s(\boldsymbol{\gamma},\mathcal{K}^s,\mathcal{B}^s)$ over $\Gamma_a^{m,t}$. The ILSM optimality cut is

$$\theta^s \geq \Big(Q_s(\boldsymbol{\gamma},\mathcal{K}^s,\mathcal{B}^s)-L\Big)\Big(\sum_{(a,t,m)\in\Omega_1}\gamma_a^{m,t}-\sum_{(a,t,m)\in\Omega_0}\gamma_a^{m,t}-|\Omega_1|\Big)+Q_s(\boldsymbol{\gamma},\mathcal{K}^s,\mathcal{B}^s). \tag{4.6}$$

Algorithm 4 details the steps taken to generate ILSM cuts given a master problem solution and an ordered (if available) list of subproblems.

Algorithm 4: ILSM cut generation

```
// Inputs:
// - Scored scenario list (list of scenarios with associated scores)
// - Current MP solution (current master problem solution)
// - σ% (percentage of scenarios to consider)
// Output:
// - Cuts list (list of violated cuts added)
// - Updated scored scenario list (scenario list with updated scores)
Function Main()
    Sort scenarios in descending order of scores
    Solve INT-SP for σ% · |S| scenarios                          // solve integer subproblems
    Generate σ · |S|% ILSM cuts                                  // generate Integer L-Shaped Method cuts
    Add violated cuts to cut list
    while cut list is empty do
        Solve INT-SP for next scenario in list
        Generate an ILSM cut
        if cut is violated then
            Add cut to cut list
    end
    Update scenario scores
```

Benders Dual Decomposition (BDD)

BDD method consists of reformulating the subproblems incorporating local copies of master problem variables, and using Lagrangian duality to price out the coupling constraints that link the local copies to the master problem variables (131).

Let $\hat{\Gamma}_a^{m,t}$ and $\gamma_a^{m,t}$ be the local copy of the master problem variables X_a^t and the optimal solution of the RMP(PBD), respectively. The primal Benders subproblem can be written as

$$Q_s(\boldsymbol{x}, \mathcal{K}^s, \mathcal{B}^s) = \text{Min} \quad \sum_{a\in\mathcal{A}}\sum_{t\in\mathcal{T}}(c_a^{SM}V_a^t + c_a^{CS}Z_a^t) \tag{4.7a}$$

$$\text{s.t.} \quad (4.2\text{b}), (4.2\text{c}), (4.2\text{d}), (4.2\text{e})$$

$$\sum_{m=0}^{M_a^t} m\hat{\Gamma}_a^{m,t} = \sum_{m=0}^{M_a^t} m\gamma_a^{m,t} \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{4.7b}$$

$$\hat{\Gamma}_a^{m,t}, V_a^t, Z_a^t \in \mathbb{R} \qquad \forall m \in \{0, .., M_a^t\}, a \in \mathcal{A}, t \in \mathcal{T}. \tag{4.7c}$$

Notice that although the master problem integer variables X_a^t are converted to binary variables $\Gamma_a^{m,t}$, the subproblems can still be solved based on X_a^t variables with quick conversion. This is done to decrease the optimization time of the subproblems. Let λ_a^t represent the

dual variables associated with Constraints (4.7b). For any first-stage feasible solution $\boldsymbol{x}$, the following generalized Benders cuts can be created (73; 66; 78; 65; 169):

$$\theta^s \geq Q_s(\boldsymbol{x}, \mathcal{K}^s, \mathcal{B}^s) + \sum_a \sum_t (\sum_{m=0}^{M_a^t} m\gamma_a^{m,t} - \sum_{m=0}^{M_a^t} m\hat{\Gamma}_a^{m,t})\lambda_a^{t\,*} \tag{4.8}$$

Then, cut (4.8) can be strengthened by solving Problem 4.9, where $\lambda_a^{t\,*}$ is generated in Model (4.7).

$$Q_s(\boldsymbol{x}, \mathcal{K}^s, \mathcal{B}^s) = \text{Min} \quad \sum_{a\in\mathcal{A}} \sum_{t\in\mathcal{T}} (c_a^{SM} V_a^t + c_a^{CS} Z_a^t) + \sum_{a\in\mathcal{A}} \sum_{t\in\mathcal{T}} \lambda_a^{t\,*} (\sum_{m=0}^{M_a^t} m\gamma_a^{m,t} - \sum_{m=0}^{M_a^t} m\hat{\Gamma}_a^{m,t}) \tag{4.9a}$$

$$\text{s.t.} \quad (4.2b), (4.2c), (4.2d), (4.2e)$$

$$\hat{\Gamma}_a^{m,t} \in \mathbb{Z}_+ \qquad \forall m \in \{0, .., M_a^t\}, a \in \mathcal{A}, t \in \mathcal{T}. \tag{4.9b}$$

$$V_a^t, Z_a^t \in \mathbb{R}_+ \qquad a \in \mathcal{A}, t \in \mathcal{T}. \tag{4.9c}$$

(131) illustrate that when the solution $\boldsymbol{x}$ from the master problem is integer, the strength of the optimality cut (4.8) is, at most, equivalent to classical Benders cuts. Conversely, for fractional solutions, (4.8) offers an additional advantage.

Algorithm 5 details the steps taken to generate BDD cuts given a master problem solution and an ordered (if available) list of subproblems.

Algorithm 5: BDD cut generation

```
// Inputs:
// - Scored scenario list (list of scenarios with associated scores)
// - Current MP solution (current master problem solution)
// - σ% (percentage of scenarios to consider)
// Output:
// - Cut list (list of violated cuts added)
// - Updated scored scenario list (scenario list with updated scores)
Function Main()
    Sort scenarios in descending order of scores
    Solve Model (4.7) for σ% · |S| scenarios                          // solve Model (4.7)
    Generate σ · |S|% BDD cuts                                        // generate BDD cuts
    Strengthen cuts by solving 4.9 for σ% · |S| scenarios             // strengthen BDD cuts using Model 4.9
    Add violated cuts to cuts list
    while cut list is empty do
        Solve Models (4.7) and (4.9) for the next scenario in list
        Generate a strengthened BDD cut
        if cut is violated then
            Add cut to cuts list
    end
    Update scenario scores
```

4.4.3 Solution Algorithm

Figure B.1 depicts the steps of our solution approach which is also outlined in Algorithm 6. Our solution approach integrates three types of optimality cuts: classical BD cuts, ILSM cuts, and BDD cuts, added to the RMP(PBD). Drawing upon the foundational work by (114) and its subsequent applications in studies such as (132; 131), our solution methodology unfolds in two distinct phases.

Initial Phase. In this phase, we start by solving the linear relaxation of the RMP(PBD), denoted LP-RMP(PBD) (line 7), incorporating classical BD cuts generated based on the LP relaxations of the subproblems, LP-SPs. Over the first μ solutions of the master problem, all LP-SPs are solved and any violated cuts are appended to the LP-RMP(PBD) (lines 10-11). Subsequently, the LP-SPs are scored based on their average violation levels (line 12), where the violation of a subproblem is quantified as the difference between its objective function, denoted as $Q_s(x, \mathcal{K}^s, \mathcal{B}^s)$, and its underestimator, θ^s. The subproblems are ordered according to this score (decreasing order of violation). Following the solution of a subproblem, its score is updated as the average violation across the solutions obtained. After each subsequent master problem solution, i.e., $iter > \mu$, only the top $\sigma\%$ of the LP-SPs are solved, and any violated cuts are integrated into the master problem. If none of the $\sigma\%$ LP-SPs generates a violated cut, the remaining LP-SPs are solved following the same ranking until a cut is identified. If none of the remaining subproblems can generate a violated cut, then the LP-RMP(PBD) has converged, and the second phase begins (line 23).

Second Phase. In this phase, binary constraints are imposed on the RMP(PBD) (line 24) and a branch-and-cut tree is formed to initialize the solution of RMP(PBD). The same tree is utilized until convergence, with cuts being added to the RMP(PBD) through callbacks. For every identified fractional solution of RMP(PBD), the BDD subproblems are solved, and any violated cuts are added to the RMP(PBD) (lines 30-32). Alternatively, for each integer solution discovered, initially, the top $\sigma\%$ LP-SPs according to their scores, are solved

to generate classical BD cuts (lines 26-29). If no violated cut is generated, the remaining LP-SPs are solved until a violated cut is found. If none of the LP-SPs exhibit violations, then the top $\sigma\%$ integer subproblems (INT-SPs) are solved to produce ILSM cuts (line 38). If no violated cut is found, then the remaining INT-SPs are solved following the established ranking. If no violated cuts are found among the remaining INT-SPs, the UB is updated (line 42). It should be noted that, the UB of the RMP(PBD) is only updated when none of the ILSM cuts are violated. Thus, in order to ensure the optimality of the solution, we do not implement ϵ-optimality to the subproblems in Algorithm 4.

4.5 Partially Adaptive Stochastic Programming Approach

In an environment where the information is revealed over time, a pre-determined set of decisions may prevent the system from achieving the best performance (16). Therefore, being able to revise the decisions made at the tactical level, could result in cost savings. In our setting, the first-stage decisions, i.e., the set of a priori routes designed at the tactical level are examples of such decisions. However, making updates too frequently might be impractical due to (1) the limited time available to the decision-makers to re-optimize the system, and (2) the complications associated with changing drivers' schedules during the operational period or other contractual limitations. Thus, the decision maker may be interested in assessing the potential improvement in efficiency and performance of the system under different updating policies within the limits of available system flexibility.

In the context of the courier company considered in this study, the corporate drivers are assigned to jobs according to the tactical plan at the beginning of an operational period. Then, during the operational period, more accurate information about demand and CS driver availability may be obtained. Let us assume that at certain times during the operational period (potentially while the tactical plans are partially executed), the plans can be revised, considering the system state at that time. The system state at any epoch $|\mathcal{T}| - \tau$ during the current operational period consists of the status of in-use CD, SM, and CS drivers (See

Algorithm 6: Benders Decomposition Algorithm

```
   // Inputs:
   // - 𝒢:  graph representing the service network
   // - 𝒮:  set of scenarios
   // - All problem parameters
   // Output:
   // - Optimal solution to the master problem
 1 Function Main()
 2 |   InitialPhase()
 3 |   SecondPhase()
 4 Function InitialPhase()
 5 |   Initialize LP-RMP(PBD) {LP Relaxation of the RMP(PBD)}
 6 |   iter := 0
 7 |   Solve LP-RMP(PBD) to obtain tentative solution and dual bound
 8 |   while LP-RMP(PBD) not converged do
 9 |   |   if iter ≤ μ then
10 |   |   |   Generate cut list using Algorithm 3 (σ% = 100)
11 |   |   |   Add cuts to LP-RMP(PBD)
12 |   |   |   Score scenarios based on avg. violations
13 |   |   else
14 |   |   |   Generate cut list using Algorithm 3 (σ% < 100)
15 |   |   |   if cut list is not empty then
16 |   |   |   |   Add cuts to LP-RMP(PBD)
17 |   |   |   else
18 |   |   |   |   LP-RMP(PBD) converged
19 |   |   |   end
20 |   |   end
21 |   |   iter++
22 |   end
23 Function SecondPhase()
24 |   Initialize B&C tree to solve RMP(PBD); {RMP(PBD) with binary constraints}
25 |   Initialize LB, UB
26 |   while LB < UB or unfathomed node exists do
27 |   |   for each node solved on the B&C tree do
28 |   |   |   Update LB
29 |   |   |   if solution is fractional then
30 |   |   |   |   Generate cut list using Algorithm 5 (σ% < 100)
31 |   |   |   |   if cut list is not empty then
32 |   |   |   |   |   Add cuts to RMP(PBD)
33 |   |   |   else
34 |   |   |   |   Generate cut list using Algorithm 3 (σ% < 100)
35 |   |   |   |   if cut list is not empty then
36 |   |   |   |   |   Add cuts to RMP(PBD)
37 |   |   |   |   else
38 |   |   |   |   |   Generate cut list using Algorithm 4 (σ% < 100)
39 |   |   |   |   |   if cut list is not empty then
40 |   |   |   |   |   |   Add cuts to RMP(PBD)
41 |   |   |   |   |   else
42 |   |   |   |   |   |   Update UB
43 |   |   |   |   |   end
44 |   |   |   |   end
45 |   |   |   end
46 |   |   end
47 |   end
```

Figure 4.1: Planning horizon and operational period representation

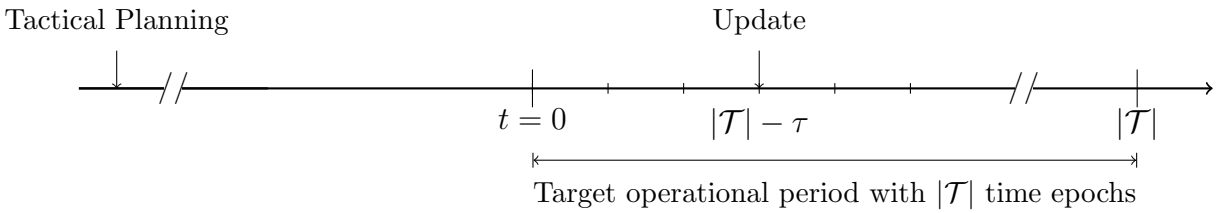


Figure 4.1). The decision-maker also has full information regarding CS driver availability and commodities arrived within the time interval $[t = 0, |\mathcal{T}| - \tau]$ and potentially has a more accurate estimation regarding events to happen within $(|\mathcal{T}| - \tau, |\mathcal{T}|]$.

At time $|\mathcal{T}| - \tau$, a driver might be at a hub involved in loading/unloading activities or on-route to his/her next destination. Analogously, the freight associated with commodities arrived by $|\mathcal{T}| - \tau$ are either at a hub waiting to be loaded on a vehicle, on a vehicle moving toward their next destinations, or already arrived at their final destinations. Note that, as can be from our Model (4.2), the flow associated with a commodity may be split along the way and different parts may go on different vehicles and even on different paths.

Updating the tactical plans at a given time epoch $|\mathcal{T}| - \tau$ consists of constructing a new instance of the problem in compliance with the state of the system at that time. Also, given the improved level of forecast of future events (future commodities and CS driver availability over the remaining τ epochs), one can create a new set of scenarios to be used in the updated Models (4.1) and (4.2). The updated instance is created along the following rules.

- CD vehicles: The number of CD vehicles in use and their locations at the time of update must be explicitly enforced in the master problem;

- Commodities: New commodities associated with in-transfer commodities are generated. Specifically, any non-zero flow $Y_{a,k}^{t-}$ associated with commodity k with $t \geq |\mathcal{T}| - \tau$ and a departure time from the tail of a prior to $|\mathcal{T}| - \tau$ becomes a new commodity $k' : \langle o_{k'} := h_a, d_{k'} := d_k, t_{k'} := t, q_{k'} := Y_{a,k}^{t-} \rangle$, where h_a represents the head hub of arc a.

In the new problem, $|\mathcal{T}| - \tau$ is considered as the start of the operational period, and the time represented by the remaining τ time epochs constitutes the new operational period. This update can be done once or multiple times during a given operational period depending on the flexibility the contracts with CD drivers avail. The potential improvements in performance as a result of different updating policies are analyzed in Section 4.6.3.

4.6 Computational Results

To evaluate the effectiveness of our proposed approach and gain valuable managerial insights, we conduct a set of computational experiments using a set of instances generated. In Section 4.6.1, we explain the experimental setup and instance/scenario generation procedures. Then, in Section 4.6.2, we evaluate the effect of enforcing integrality in the second-stage variables in terms of cost and solution time. In Section 4.6.2, we perform a sensitivity analysis on the solution approach to evaluate its performance under various circumstances. Additionally, in Section 4.6.2, we examine the effect of the BD acceleration methods, discussed at the beginning of Section 4.4.2. Finally, in Section 4.6.3, we explore the potential benefits of implementing the PASSND.

The tests were coded in Python 3.6.5 and ran on a machine with 2×2.4 GHz Intel Xeon E5-2640 v4 processor and 96 GB of RAM, linear problems were solved using Gurobi 9.0.3.

4.6.1 Data and Instance Generation

The instances are defined by a collection of deterministic and stochastic parameters, as outlined in Section 4.2. In Section 4.6.1, we describe the network structure and deterministic parameters considered. The stochastic parameters and the scenario generation procedure are explained in Section 4.6.1.

Instance Generation.

We conduct our experiments over a stylized graph $\mathcal{G}$ which takes the form of a solid grid with $|\mathcal{H}|$ nodes. Graph $\mathcal{G}$ is assumed to be incomplete, with each hub linked to its neighbors

Table 4.1: Parameter configurations considered in the design of experiments

Attribute	Description	Level Values
$\|\mathcal{H}\|$	Number of hubs	{5, 10, 20, 30}
$\|\mathcal{T}\|$	Number of time epochs in an operational period	$\{16, 32\}$
$\|\mathcal{K}\|$	Number of commodities	$\{10, 20, 30, 40, 50, 75, 100, 125\}$
$\|\mathcal{S}\|$	Total scenarios generated	{10, 50, 100, 200200}
u^{CS}, u^{SM}, u^{CD}	Vehicle capacities for CS, SM, CD	{10, 20, 100}
CS_{avail}	Crowdshipper availability (%)	$\{0\%, 10\%, 20\%, 30\%\}$
T	Operational period (hrs)	$\{8\}$
R	Operational period and service commitment (hrs)	$\{4\}$
κ	Similarity percentage for scenario generation	$\{30\%\}$
μ	Parameter for scoring subproblems	$\{2\}$
σ	Percentage of scenarios considered in each iteration	$\{20\%\}$
ϵ	Subproblem optimality parameter	$\{0.5\%\}$

via horizontal, vertical, and diagonal arcs. Loading, unloading, and sorting at facilities are assumed to take a fixed amount of time (0.2 hours) and are thus added to the arc lengths.

The computational experiments are designed based on the values considered for different attributes of the problem as given in Table 4.1. We consider an operational period of 8 hours that is discretized into $|\mathcal{T}|$ equal-length time epochs when $|\mathcal{T}| \in \{16, 32\}$. We solve 5 instances for each combination and report the average results. The service commitment R is assumed to be 4 hours and the same for all commodities.

Scenario Generation.

We construct a set of scenarios to formulate stochasticity in our problem. Each scenario represents a potential realization of stochastic parameters, i.e., crowdshipper (CS) availability at each arc during each time epoch, as well as the set of commodities that will arrive at specific times in the system. First, we generate a single scenario, denoted SA to represent the actual sequence of events. Specifically, scenario SA corresponds to the events that will occur in the future, and will only be revealed to the decision-maker as the events unfold. However, in the presence of forecasts, the decision maker is capable of generating scenarios that are to some degree similar to the actual scenario SA.

The availability of the crowdshippers in SA is determined based on CS_{avail} parameter, which defines the percentage of arcs that have at least one CS at a specific time epoch. The existence of at least one crowdshipper on a given arc a at a time epoch t is determined by drawing a random number between 0 and 1 for each pair (a, t). If the random number is below CS_{avail}, at least one CS driver is assigned to pair (a, t). If at least one CS is determined to be available on a given (a, t), the quantity of available CSs on the pair is sampled from a Poisson distribution with $CS_{avg} = 2$ parameter. To generate the commodities of the actual scenario SA, a total of $|\mathcal{K}|$ commodities are randomly sampled from $|\mathcal{N}| \times (|\mathcal{N}|-1) \times |\mathcal{T} - \mathcal{R}|$ origin-destination-epoch combinations. The demand volume of each commodity is sampled from a Poisson distribution with q_{avg} parameter. We consider four levels of crowdshipper availability as $CS_{avail} \in \{0\%,\ 10\%,\ 20\%,\ 30\%\}$, with an average commodity demand volume of $q_{avg} = 25$.

Once the actual scenario SA is generated, a set of scenarios $\mathcal{S}$ is generated according to a *similarity percentage*, κ, to SA, each with the same probability of occurrence p^s such that $\sum_{s \in \mathcal{S}} p^s = 1$. A scenario is considered "*similar*" to the actual scenario SA if at least $\kappa\%$ of its crowdshippers are on the same (a, t) pairs as in SA and at least $\kappa\%$ of its commodities are the same as commodities in SA in terms of origin, destination, and arrival time. The $\kappa\%$ commodities and crowdshippers that are similar to the actual scenario may vary for each scenario, and the remaining $(1-\kappa)\%$ is randomly selected for each scenario. Specifically, the number of available CSs and the volumes of commodities are re-sampled for each scenario, for both the similar and different parts of the scenarios.

The similarity percentage κ used in scenario generation can be seen as the forecasting accuracy of the courier company about the future demand and CS availability. In practice, the accuracy of forecasts is a function of how far in the future the forecast is made for. Therefore, at the tactical level planning, this accuracy is at its lowest level. In our implementations, the scenarios are generated based on a similarity percentage $\kappa = 30\%$, which represents a high level of variability of the stochastic parameters across the scenarios at the tactical level.

At the operational level, however, the accuracy could significantly improve, although the decision maker may have limited opportunity to benefit from the more accurate information due to restrictions on making adjustments to the plans. In Section 4.6.3, we study the potential benefit of taking advantage of improved forecast accuracy at the beginning or during the operational phase, if updating plans is permitted.

A total of 200 scenarios are generated, and the convergence of Benders Decomposition is checked to the tolerance level of $1e-6$. The parameters related to scoring subproblems are set to $\mu = 2$ and $\sigma = 20\%$, and the subproblem optimality parameter ϵ is set to 0.5% as in (131).

4.6.2 Results of the Numerical Analysis

We consider a default setting based on the following parameter values, unless stated otherwise: $CS_{avail} = 30\%$, $|\mathcal{T}| = 16$, $|\mathcal{S}| = 200$, and $|\mathcal{H}| = 10$. All the reported results are averages over 5 instances of a given size combination.

Performance of Solution Approach

In Table 4.2, we provide a comprehensive evaluation of our solution approach across a diverse test bed consisting of 14 instance classes, organized into three distinct segments. Initially, we investigate the effects of varying the number of scenarios from 10 to 200, finding no significant impacts beyond 200 scenarios. Subsequently, we analyze the influence of different numbers of commodities, ranging from 20 to 125. In the third segment, we examine the impact of varying the size of the physical network, considering 10, 20, and 30 hubs. For each instance class, we present the total cost breakdown of CD, CS, and SM, along with the total number of CD routes and the solution time in hours, categorized into the Initial Phase and Second Phase.

Upon analyzing the influence of the number of commodities, $|\mathcal{K}|$, in the second segment, we observe a proportional increase in both total cost and solution time with higher commodity numbers. Within the 12-hour time frame, our approach effectively handles up to 125

Table 4.2: Evaluation of the solution approach, examining impacts of number of scenarios, commodities, and hubs

Instance			Cost				#CD	Time (hrs)		
$\|\mathcal{H}\|$	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	Total	CD	CS	SM	routes	Initial Ph.	Second Ph.	Total
5	**10**	**10**	\$127.4	\$59.9	\$32.6	\$34.9	3.1	0.0	0.0	0.0
5	**10**	**50**	\$120.3	\$48.1	\$25.8	\$46.3	2.1	0.0	0.1	0.1
5	**10**	**100**	\$121.2	\$48.4	\$23.5	\$49.4	2.0	0.0	0.3	0.4
5	**10**	**200**	\$122.0	\$48.4	\$20.4	\$53.1	1.2	0.0	0.6	0.6
10	**20**	**200**	\$219.1	\$113.7	\$59.6	\$45.8	3.4	0.1	1.4	1.5
10	**30**	**200**	\$303.7	\$132.4	\$79.9	\$91.4	5.2	0.3	2.5	2.8
10	**40**	**200**	\$383.9	\$175.4	\$112.1	\$96.4	5.4	0.3	2.6	3.0
10	**50**	**200**	\$458.6	\$168.8	\$135.8	\$154.1	6.4	0.5	2.8	3.3
10	**75**	**200**	\$613.1	\$252.0	\$201.7	\$159.4	9.4	0.6	4.6	5.1
10	**100**	**200**	\$733.5	\$361.6	\$206.9	\$165.0	12.2	1.0	6.6	7.5
10	**125**	**200**	\$834.7	\$386.5	\$237.1	\$211.2	16.6	0.9	9.3	10.2
10	**30**	**200**	\$303.7	\$132.4	\$79.9	\$91.4	5.2	0.3	2.5	2.8
20	**30**	**200**	\$413.5	\$146.4	\$104.2	\$162.9	4.6	0.6	4.8	5.4
30	**30**	**200**	\$482.8	\$125.5	\$122.1	\$235.1	3.4	1.3	9.2	10.5

commodities when $|\mathcal{H}| = 10$ and $|\mathcal{S}| = 200$. Additionally, as the network demand density increases from $|\mathcal{K}| = 20$ to $|\mathcal{K}| = 125$, there is a significant rise in the number of CD routes from 3.4 to 16.6, driven by increased consolidation opportunities. Correspondingly, the total cost goes up from \$219.1 to \$834.7 (2.8 times increase), despite a 5.25 times increase in the number of commodities, due to the higher consolidation rate. In the third segment, when $|\mathcal{K}| = 30$ and $|\mathcal{S}| = 200$, our approach efficiently handles up to $|\mathcal{H}| = 30$ within the 12-hour time constraint. However, increasing the number of hubs while maintaining a fixed number of commodities leads to a decrease in network demand density, resulting in reduced consolidation opportunities, and reducing the attractiveness of CD routes. This trend is evidenced by the decline in the number of CD routes.

Evaluation of the Components of the Solution Approach

In this section, we investigate the influence of acceleration methods discussed at the beginning of Section 4.4.2, on the convergence time within our solution approach. Table 4.3 presents an evaluation of these methods across 14 instance classes. To assess their impact, we run the solution approach while excluding one method at a time and compare the convergence times in separate columns. Specifically, the version labeled "Solve all subproblems" entails

Table 4.3: Evaluation of solution approach components on solution time across different instance classes

Instances			Complete Solution Time (hrs)	Solution Time Increase %				
$\|\mathcal{H}\|$	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$		No selective SPs	No PBD	No ϵ-opt.	No BDD cuts	No Parallel SPs
5	**10**	**10**	0.02	8%	0%	4%	17%	1%
5	**10**	**50**	0.1	12%	-2%	4%	15%	1%
5	**10**	**100**	0.4	12%	4%	4%	19%	3%
5	**10**	**200**	0.6	15%	2%	5%	20%	4%
10	**20**	**200**	1.5	15%	8%	5%	16%	8%
10	**30**	**200**	2.8	12%	8%	4%	17%	8%
10	**40**	**200**	3.0	15%	10%	5%	24%	8%
10	**50**	**200**	3.3	18%	11%	3%	22%	11%
10	**75**	**200**	5.1	22%	12%	4%	20%	13%
10	**100**	**200**	7.5	22%	11%	4%	20%	13%
10	**125**	**200**	10.2	25%	13%	5%	24%	15%
10	**30**	**200**	2.8	12%	7%	4%	17%	8%
20	**30**	**200**	5.4	17%	6%	5%	24%	11%
30	**30**	**200**	10.5	20%	6%	5%	23%	13%

not implementing the subproblem scoring mechanism, resulting in solving all subproblems after finding a master problem solution. The version "No PBD" denotes the absence of the partial Benders Decomposition approach in the master problem. Similarly, "No ϵ-opt" involves solving all subproblems to optimality, while "No BDD cuts" indicates the exclusion of the Benders Dual Decomposition cuts. "No Parallel SPs" entails solving all subproblems in a series rather than in parallel.

From Table 4.3, it becomes apparent that BDD cuts exert the most significant effect across a majority of the instance classes, followed by our proposed selective subproblems method. Selective subproblems strategy becomes the top contributor in the instance classes with a large number of scenarios and commodities, i.e., $|\mathcal{H}| = 10,\ |\mathcal{K}| \in \{75, 100, 125\},\ |\mathcal{S}| = 200$. The PBD approach exhibits moderate positive impact, particularly for very low $|\mathcal{S}|$ values and small instances, although it demonstrates an increasing trend with higher network demand density in larger instances. Conversely, the overall impact of ϵ-optimality on solution time is comparatively minor, with consistent effects observed across different instance classes.

Enforcing Integrality in the Second Stage

The classical BD does not guarantee the integrality of the second-stage variables, as it solves the dual of the linear relaxation of the subproblems. In contrast, in our PBBC approach, the subproblems solved for generating ILSM cuts enforce the binary nature of the second-stage variables. This ensures the integrality of both first- and second-stage variables in the optimal solution. Given the tactical nature of decisions in our setting, one may settle for only integer-first-stage solutions as considering the subproblems associated with different scenarios is for the sole purpose of improving the estimation of the second-stage cost. Alternatively, one can initially drop the integrality requirements for subproblems (disabling ILSM), and solve the integer first-stage problem. Subsequently, given the obtained first-stage solution, subproblems are solved individually while enforcing integrality. This approach will likely result in a suboptimal solution, however, could also require less computational effort, and hence be of interest to some decision-makers. These approaches will be referred to as PBBC "with" and "without" ILSM, respectively.

Next, we aim to evaluate the efforts required to ensure the integrality of the second-stage variables using ILSM within the solution approach. To make a meaningful comparison, we set $\epsilon = 0$ for both PBBC implementations (with and without ILSM). The implementation of the PBBC approach without ILSM closely resembles that of the approach with ILSM. In the initial phase, it mirrors that of the approach with ILSM. However, in the second phase of the algorithm upon enforcing integrality requirements to the master problem, no ILSM cuts are generated (lines 38-40 of Algorithm 6). Subsequently, all subproblems are resolved once more, and the total expected cost is updated accordingly.

Table 4.4 shows the percentage increase in total expected cost and the corresponding decrease in solution time observed from the PBBC approach with ILSM to PBBC without ILSM. We observe that the version without ILSM consistently results in higher expected cost, with a maximum increase of 8.1% and an average increase of 5.1% across 14 reported

Table 4.4: Comparison of PBBC approaches with and without ILSM, showing changes in total expected cost and solution time across different instance classes

Instances			Cost	Time
$\|\mathcal{H}\|$	$\|\mathcal{K}\|$	$\|\mathcal{S}\|$	increase %	decrease %
5	**10**	**10**	5.0%	1.2%
5	**10**	**50**	5.1%	2.7%
5	**10**	**100**	8.1%	2.1%
5	**10**	**200**	3.1%	1.5%
10	**20**	**200**	4.1%	4.8%
10	**30**	**200**	3.3%	7.4%
10	**40**	**200**	5.4%	9.9%
10	**50**	**200**	5.1%	8.3%
10	**75**	**200**	6.4%	9.5%
10	**100**	**200**	4.8%	7.5%
10	**125**	**200**	5.6%	6.7%
10	**30**	**200**	3.3%	7.4%
20	**30**	**200**	7.0%	9.7%
30	**30**	**200**	5.8%	9.7%

instance classes. However, it requires less solution time, with reductions ranging from 1.2% to 9.9%, averaging 6.3% across the same instance classes.

Effect of CS Availability (CS_{avail}) on Cost and Vehicle Usage.

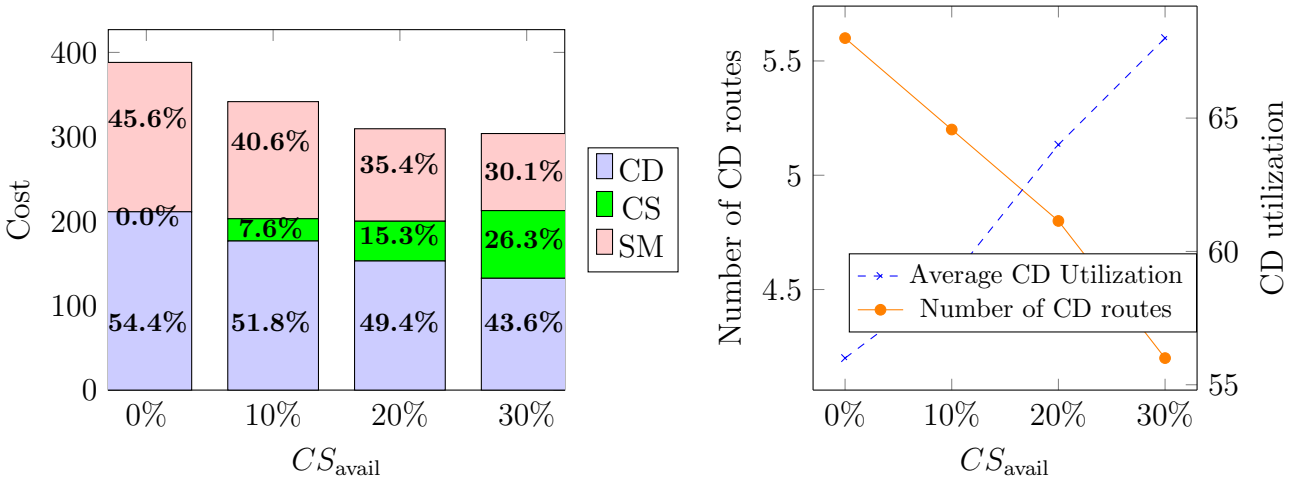
Figure 4.2a presents the impact of CS availability (CS_{avail}) on the expected total cost of the system, as well as the cost breakdown between the three driver types. We present the average cost of all the default parameter settings for four different values of CS_{avail}. As can be observed from Figure 4.2a, CS availability has a direct effect on the total operational costs. Specifically, the highest total cost corresponds to $CS_{avail} = 0\%$, which represents the case with no CS drivers, and the lowest total cost corresponds to $CS_{avail} = 30\%$ which is the case with the highest level of CS availability considered in these analyses. It is worth noting that since CS availability and commodities are generated independently in the network, a certain increase in CS availability does not translate into the same level of increase in employed CS drivers. It is observed that increasing the CS_{avail} from 0% to 10% leads to a 12% reduction in the total cost, while further increasing it to 20% results in a 21% reduction

over $CS_{avail} = 0\%$, and increasing the CS availability to 30% results in around 22% cost reduction over $CS_{avail} = 0\%$.

The results show that the use of CSs can reduce both the CD cost and the dependency of the system on the spot market. This observation indicates the benefit of employing CDs and facilitating their participation as much as possible, especially on arcs with a higher flow of shipments.

In Figure 4.2b, for each value of CS_{avail}, we show the average number of CD routes generated and the average CD utilization over the 5 instances tested. The increase in CS_{avail} from 0% to 30% decreased the number of CD routes by around 25% while the average utilization went up from 56% to 68%. These results show that being able to employ crowdshippers would allow the service provider to reduce the number of required corporate vehicles, and improve the efficiency of vehicle usage.

Figure 4.2: The effect of CS availability parameter on cost and CD utilization



(a) CS_{avail} effect on cost

(b) CS_{avail} effect on # of CDs

Next, we assess the effect of using CSs in terms of vehicle and freight movements in the network. Table 4.5 provides comparative data across two scenarios, i.e., $CS_{avail} = 30\%$ and $CS_{avail} = 0\%$ for the instances with $|\mathcal{H}| = 10$, $|\mathcal{K}| = 30$, $|\mathcal{S}| = 200$. The evaluation spans several key metrics, including the number of vehicle movements, the average length of selected vehicle movements, vehicle utilization rates, and the associated costs, broken down

by vehicle type. Moreover, Figures 4.3 and 4.4 visually depict the edges covered by different vehicle types and their movement frequencies under these two scenarios, where thicker links represent a higher frequency of vehicle movements over time.

Table 4.5: Comparison of solutions when $CS_{avail} = 30\%$ and $CS_{avail} = 0\%$

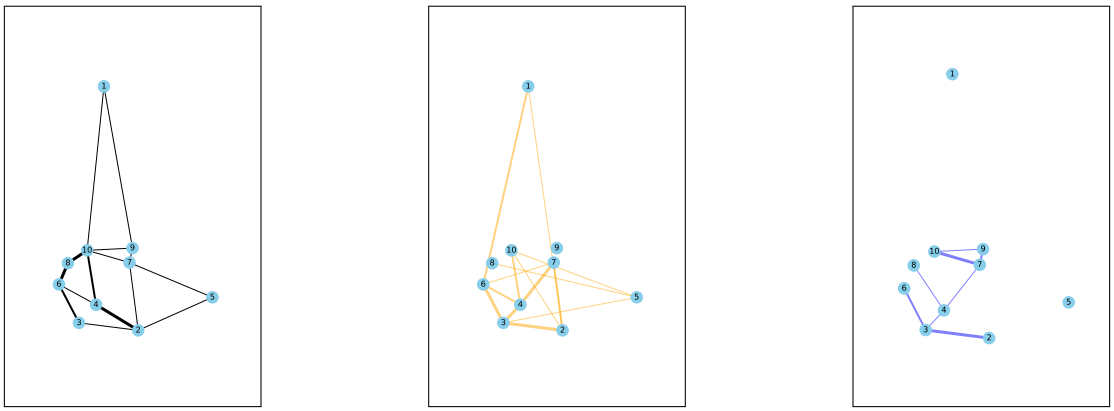
CS avail	#vehicle movements				Avg. movement length (mins)			Avg. movement load			Avg. movements per commodity				Cost			
	CD	CS	SM	Total	CD	CS	SM	CD	CS	SM	CD	CS	SM	Total	CD	CS	SM	Total
30%	23.2	26.1	14.2	63.5	13.4	16.4	7.8	73.5	9.5	18.3	2.4	0.4	0.2	3.0	$132.4	$79.9	$91.4	$303.7
0%	32.5	-	23.5	56.0	16.8	-	11.1	56.1	-	17.2	2.8	-	0.4	3.3	$210.7	-	$176.5	$387.2

The data illustrate a notable reduction in the total number of vehicle movements when CS availability decreases from 30% to 0%, dropping from 63.5 to 56.0 movements. Additionally, there is an observable increase in the average length of CD and SM movements in the absence of CS drivers. The movement length for CDs increases from 13.4 to 16.8 minutes, and for SMs, from 7.8 to 11.1 minutes. Concurrently, the absence of CSs leads to declining vehicle utilization efficiency, with CD utilization decreasing from 74% to 56%. This reduction in utilization rate could indicate that despite traveling longer distances, the CDs are carrying less per trip relative to their capacity, reflecting less efficient use of resources. Removal of CSs will result in increasing the number of CD and SM movements, and an increase in the total cost by 27.5%.

Figures 4.3 and 4.4 support these findings, with CS vehicles shown to operate longer arcs, while SM vehicles predominantly handle shorter arcs, typically moving commodities to neighboring hubs for consolidation. Notably, comparing 4.3a and 4.4a, there is a discernible increase in the frequency of longer arcs in CD movements, compensating for the absence of CS shipments. Additionally, the average number of vehicle movements per commodity increases slightly from 3.0 to 3.3 when CS availability decreases from 30% to 0%. This increase suggests that the absence of CSs can lead to a slight rise in the number of stopovers. The longer arcs operated by CSs will allow moving shipments closer to their destinations, resulting in a potentially smaller number of stopovers for commodities.

In summary, it is best to employ (and encourage the participation of) CSs on longer arcs connecting main consolidation points and between low-demand hub pairs. Due to their cost differences, SMs and CSs are used substantially differently, that is, the solution in the presence of CSs would not be obtained by a simple replacement of SMs in the solution without CSs with CSs.

Figure 4.3: Vehicle movement frequency on the physical network $|\mathcal{H}| = 10,\ |\mathcal{K}| = 30,\ |\mathcal{S}| = 200,\ CS_{avail} = 30\%$



(a) CD movements (b) CS movements (c) SM movements

Figure 4.4: Vehicle movement frequency on the physical network $|\mathcal{H}| = 10,\ |\mathcal{K}| = 30,\ |\mathcal{S}| = 200,\ CS_{avail} = 0\%$

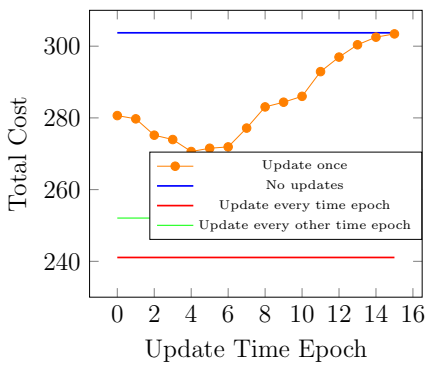
(a) CD movements (b) SM movements

4.6.3 Analyzing the Benefits of the PASSND

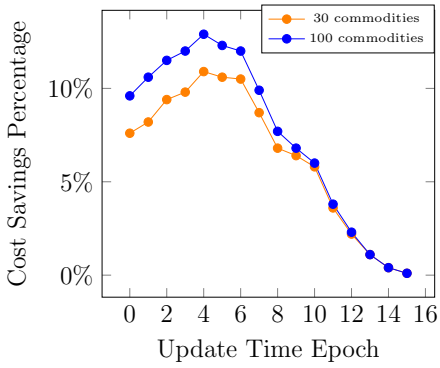
As stated in Section 4.2, plans designed based on resources acquired through the forward market often allow for no to limited updates. In this section, we analyze the potential benefit of the PASSND, i.e., revising/updating such plans at the operational level as the exact information regarding demand and CS availability unfolds, if such an update is permitted. We conduct these tests in two settings: (1) $|\mathcal{T}| = 16$, and $R = 8$, and (2) $|\mathcal{T}| = 32$, and $R = 8$, and report the average values for 5 instances for each setting. The PASSND involves re-optimizing the system during operation with updated information. This requires updating the set of scenarios based on the new information revealed. Specifically, the constant similarity percentages of scenarios are replaced with similarity percentages that linearly decrease over the remaining time epochs of the operational period. Let $\{\kappa_0, \kappa_1, \ldots, \kappa_{|\mathcal{T}|}\}$ be the updated similarity percentages over the next immediate $|\mathcal{T}|$ time epochs of the operational phase. We assume that $\kappa_0 = 100\%$, and $\kappa_{|\mathcal{T}|}$ is greater than the similarity percentage used at the tactical level. Now, if an update at time epoch $t = |\mathcal{T}| - \tau$ is of interest, the similarity percentages $\{\kappa_0, \kappa_1, \ldots, \kappa_\tau\}$ are considered respectively for time epochs $|\mathcal{T}| - \tau, |\mathcal{T}| - \tau + 1, \ldots, |\mathcal{T}|$. This approach relies on the idea that the decision maker has access to full information regarding the stochastic parameters of that epoch, and therefore, all scenarios contain the same information as the actual scenario SA for the current time epoch. The knowledge about the subsequent time epochs is reduced linearly over the remaining time epochs. Revised similarity percentages are used to generate a new set of scenarios at the time of the update. The crowdshipper availability and the set of commodities for the new scenarios are regenerated in the same way as with the initial scenario generation but according to the updated similarity percentages. Using the new set of scenarios, the BD method is solved for the period of time represented by the remaining τ time epochs, by fixing the variables associated with past epochs as described in Section 4.4.

Updating the system only once per day results in lower cost savings, with the highest improvements achieved at epoch $t = 4$. Figure 4.5a illustrates the total cost reduction as a

Figure 4.5: Cost reduction resulting from the PASSND with different updating policies when $|\mathcal{T}| = 16$, and $R = 8$



(a) The effect of update time epoch to total cost ($) for different policies

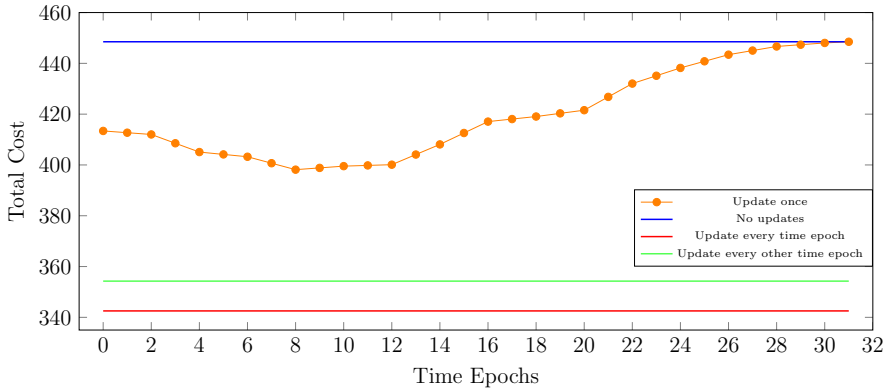


(b) The effect of update time epoch to cost savings percentage, for $|K| \in \{30, 100\}$

result of update policies over time, indicating that the cost savings are higher around the first half of the planning horizon, and the effect of updating diminishes as the update time gets closer to the end of the planning horizon. The update time can affect the total cost in two different ways. First, an update in an early time epoch benefits from a smaller amount of revealed information. On the other hand, an update in a later time epoch will have a lower effect as a smaller number of epochs would benefit from the updated plans. An update in an epoch later than $t = |\mathcal{T}| - R$ will have limited effect as all commodities have already arrived in the system and the updates can only improve the plans with respect to more accurate CS availability predictions — hence the significant drop in cost savings as of $t = 4$ in Figure 4.5b. The interaction of the above-mentioned phenomena explains the fact that the best cost savings among policies with a frequency of one were observed in $t = 4$.

To better illustrate the effect of plans updates, we also experiment with a setting with a finer time discretization and a shorter service commitment (i.e., $|\mathcal{T}| = 32$, and $R = 8$). For this setting, Figure 4.6 presents the total cost savings of different update policies over time. We observe that the savings increase until epoch $t = 8$, after which the effect of updates starts to diminish until the end of the operational period. The results suggest that the highest savings are obtained in the early stages of the operational period when a significant amount of information is revealed while still having plenty of time to adjust capacity accordingly.

Figure 4.6: Total cost comparison of PASSND policies ($|\mathcal{T}| = 32$)

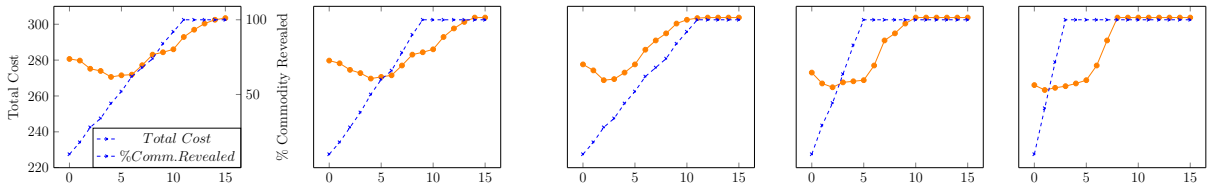


In a scenario where the system undergoes a single update during the operational period, the timing of the update can significantly impact potential cost savings. Figure 4.7 displays the total cost of the system as a single update takes place at various time epochs during the operational period. In these analyses, we assess the correlation between the level of cost reduction as a result of an update at a given time epoch and the percentage of commodities and their demands that are known by the decision-maker by the time of the update. Let T^{com} be the time epoch by which 100% of commodities are revealed, while commodities enter the system over the time interval $[0, T^{com}]$ uniformly. Figures 4.7a - 4.7e present the cases where T^{com} varies between 11 and 3, when $|\mathcal{T}| = 16$. For instance, in Figure 4.7a, the blue dashed line shows the cumulative percentage of commodities realized up to any particular time epoch, while the solid orange line represents the total cost of the system if an update takes place at that epoch. Upon examining the five different demand arrival patterns, we can observe that earlier demand arrivals lead to an earlier optimal time for updating the system. Moreover, in general, the optimal time for updating falls around the time when 30-45% of the demand over the operational period is revealed.

4.7 Discussion

In this paper, we studied the stochastic service network design problem for an intra-city courier service provider, focusing on efficiently managing delivery tasks with a hybrid fleet

Figure 4.7: Best update time epochs for different levels of demand arrival rates



(a) $T^{com} = 11$ (b) $T^{com} = 9$ (c) $T^{com} = 7$ (d) $T^{com} = 5$ (e) $T^{com} = 3$

comprising contracted drivers, crowdshippers, and third-party drivers. Our planning framework addresses uncertainty in demand and crowdshipper availability, optimizing capacity acquisition both in advance and on demand. At the tactical level, we acquire transportation capacity through the forward market taking into account future demand and crowdshipper availability estimations. At the operational level, we dynamically supplement capacity using spot market acquisitions and employing crowdshippers, leveraging flexibility without prior commitments. We formulate the problem as a two-stage stochastic integer programming model and develop a branch-and-Benders-cut with a partial Benders Decomposition approach to solve it. Our solution framework proposes a mechanism to integrate classical Benders Decomposition, integer L-shaped method, and Benders Dual Decomposition, leveraging different types of optimality cuts. To enhance efficiency, we employ acceleration techniques such as the subproblem scoring mechanism, parallelism, and ϵ-optimality.

Additionally, we introduced the concept of partially adaptive stochastic service network design. In this concept, once the tactical plans are made well before the start of the operational phase, the decision maker may be authorized to update the plans on limited occasions. We performed numerical analysis to identify the best time for an update in the plans and discovered that the benefits of an update would be maximized if it is done in the early stages of the operational phase, however, after a few time epochs, when some actual information about the system is revealed.

The main focus of our study was on tactical-level planning, which justifies our choice of solution method, despite its relatively high computational times. In a setting where tactical

plans are updated rarely, a relatively long computational time might not be a concern. However, in settings with much higher paces, the required computational time might be seen as one of the limitations of our proposed solution approach.

The future work will include generalizing the problem setting to better represent the real conditions by considering the hub limitations such as parking, sorting, and storage capacity in the city environment. Developing solution methodologies based on heuristic or metaheuristic approaches may be required to solve larger instances in a reasonable amount of time. Additionally, a heuristic algorithm fast enough to be used for updating the system during daily operation can be another future direction.

CHAPTER 5

SERVICE NETWORK DESIGN FOR EXPRESS INTRA-CITY COURIER SERVICES

5.1 Introduction

Efficient and effective design of service networks for intra-city package delivery is critical in the evolving landscape of urban logistics. This is especially important in addressing the increasing demands for express last-mile delivery services. Courier companies face many challenges in navigating complex urban environments, including competitive service commitments, limited transportation, and drop-off and pick-up locations with limited storage and handling capacities. In megacities with high population densities, a significant number of packages are shipped each day within and outside the city. This paper aims to address the challenges faced by major package services companies, such as FedEx, UPS, or SF Express, in providing cost-efficient, express intra-city delivery services within megacities. This includes the transport of shipments with both origin and destination in a target city or the first or last-mile delivery of inter-city shipments within the target city.

Major courier companies strategically position multiple facilities throughout large cities (e.g., UPS Stores or SF Express locker hubs). Such facilities, with a limited storage capacity and relatively basic sorting capability, function as both drop-off and pick-up points, serving the surrounding geographical regions. We refer to these facilities as hubs. Traditionally, such companies adopt a hierarchical system with a hub-and-spoke network structure. This structure necessitates the transfer of all shipments to centralized or local distribution centers

(DC) for sorting before redistribution to their final destination hubs. However, this organizational setup, often with DCs located on the city outskirts, may result in extended routes and increased time spent at terminals (56; 142). The proposed alternative in the literature eliminates the DCs by allowing direct communications among the hubs or indirect communications through other hubs of the network. In essence, the hubs serve as the origin and destination of the shipments, as well as the transfer and consolidation points in the network. That is, from its origin to its destination, a shipment may be handled at zero or more intermediate hubs. This setting has a strong resemblance to the service network design problem and the less-than-truckload problem in long-haul transportation settings (40). Intra-city delivery poses unique challenges due to the typically short service commitments and hub-related restrictions. These hubs are sometimes located in well-established city areas with limited available space. Consequently, there is restricted space for vehicle loading and unloading, as well as for temporary freight storage, often resulting in waiting times for available space.

The goal is to determine an efficient design of operations to flow the freight in the network between origin-destination (OD) hub pairs using an existing fleet of vehicles, given the capacity constraints at the hubs, while adhering to the offered service guarantees. Cost efficiency can be achieved by consolidating shipments to improve vehicle utilization. Consolidation may imply that shipments do not necessarily travel directly from origin to destination, but potentially pass through one or more intermediate hubs on the way to their destination. Consolidation may also happen over time by holding some shipments at a hub for some time, awaiting consolidation with other arriving shipments. Aggressive service commitments reduce the chances of performing high levels of consolidation.

To achieve the above-mentioned goal, two sets of interdependent decisions must be made: (1) how to flow the freight in the network to meet the service commitments, and (2) how to route the available fleet of vehicles to provide the transportation capacity required at specific times and locations in the network in order to support this flow. In this study, we explore two strategies. First, we develop an integer programming formulation that allows us to make

the above-mentioned decisions concurrently. Second, we propose a metaheuristic solution framework that relies on decomposing the problem into two subproblems, and repeatedly and sequentially solving them, intending to identify better solutions. Therefore, our main contributions in this paper are:

- We introduce, formalize, and model a new service network design problem inspired by the needs and activities of intra-city courier service providers. We explicitly account for practical constraints such as hub storage capacity and the number of vehicles being loaded/unloaded simultaneously at a hub.

- We propose an integer programming mathematical formulation (referred to as the comprehensive IP) for the problem based on a time-expanded network that aims to identify the least-cost solution incorporating freight movements and vehicle routes, such that on-time delivery of the shipments is guaranteed.

- We develop a metaheuristic framework that decomposes the problem into two subproblems, a freight routing problem (FRP) and a vehicle scheduling problem (VSP). Our metaheuristic is implemented as a multi-thread search, each conducted independently while sharing information through a central memory. The search procedure of each thread consists of using a matheuristic incorporating a memory-based adaptive large neighborhood search designed to deconstruct a solution combined with an IP-based repair mechanism to generate a solution to the FRP. The solution of the FRP is fed into the VSP, which generates vehicle routes using a unified tabu search equipped with efficient move evaluation techniques. At the master level, a simulated annealing framework governs the search of each thread in terms of acceptance/rejection of the newly generated solutions and the stopping conditions.

- We conduct an extensive computational experiment to systematically evaluate the performance of the proposed comprehensive IP and the metaheuristic approach. Further, we analyze the possibility of solving the FRP using an IP solver and propose enhancing

strategies such as design balance constraints. Our computational study shows that our proposed metaheuristic can generate solutions to smaller instances with comparable quality to those of the comprehensive IP in a shorter time and can generate "good" solutions to larger instances for which the IP solvers struggle to find feasible solutions or close the gap, in a reasonable amount of time. Therefore, our proposed framework constitutes a viable approach to be used in practical intra-city courier service settings.

The remainder of the paper is organized as follows. Section 5.2 provides a high-level survey of the relevant literature. Our problem alongside the notation used in the paper is formally described in Section 5.3. The proposed comprehensive IP formulation as well as the decomposed formulation are presented in Section 5.4. Next, we discuss our solution approach in Section 5.5. The computational study is presented in Section 5.6, and finally, Section 5.7 provides some concluding remarks and future work directives.

5.2 Literature Review

The service network design problem (SNDP) is a critical facet of transportation planning, situated at the tactical level of logistic operations. Initially introduced by (34), SNDP encompasses a spectrum of operational considerations within less-than-truckload carriers, spanning routing selections, terminal operations, service schedules, and the strategic repositioning of empty assets (34). In the context of multi-commodity service network design, the challenge entails identifying a set of paths and itineraries associated with commodities and a cost-efficient set of vehicle movements. Notable reviews of the SNDP have been conducted by (35; 163; 79; 39).

The SNDP exhibits deterministic and stochastic variants, based on the availability of information about the state of the system. Moreover, based on the treatment of the temporal dimension of the problem, static SNDP and time-dependent (scheduled) SNDP (SSNDP) are two separate variants. The SSNDP explicitly integrates temporal considerations by

representing demand and activities over time, in contrast to static SNDP which models the frequency of executed services.

Given the NP-hard nature of the mixed integer programming (MIP) formulation of the service network design problem, research efforts have predominantly focused on crafting heuristics and metaheuristics. The exploration of exact algorithms and the provision of lower bounds for SND are relatively less common in the literature. (4) and (103) propose a branch-and-price framework to solve SNDP with asset management and SNDP with heterogeneous resource constraints, respectively. In addition, (21) and (80) use the dynamic discretization discovery algorithm to find a continuous-time solution to the SNDP.

Various heuristic methods have been developed to address service network design problems. (74) introduce metaheuristics with cycle-based neighborhood structures, while (127) propose a two-phase tabu search metaheuristic for arc-based SND. (157) present a three-phase meta-heuristic combining path relinking and tabu search. (26) introduce a metaheuristic with cutting-plane and variable fixing procedures. (41) combine column generation and slope-scaling for SND with resource constraints. (102) propose a metaheuristic based on tabu search for SND with heterogeneous assets. (160) present a hybrid algorithm with pricing, cutting techniques, and local search, designed for large-scale SND scenarios with a heterogeneous fleet.

While significant literature exists on optimizing service networks for package express carriers, the prevailing focus has been on inter-city package flows rather than the nuanced dynamics of same-day delivery within a city. Contributions include works by (86; 62; 26; 157; 167). (167) point out that increasing the efficiency of intra-city operations is one of the factors that has the highest impact on company profits. Recently, attention to the intra-city express courier applications of SND has increased with examples such as (142; 56; 164; 80).

The same-day, intra-city delivery landscape presents unique challenges due to tight time constraints and the compact nature of hubs. (164) focus on the service network design problem within SF Express's intra-city same-day operations, considering hub capacity

constraints and limited loading/unloading capacity. They propose an IP model but find it impractical to solve directly, leading to the development of three heuristic approaches: an IP-based heuristic, a metaheuristic, and a hybrid metaheuristic. (80) propose an exact solution algorithm using dynamic discretization discovery, capable of solving smaller instances in one day. Our study differs in several aspects: (1) We incorporate loading, unloading, and parking spaces as shared resources with limitations, including storage space availability. (2) Our approach considers preferred and secondary physical paths for each commodity, penalizing the use of secondary paths. (3) We involve a company fleet generating routes for each vehicle, contrasting with independent vehicle movements considered in previous studies.

This paper shares similarities with (142) in terms of operating within a single-layer physical network framework, using a homogeneous fleet, and adhering to fixed service commitments. However, differences arise in the treatment of demand dynamics; while this study assigns arrival and latest delivery time attributes to commodities, (142) utilize a rate-based demand model. Additionally, this paper considers hub constraints and central parking location considerations, whereas (142) do not. Furthermore, while this study focuses on scheduled SNDP, (142) concentrate on frequency-based SNDP, highlighting the diverse aspects within the realm of vehicle routing and scheduling problems.

5.3 Problem Definition and Notation

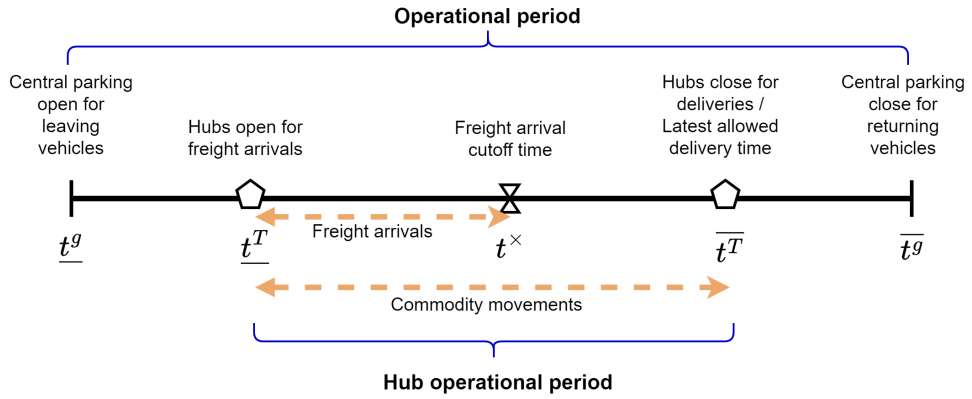
In the intra-city environment, the courier companies rely on their network in the city and a fleet of vehicles to perform pickup and delivery tasks while complying with the service commitments offered to customers. Let $\mathcal{G} = (\mathcal{I} \cup \{g\}, \mathcal{L})$ be the physical network, where $\mathcal{I}$ is the set of local package drop-off and pick-up stores or parcel lockers (referred to as *hubs* in this paper), g is the central parking location from which the vehicles start their routes and return at the end of the day, and $\mathcal{L}$ is the set of *arcs* between hubs defined according to the city road infrastructure. Every arc $(i, j) \in \mathcal{L}$, where i represents the *tail hub* and j represents the *head hub* of the arc, has a deterministic travel time $d_{(i,j)}$.

Each hub represents the potential origin or destination of some shipments. During operation hours, customers or some courier personnel (those who pick up and drop off packages from and to customer locations) may drop off one or more shipments at the hubs with specific destination hubs and a service commitment. The shipments are sorted locally in the hubs and delivered to their destination hubs within their time. Once the shipments are delivered to their destination hubs, they are picked up by the customers or courier personnel for final delivery to customer locations. Additionally, the inbound and outbound inter-city demand can be captured by gateway hubs. That is, a specific subset of hubs can represent collecting centers where shipments coming from or bound for outside the city aggregate. The focus of this research is on the movements between the hubs; the final deliveries from a given hub to the customer locations are not planned here. A hub $i \in \mathcal{I}$ is characterized by different factors such as its storage capacity s_i and vehicle parking capacity p_i. Hub storage capacity defines the volume of shipments that can be stored in the hub at any time waiting to be shipped at a later time. Hub parking capacity defines the number of parking spots available in each hub, limiting the number of vehicles that undergo the handling process in the hub at a given time.

Companies may choose to offer OD-dependent service commitments or alternatively offer a blanket service commitment for all intra-city shipments. For the sake of simplicity, we assume that all shipments share the same service commitment, R. Extending the formulations and solutions approach discussed in this paper to an OD-dependent service commitment is straightforward. We refer to a group of shipments sharing the same origin and destination hubs and the same due time at the destination as a *commodity*. Let $\mathcal{K}$ be the set of commodities, where commodity k corresponds to the group of shipments with origin hub o_k, destination hub d_k, the arrival time to the system e_k, and the latest allowed delivery time to their destination, $l_k = e_k + R$. The size of each commodity is denoted by q_k.

We focus on a deterministic planning problem associated with the activities taking place within an *operational period*, which could correspond to a workday, assuming full demand

Figure 5.1: Illustration of the operational period



information for the target operational period. As shown in Figure 5.1, the operational period is defined between times $\underline{t^g}$ and $\overline{t^g}$, i.e., the opening and closing times of the central parking (garage), g, respectively. Within the operational period, $\underline{t^T}$ and $\overline{t^T}$ denote the opening and the closing times of the hubs, where the time frame $[\underline{t^T}, \overline{t^T}]$ is called the *hub operational period*. Additionally, $t^\times$ represents the cutoff time after which no new shipments are accepted to be delivered by the end of the same operational period: $t^\times$ is set a time equal to or earlier than $\overline{t^T} - R$.

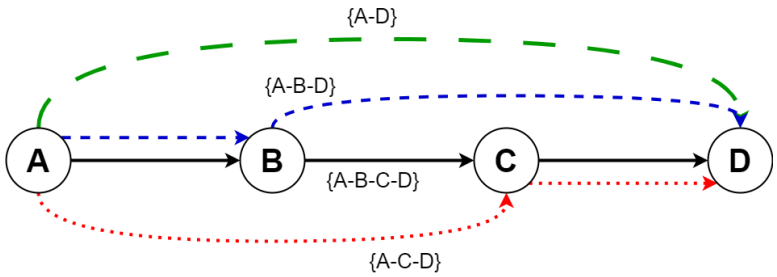
The required transportation capacity is provided by a fleet of homogeneous vehicles $\mathcal{V}$, with capacity Q. Each vehicle starts its route at g and returns to g at the end of the operations. A fixed cost c_v is incurred for each vehicle operated during a day, and a variable cost $c_{(i,j)}$ is incurred proportionally to the length of arc (i, j) on which a vehicle movement takes place. Vehicles are allowed to wait at the hubs until their next scheduled departures as long as there are enough parking spots available at the hubs. Upon arrival of a vehicle at a hub, it undergoes the handling process, i.e., loading and unloading processes. Each time a vehicle is loaded and unloaded at a hub i, it takes a total duration of d_i^H. We allocate a handling time $d_{(i,j)}^H$ to each arc (i, j) and set $d_{(i,j)}^H = d_i^H/2$ (loading) $+d_j^H/2$ (unloading).

Each commodity k is transported on a *path* from its origin o_k to its destination d_k. A path is a sequence of hubs connecting the origin hub to the destination hub. We assume for each OD pair, we are given one *preferred path* (p-path), and a set of *secondary paths*

(s-paths). For each commodity, the decision maker selects to either use a p-path or use one of the s-paths. Let the set $\mathcal{P}^k$ denote the set of paths for commodity k such that the first and last hubs of any path $p \in \mathcal{P}^k$ are o_k and d_k, respectively. Although paths are defined for each OD pair, different paths can be selected for commodities with the same OD pair but different arrival times. To encourage the utilization of the p-path rather than the s-paths, a penalty γ_p^P is incurred for selecting any path p, however, if p is a p-path, its penalty is set to zero. While the sequence of visits is prescribed by the path, based on the available flow the decision-maker may decide to skip the visit to an intermediate hub along the path. This results in a *path realization* of the original path offering some potential time savings (due to the triangular inequality, and the handling time d_i^H for visiting the intermediate hub i). For the sake of illustration, in Figure 5.2, different realizations of a generic path $(A - B - C - D)$ are shown, each with a different color and arrow style for each 4 different path realizations. For the path $(A - B - C - D)$, the path realizations are through arcs $\{(A, B),\ (B, C),\ (C, D)\}$, $\{(A, C),\ (C, D)\}$, $\{(A, B),\ (B, D)\}$, and $\{(A, D)\}$. Note that if a path realization of an s-path coincides with a realization of the p-path, then it is considered to be a path realization of the p-path. It is worth mentioning that a path or path realization is defined only in the physical network. Depending on the time flexibility of a path realization, several temporal realizations of it could be admissible. We refer to a temporal realization of a path (or path realization) as an *itinerary*, which specifies the scheduled departures of freight and movements along the selected path. A *vehicle route* denotes the movement of a vehicle in space and time during the operational period.

In the defined service network design problem, the decisions to make are selecting a path for each commodity, creating an itinerary based on the selected path, and constructing vehicle routes that provide sufficient capacity for those itineraries. The objective of the problem is to minimize the total fixed and variable cost, and the penalty incurred for path selection. In the next section, we introduce an IP-based formulation to create a set of commodity itineraries and a set of vehicle routes.

Figure 5.2: Illustration of path realizations

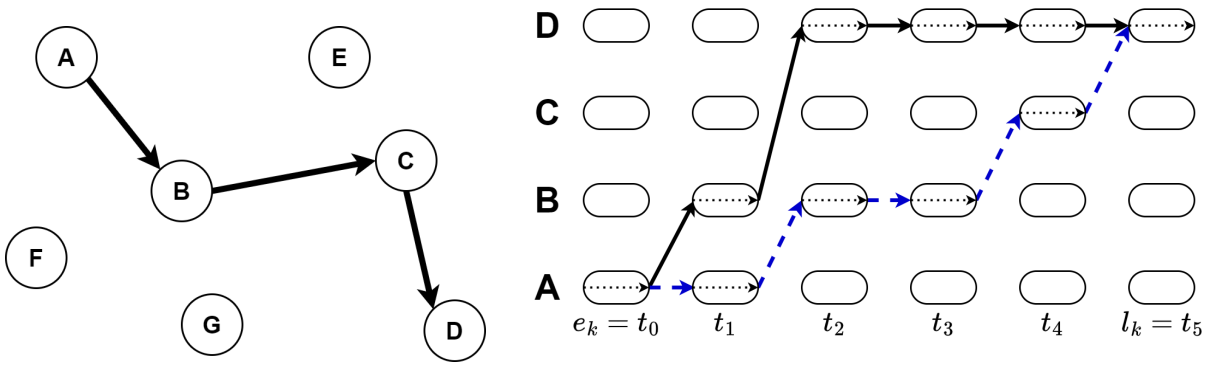


5.4 Intra-city Service Network Design Formulation

To efficiently manage the movements of freight and vehicles in space and time, we construct a time-space network. The operational period is divided into a set of shorter time periods $\mathcal{T} = \{0 = \underline{t^T}, 1, ..., T = \overline{t^T}\}$ of equal length κ. In the time-space network, each pair $(i, t),\ i \in \mathcal{I},\ t \in \mathcal{T}$ is represented by a *node*. Let $\mathcal{N} = \{\mathcal{I} \cup \{g\}\} \times \mathcal{T}$ denote the set of nodes. The time-space network is defined as $\mathcal{G}' = (\mathcal{N}, \mathcal{A})$. The set $\mathcal{A}$ denotes the union of all time-indexed arc sets. Each arc a is characterized by tuple $\langle[\underline{I(a)}, \underline{t^a}], [\overline{I(a)}, \overline{t^a}]\rangle$, the pair of tail hub and departure time of the arc, and the pair of head hub and arrival time of the arc, respectively. The arcs that connect two nodes of the same hub at consecutive periods are called *holding arcs* (and $\mathcal{W} \subset \mathcal{A}$ the set of such arcs), while the arcs connecting temporal copies of different hubs are called *movement arcs* ($\mathcal{M} \subset \mathcal{A}$), representing their function for vehicle or commodity flows. The travel time of the arc $a \in \mathcal{A}$ based on the road map, d_a is converted into a number of periods in the time-space network as $\overline{d_a} = \lceil \frac{d_a + d_a^H}{\kappa} \rceil$. In this definition, for the ease of computations, the handling time is added to the travel time of an arc. The variable cost associated with vehicle movement on arc a is c_a. Similarly to the handling time, the handling cost is also incorporated into the arc cost c_a.

Figure 5.3a depicts a sample path ($A-B-C-D$) within the physical network, connecting the OD pairs A and D. In Figure 5.3b, two distinct itineraries along this path are portrayed on the time-space network. Within the time-space network, each visit to a hub involves a handling process (black dashed arrows). These two itineraries are formed based on different

Figure 5.3: Illustration of the example time-space network



(a) Path A-B-C-D on the physical network (b) Two example itineraries on path A-B-C-D

path realizations. Specifically, the itinerary with solid arcs bypasses hub C (path realization: $A - B - D$), while the alternative itinerary includes a visit to every hub along the path.

A model that selects a path for each commodity, creates commodity itineraries and vehicle routes enforcing the capacity requirements of the vehicles and hubs can be formulated. Let the binary variable X_{av} get the value of 1, if vehicle v moves on arc a, 0 otherwise, and the binary variable Y_{pk} get the value of 1 if path $p \in \mathcal{P}^k$ is selected for commodity k, 0 otherwise. The binary variable Z_{ak} takes the value of 1 when commodity k moves on a movement arc $a \in \mathcal{M}$, 0 otherwise, while the binary variable W_{ak} is 1 if commodity k is held on a holding arc $a \in \mathcal{W}$, 0 otherwise. Let sets $\delta(i)^+$ and $\delta(i)^-$ denote the set of outgoing and incoming arcs of node i in the time-space network, respectively. Additionally, let binary parameter θ_{apk} be set to 1, if arc a can possibly be part of any realization of path $p \in \mathcal{P}^k$, 0 otherwise.

The objective function (5.1a) minimizes the total variable cost of vehicle movements on all arcs, the fixed cost of owning and operating a vehicle for used vehicles, and the penalty paid for path selection. Constraints (5.1b) enforce that exactly one path is selected for each commodity. Constraints (5.1c) enforce that commodities can only be assigned to the arcs allowed by their selected paths. Constraints (5.1d) enforce vehicle capacity limitations and guarantee that if an arc is assigned any flow, sufficient vehicle capacity must be provided. Constraints (5.1e) enforce the parking capacity limitations at the nodes: the number of vehicles that arrive at a hub at each time period should be less than the available parking

space at that hub. Constraints (5.1f) enforce the hub storage capacities on the holding arcs: The volume of shipment on a holding arc should be less than the capacity of the corresponding hub. Constraints (5.1g) enforce the flow conservation for each vehicle. That way, each vehicle will be assigned a route starting and ending at g. Constraints (5.1h) enforce the flow conservation for each commodity, building an itinerary starting from its origin node. Constraints (5.1i) to (5.1l) define the domain of the decision variables. For each commodity, this model selects a path, decides which path realization will be used, and creates an itinerary that ensures delivery without any delays. Notice that all shipments associated with a commodity travel along the same itinerary, however, they may be loaded on different vehicles. The model also creates vehicle routes for each vehicle taking into account the vehicle capacities, hub storage capacities, and hub parking capacities.

[Comprehensive IP]

$$\text{Min} \quad \sum_{v\in\mathcal{V}}\sum_{a\in\mathcal{A}} c_a X_{av} + \sum_{v\in\mathcal{V}} \sum_{\substack{a\in\mathcal{A}|\underline{I(a)}=g,\\ \overline{I(a)}\neq g}} c_v X_{av} + \sum_{k\in\mathcal{K}}\sum_{p\in\mathcal{P}^k} \gamma_p^P Y_{pk} \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}^k} Y_{pk} = 1, \qquad \forall k\in\mathcal{K} \tag{5.1b}$$

$$Z_{ak} \le \sum_{p\in\mathcal{P}^k} \theta_{apk} Y_{pk}, \qquad \forall a\in\mathcal{A}, p\in\mathcal{P}^k, k\in\mathcal{K} \tag{5.1c}$$

$$\sum_{k\in\mathcal{K}} q_k Z_{ak} \le \sum_{v\in\mathcal{V}} Q X_{av}, \qquad \forall a\in\mathcal{M} \tag{5.1d}$$

$$\sum_{v\in\mathcal{V}}\sum_{a\in\delta^-(i)} X_{av} \le p_i, \qquad \forall i\in\mathcal{N} \tag{5.1e}$$

$$\sum_{k\in\mathcal{K}} q_k W_{ak} \le s_{\underline{I(a)}}, \qquad \forall a\in\mathcal{W} \tag{5.1f}$$

$$\sum_{a\in\delta^-(i)} X_{av} - \sum_{a\in\delta^+(i)} X_{av} = \begin{cases} -1 & \text{if } j=g, t=0 \\ +1 & \text{if } j=g, t=T \\ 0 & \text{otherwise} \end{cases}, \qquad \forall v\in\mathcal{V}, i=(j,t)\in\mathcal{N} \tag{5.1g}$$

$$\Big(\sum_{\substack{a\in\delta^-(i)\\ a\in\mathcal{W}}} W_{ak} + \sum_{\substack{a\in\delta^-(i)\\ a\in\mathcal{M}}} Z_{ak}\Big) - \Big(\sum_{\substack{a\in\delta^+(i)\\ a\in\mathcal{W}}} W_{ak} + \sum_{\substack{a\in\delta^+(i)\\ a\in\mathcal{M}}} Z_{ak}\Big) =$$

$$\begin{cases} -1 & \text{if } j=o_k, t=e_k \\ +1 & \text{if } j=d_k \\ 0 & \text{otherwise} \end{cases}, \qquad \forall k\in\mathcal{K}, i=(j,t)\in\mathcal{N} \tag{5.1h}$$

$$X_{av}\in\{0,1\}, \qquad \forall a\in\mathcal{A}, v\in\mathcal{V} \tag{5.1i}$$

$$Y_{pk}\in\{0,1\}, \qquad \forall p\in\mathcal{P}^k, k\in\mathcal{K} \tag{5.1j}$$

$$W_{ak}\in\{0,1\}, \qquad \forall a\in\mathcal{W}, k\in\mathcal{K} \tag{5.1k}$$

$$Z_{ak}\in\{0,1\}, \qquad \forall a\in\mathcal{M}, k\in\mathcal{K} \tag{5.1l}$$

5.4.1 A Decomposition-based Formulation

Model 5.1 makes several decisions both for the freight and vehicles concurrently. Solving such a comprehensive model is computationally impractical for real-world instances. Alternatively, we propose an approach that decomposes the problem into two subproblems and solves them sequentially. In the first subproblem, referred to as the freight routing problem (FRP), we select a path for each commodity, based on the itineraries created. The output of the FRP is a set of vehicle movements on specific arcs of the time-space network generated without considering any vehicle routes. In the second subproblem, referred to as the vehicle scheduling problem (VSP), we construct a set of vehicle routes to carry out the vehicle movements of the FRP. This problem resembles the vehicle routing problem with time windows (VRPTW).

Freight Routing (Sub)Problem.

In this subproblem, we consider an arc-based formulation that selects a path and constructs an itinerary based on the selected path for each commodity from its origin to its destination, ensuring on-time delivery of all of the commodities. Let variables X_a denote the number of vehicles on arc $a \in \mathcal{A}$.Let parameter β_a^t get the value 1, if $\overline{t^a} \leq t$, and 0 otherwise. Also, let the integer variable B represent the lower bound on the number of required vehicles. The term γ^E represents the penalty for unused vehicle capacity.

$$\text{[FRP]} \quad \text{Min} \quad \sum_{a\in\mathcal{A}} c_a X_a + \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}^k} \gamma_p^P Y_{pk} + c_v B + \gamma^E \sum_{a\in\mathcal{M}} (QX_a - \sum_{k\in\mathcal{K}} q_k Z_{ak}) \tag{5.2a}$$

$$\text{s.t.} \quad (5.1b), (5.1c), (5.1h), (5.1j), (5.1k), (5.1l)$$

$$\sum_{k\in\mathcal{K}} q_k Z_{ak} \leq QX_a, \qquad \forall a \in \mathcal{M} \tag{5.2b}$$

$$\sum_{k\in\mathcal{K}} q_k W_{ak} \leq s_{\underline{I(a)}}, \qquad \forall a \in \mathcal{W} \tag{5.2c}$$

$$\sum_{\substack{a\in\delta^-(i)\\ a\in\mathcal{M}}} X_a \leq p_i \qquad \forall i \in \mathcal{N} \tag{5.2d}$$

$$\sum_{\substack{a\in\delta^+(i)\\ a\in\mathcal{M}}} X_a \leq p_i \qquad \forall i \in \mathcal{N} \tag{5.2e}$$

$$B \geq \sum_{\substack{a\in\mathcal{A}\\ \underline{t^a}\leq t}} (1-\beta_a^t) X_a, \qquad \forall t \in \mathcal{T} \tag{5.2f}$$

$$X_a, B \in \mathbb{Z}^+, \qquad \forall a \in \mathcal{A} \tag{5.2g}$$

Objective function (5.2a) minimizes the total variable cost of individual vehicle movements, the total penalty paid for path selections, an approximation (underestimator) of the fixed vehicle cost, and the total empty capacity penalty. Including the fixed vehicle cost on the lower bound of fleet size, and also the penalty for any empty capacity aims to create a more balanced set of vehicle movements that can be operated by a minimal-sized fleet in the VSP. Constraints (5.2d) and (5.2e) enforce parking capacity limitations at each node. It is essential to highlight a key distinction between Model 5.1 and Model 5.2 regarding the enforcement of parking capacities. In Model 5.1, parking capacities are enforced through a single set of constraints. This approach is justified by the fact that Model 5.1 tracks the routes of each vehicle. Conversely, Model 5.2 necessitates the imposition of two distinct sets of constraints. This difference stems from the fact that the vehicle movement entering a node and the vehicle movement exiting the same node are not guaranteed to be executed by the same vehicle. Thus, separate sets of constraints are needed to limit the number of vehicle

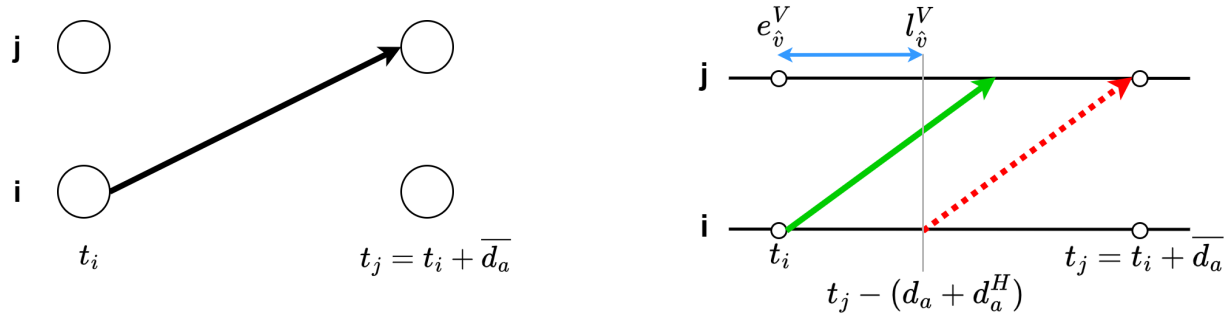
movements entering and leaving each node. Constraints (5.2f) determine the lower bound of the number of vehicle routes. In Constraints (5.2f), the lower bound is defined as the maximum number of active vehicle movements at any time period, calculated by counting the number of vehicle movements that have departed at or before a given time period but have not reached the head hub of their corresponding arcs yet, as $\max_{t \in \mathcal{T}}(\sum_{a \in \mathcal{A} | \underline{t^a} \leq t} (1 - \beta_a^t) X_a)$.

Vehicle Scheduling (Sub)Problem.

In this subproblem, we address the problem of creating the least-cost set of vehicle routes to execute the vehicle movements designed in the FRP. This problem can be modeled as a VRPTW, considering a network in which each identified vehicle movement in the FRP is represented by a vertex that needs to be covered. We formally define the VRPTW network for the VSP and formulate a MIP model that creates vehicle routes departing and ending at g and covering all vertices of the network. The VRPTW is defined on graph $\hat{\mathcal{G}} = \{\hat{\mathcal{V}}, \hat{\mathcal{A}}\}$ where $\hat{\mathcal{V}} = \{\hat{v}_0 = g, \hat{v}_1, ..., \hat{v}_n\}$ is the vertex set and $\hat{\mathcal{A}} = (\hat{v}_u, \hat{v}_w) : \hat{v}_u, \hat{v}_w \in \hat{\mathcal{V}}, u \neq w$ is the arc set. Each vertex $\hat{v}$ in $\{\hat{v}_1, ..., \hat{v}_n\}$ represents a vehicle movement with non-zero flow according to the solution of the FRP on a specific arc $a : \langle [\underline{I(a)}, \underline{t^a}], [\overline{I(a)}, \overline{t^a}] \rangle$ in the time-space network $\mathcal{G}'$.

Let vertex $\hat{v}$ represent a vehicle movement on arc $a : \langle [i := \underline{I(a)}, t_i := \underline{t^a}], [j := \overline{I(a)}, t_j := \overline{t^a}] \rangle \in \mathcal{A}$. Each vertex $\hat{v}$ in $\hat{\mathcal{V}} \setminus \hat{v}_0$ is associated with a service time equal to the travel time of the corresponding vehicle movement according to the physical network plus the handling time at the hubs of the arc: $d_a + d_a^H$. The time window to visit each vertex $\hat{v}$ in $\hat{\mathcal{V}} \setminus \hat{v}_0$ is denoted $[e_{\hat{v}}^V, l_{\hat{v}}^V]$. The flexibility of departure time comes from the fact that through the time discretization process, to obtain the travel time of an arc a in the time-space network, $\overline{d_a}$, the actual travel time according to the physical network, $d_a + d_a^H$ is rounded up to the smallest multiple of κ (the length of a period) that is greater than or equal to $d_a + d_a^H$. In Figure 5.4, the time window of an example vehicle movement is illustrated. Figure 5.4a shows a vehicle movement selected in the solution of the FRP. It is defined on

Figure 5.4: Illustration of the time window of a vertex in VRPTW



(a) A vehicle movement in discretized time-space network

(b) Time window of a vehicle movement in VRPTW

arc $a : \langle [i := \underline{I(a)}, t_i := \underline{t^a}], [j := \overline{I(a)}, t_j := \overline{t^a}] \rangle$ in the time-space network, arriving at hub j at time $t_i + \overline{d_a}$. In Figure 5.4b, however, the green arrow shows the actual arc duration, $d_a + d^H_a$. The time window $[e^V_{\hat{v}} = t_i, l^V_{\hat{v}} = t_j - (d_a + d^H_a)]$ represents the earliest and last time to depart from hub i to make it to hub j by t_j is shown.

Vertices in $\hat{\mathcal{G}}$ are connected by a set of arcs $\hat{a} : (\hat{v}_u, \hat{v}_w) \in \hat{\mathcal{A}} \subset \mathcal{L}$. The arcs represent the relocation of a vehicle from the head hub of vehicle movement $\hat{v}_u$ to the tail hub of the subsequent vehicle movement $\hat{v}_w$. Each arc $\hat{a} : (\hat{v}_u, \hat{v}_w)$ has a cost $\hat{c}_{uw}$, proportional to $d_{\hat{a}}$.

The objective of VRPTW on the defined network is to generate at most $|\mathcal{V}|$ feasible routes that minimize the total cost. Each vehicle route starts and ends at $\hat{v}_0 = g$. The duration of vehicle routes is limited by the length of the operational period $\overline{t^g} - \underline{t^g}$. The capacity constraints are taken care of in the FRP by not assigning flows larger than a vehicle capacity to each vehicle movement. Let binary variable E^v_{uw} get the value of 1 if vehicle v travels on

arc $(\hat{v}_u, \hat{v}_w)$, and let variable A_u denote the arrival time to vertex $\hat{v}_u$.

$$\text{[VSP]} \quad \text{Min} \quad \sum_{u\in\hat{\mathcal{V}}}\sum_{w\in\hat{\mathcal{V}}}\sum_{v\in\mathcal{V}} c_{uw}E^v_{uw} + \sum_{w\in\hat{\mathcal{V}}}\sum_{v\in\mathcal{V}} c_v E^v_{gw} \tag{5.3a}$$

$$\text{s.t.} \quad \sum_{v\in\mathcal{V}}\sum_{u\in\hat{\mathcal{V}}} E^v_{uw} = 1, \qquad \forall w \in \hat{\mathcal{V}} \tag{5.3b}$$

$$\sum_{w\in\hat{\mathcal{V}}} E^v_{gw} = 1, \qquad v \in \mathcal{V} \tag{5.3c}$$

$$\sum_{w\in\hat{\mathcal{V}}} E^v_{wu} - \sum_{w\in\hat{\mathcal{V}}} E^v_{uw} = 0, \qquad u \in \hat{\mathcal{V}}, v \in \mathcal{V} \tag{5.3d}$$

$$E^v_{uw}(A_u + d_{uw} - A_w) \leq 0, \qquad \forall u, w \in \hat{\mathcal{V}}, v \in \mathcal{V} \tag{5.3e}$$

$$e^V_u \leq A_u \leq l^V_u, \qquad \forall u \in \hat{\mathcal{V}} \tag{5.3f}$$

$$E^v_{uw} \in \{0, 1\}, \qquad \forall u, w \in \hat{\mathcal{V}}, v \in \mathcal{V} \tag{5.3g}$$

$$A_u \geq 0, \qquad \forall u \in \hat{\mathcal{V}} \tag{5.3h}$$

The objective function (5.3a) minimizes the total cost of vehicle empty movements and the fixed vehicle costs. Constraints (5.3b) ensure that all nodes are visited exactly once. Constraints (5.3c) guarantee that all vehicles leave and start their route at the central parking. Constraints (5.3d) enforce balance in vehicle flows. Constraints (5.3e) link the routes with arrival schedules. This constraint is non-linear but its linearization is trivial. Constraints (5.3f) ensure the time window constraints are respected.

Remark: Design balance constraints. Constraints (5.3d) ensure route continuity for the vehicles. This may result in adding empty vehicle movements to connect vehicle movements selected in the FRP to form a complete route. Anticipating these additions in the FRP would allow the model to potentially use the capacity of such additional vehicle movements. A variant of Constraints (5.3d), referred to as the design balance constraints (DBCs) can be added to the FRP. The DBCs aim to set the number of inbound and outbound vehicle movements into and out of every node of the time-space network equal to each other and

can be formulated as:

$$\sum_{a \in \delta^-(i)} X_a = \sum_{a \in \delta^+(i)} X_a, \qquad \forall i \in \mathcal{N} \tag{5.4}$$

These constraints are not originally part of the FRP, however, their addition could substantially improve the quality of the FRP, while simplifying the solution of the VSP. On the other hand, adding a relatively large number of constraints to the FRP can hinder its solution process, increasing the required computational time to solve the FRP. We evaluate this trade-off in our computational study.

Note that Model 5.2 with design balance constraints yields a 'balanced' set of vehicle movements at every node of the network. Therefore, the optimal solution to the VSP can be generated without additional vehicle movements, in case no restrictions on route duration are considered. In scenarios involving route length constraints, Model 5.2 may require additional vehicle movements to generate a feasible solution to the VSP.

5.5 Solution Approach

We propose a constructive metaheuristic framework that takes a complete initial solution incorporating commodity itineraries and vehicle routes and attempts to improve it. The proposed framework has multiple diversification and intensification mechanisms. In this section, we first establish the underlying structure of the metaheuristic and then provide detailed descriptions, intuition, and rationale behind each algorithmic component.

A multi-thread, memory-based search framework. We consider a search mechanism that explores the search space via $|\Xi|$ asynchronous parallel threads sharing a central memory Ψ. As a diversification technique, each thread is fed with a diverse initial solution, and therefore, explores a different neighborhood of the search space. As shown in Figure 5.5, each thread $\xi \in \Xi$, is equipped with a chamber Ψ_ξ in the central memory (i.e., $\Psi = \cup_{\xi \in \Xi} \Psi_\xi$), storing up to n^Ψ elite solutions in terms of the total cost found through the search in that thread. Thus, the central memory stores up to $|\Xi| n^\Psi$ high-quality and diverse complete

solutions. While each thread possesses its chamber in the central memory, different threads share information extracted from the solutions in their chambers with other threads, and in specific cases, they may feed a different thread with one of their elite solutions (see Section 5.5.1). The rationale behind a multi-thread design is related to the fact that the generation of a complete solution through solving the FRP and VSP is costly, and therefore, each thread will not be able to generate a large enough set of high-quality solutions to inspire the continuation of its search. A multi-thread implementation, in addition to exploring different regions of the search space in parallel, will quickly populate the central memory Ψ.

Search mechanism inside a thread. Each thread $\xi \in \Xi$ carries out an independent search, partially informed by the central memory, while simultaneously contributing to it by feeding it elite solutions. At the master level, the search within each thread ξ is governed by a simulated annealing (SA) framework, which provides accept/reject rules for newly generated solutions as well as a thread-specific stopping criterion. The search within each thread is initiated with a diverse initial complete solution. The initial solutions are generated using one of the construction heuristics detailed in Appendix C.1. As shown in Figure 5.6, in every iteration of the search within thread ξ, the neighborhood of the current solution is explored using an ALNS+TS approach. To that end, we first activate the ALNS mechanism to partially destroy the current complete solution by removing a subset of its vehicle movements, selected based on some destruction operator, and then repair the partial solution to become a valid solution to the FRP. Next, the solution of the FRP is converted to a complete solution by solving the VSP using a tabu search (TS) that constructs the vehicle routes. The new neighboring complete solution is either accepted or rejected based on the rules imposed by the SA. If it is accepted, it takes over the current solution and will be considered as a candidate to become a member of Ψ_ξ. Unless the stopping criterion is met, a new iteration of the ALNS+TS begins. In what follows, we describe each of these components in detail.

Figure 5.5: Illustration of the Multi-thread Structure

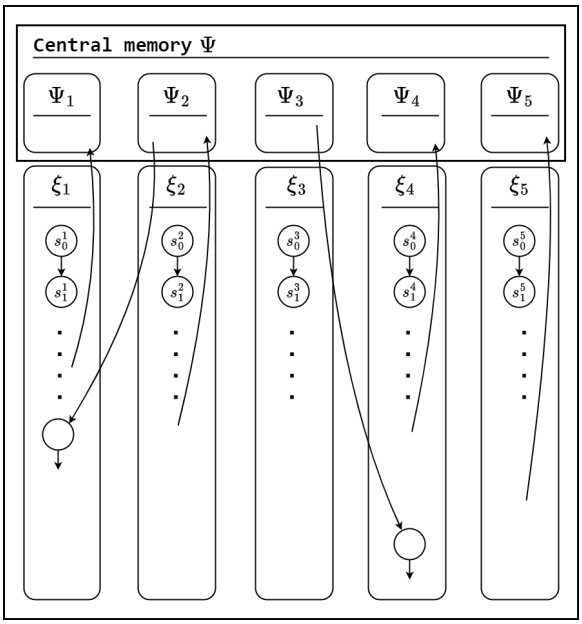
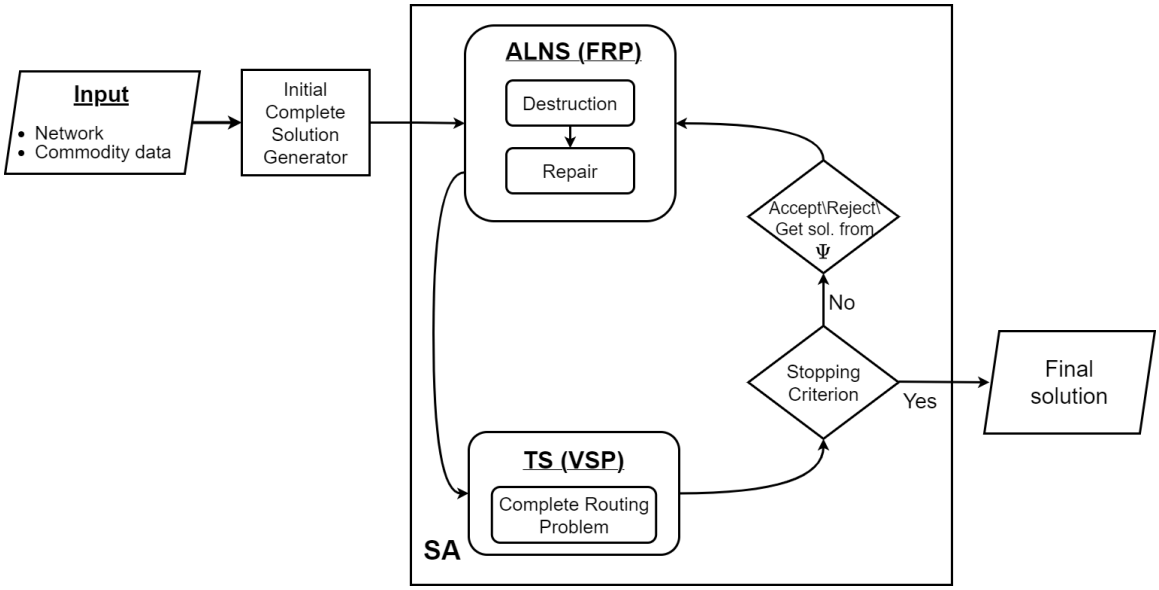


Figure 5.6: The Framework of Metaheuristic Inside Each Thread



5.5.1 Simulated Annealing

After completing each iteration of ALNS+TS, we subject the solution discovered to an acceptance criterion, guided by SA. This approach, as outlined by Kirkpatrick et al. (1983), Osman (1993), and Chiang and Russell (1999), serves as the search paradigm at the master level. SA strikes a balance between exploration and exploitation through its probabilistic acceptance of worse solutions. SA starts with an initial solution (see Figure 5.6) and iteratively explores the solution space by activating the ALNS+TS mechanism. It uses a temperature parameter that controls the level of randomness in the search process: the likelihood of accepting worse solutions. At higher temperatures, SA allows for more exploratory moves, while as the temperature decreases, the algorithm becomes more deterministic. The acceptance of solutions that are worse than the current one at higher temperatures allows the algorithm to escape local optima and explore different regions of the solution space (diversification). SA incorporates an annealing schedule to gradually reduce the temperature through a cooling procedure. As the temperature decreases, the algorithm becomes more focused on exploitation, converging towards the optimal solution. It intensifies the search by accepting only better solutions at lower temperatures, allowing the algorithm to refine the solution and converge toward the global optimum (intensification).

The new complete solution $s^{V'}$, obtained after solving the VSP (using TS), replaces the current solution s^V if $f(s^{V'}) < f(s^V)$, where $f(s^V)$ denotes the value of complete solution s^V. If $\Delta f = f(s^{V'}) - f(s^V) > 0$, the acceptance of solution $s^{V'}$ occurs with a probability governed by the formula: $exp(\frac{-\Delta f}{T})$, where $T > 0$ is the temperature parameter. The probability of accepting s' decreases as T decreases. The cooling procedure is initiated when there has not been a discovery of a thread-specific best feasible solution in the past ϕ iterations. If over the past δ^{rep} rounds of cooling procedures (i.e., $\delta^{rep}\phi$ iterations) no new solution is generated to replace the current solution of a given thread, the current solution of the thread is replaced by a solution randomly selected from a chamber of the central memory other than the one associated with that thread (see Figure 5.5).

5.5.2 A memory-based ALNS for FRP

The classical ALNS algorithm, introduced by (Ropke and Pisinger, 2006; Pisinger and Ropke, 2007), operates as an iterative procedure in which, during each iteration, a portion of the existing solution is deliberately destroyed and then rebuilt to discover an improved solution. Each solution s^F to the FRP consists of the assignment of commodities to paths, determination of the path realizations, and itineraries.

While our ALNS algorithm draws inspiration from the overarching ALNS concept, it integrates several customized enhancements to boost its efficiency. Additionally, we introduce new specialized operators tailored to address the unique characteristics of our specific problem.

Search Space. The search space associated with the proposed ALNS coincides with the search space of the FRP, i.e., the space of vehicle movements. Therefore, the destruction of a solution would involve the removal of a subset of the vehicle movements, and repairing a partial solution would insert a minimal set of vehicle movements to reconstruct a complete solution for the FRP.

During each iteration, we explore the neighborhood of the current solution, by selecting a destruction operator *opr* from the set of operators Ω matched with our IP-based repair operator. A destruction operator *opr* is selected at each iteration based on a random mechanism called the roulette wheel, favoring the operators that have been successful in recent iterations according to certain criteria.

Adaptive search engine. We use the solutions in the central memory Ψ for scoring the operators leading to new solutions. We implement an adaptive weight adjustment procedure to represent the historic performance of the operators and use these weights to bias their selection at each iteration. A weight ω_{opr} is thus assigned to each operator *opr*. Initially, all the weights are set to one. We update the operator weights after each iteration, based on the performance history of the operators. The probability of selecting *opr* is then defined as $\omega_{opr} / \sum_{o \in \Omega} \omega_o$. The performance of the operators is captured through a scoring mechanism. A

score is assigned to each operator, with the score being initially set to zero. At each iteration, we then update the scores by adding a bonus factor μ_i, where $i \in \{1, \ldots, 3\}$, as follows:

(I) μ_1 if the new solution satisfies the acceptance criterion and is inserted into the memory chamber of the corresponding thread, Ψ_ξ,

(II) μ_2 if the new solution improves the current solution but not the thread-specific best feasible solution;

(III) μ_3 if a new thread-specific best feasible solution has been found.

The bonus factor is zero in all other cases. Let π_{opr} be the total score of *opr* obtained from ν_{opr} applications of *opr* so far. We update the weight of each operator using a parameter $\zeta \in [0, 1]$, called the reaction factor, through the formula:

$$\omega_{opr,i+1} \quad = \quad \omega_{opr,i}\,(1-\zeta) + \zeta\frac{\pi_{opr}}{\nu_{opr}} \tag{5.5}$$

where $\omega_{opr,i}$ represents the weight of operator *opr* in the ith iteration.

Destruction operators. We develop several new destruction operators, each removing different sections of an FRP solution. The set of operators listed here is a subset of the best-performing ones from a larger set of operators that we created and tested through a procedure similar to the one laid out in Section 5.6.4. Depending on the topology and structural properties of the solution to be partially destroyed, we group our destruction operators into three categories: (1) independent vehicle-movement removals, (2) itinerary-based vehicle-movement removals, and (3) route-based vehicle-movement removals.

1. **Independent vehicle-movement removals** This type of removal consists of selecting a subset of independent vehicle movements of the current solution based on certain characteristics. Each time a vehicle movement is identified to be removed, the itineraries associated with the eliminated vehicle movements are also discarded from the current

solution, while preserving the path assignment decisions. The number of vehicle movements removed is determined by parameter λ^{VM}, which indicates the percentage of existing vehicle movements to be eliminated. The process of selecting vehicle movements for removal is made based on two criteria, each forming a destruction operator:

(a) **Random selection:** The set of vehicle movements to be removed is sampled randomly from the set of all vehicle movements in the current solution. Each vehicle movement holds a probability of being selected for removal inversely proportional to the number of times it appeared in the elite solutions stored in the central memory Ψ, favoring the removal of vehicle movements that have never or less frequently been part of an elite solution. This operator seeks to introduce randomness to diversify the exploration of neighboring solutions while simultaneously learning from previously explored solutions.

(b) **Maximum empty-miles:** The initial step in this selection criterion involves sorting all vehicle movements in the current solution based on a descending order of an empty-miles metric. Subsequently, the selection is made from the top of this sorted list. The empty miles metric for an arc a is calculated as follows: Let $Q_a = QX_a$ and $F_a = \sum_{k \in \mathcal{K}} q_k Z_{ak}$ represent the total capacity and total flow on arc a, according to the current FRP solution. The empty-miles metric is calculated as $\Gamma_a = \frac{Q_a - F_a}{Q} d_a$, putting an emphasis both on the empty capacity and length of a vehicle movement. This operator aims to discard low-utilization vehicle movement hoping that their flows will be captured by some existing vehicle movements, resulting in cost savings.

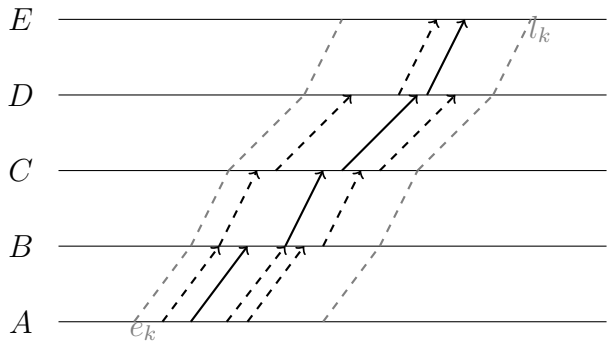
2. **Itinerary-based vehicle-movement removals** This form of vehicle-movement removal focuses on selecting a set of commodities and removing the set of vehicle movements associated with their assigned itineraries as well as relaxing their path selection decisions.

Additionally, the itineraries of any other commodities that utilize the removed vehicle movements are also eliminated, while retaining their path assignments. The parameter λ^C controls the percentage of commodities for which the itineraries are destroyed. The process of selecting commodities for removal is made based on two criteria, each forming a destruction operator:

(a) **Maximum slack time:** For each commodity itinerary, we define the slack time as the latest allowed delivery time of the commodity minus the arrival time to the destination hub according to the current itinerary. The commodities are sorted in descending order according to their slack times, and the top λ^C itineraries with the maximum slack time are selected. Then, their path and itinerary assignments, along with the set of vehicle movements associated with the itineraries, are removed. Additionally, the itineraries of any other commodities utilizing the removed vehicle movements are also eliminated, while preserving their path assignments. By identifying commodities with the highest flexibility for complete itinerary reconfiguration, this operator aims to enhance consolidation on the non-removed vehicle movements.

(b) **Maximum total empty-miles (Corridor-based):** Let $\mathcal{A}^k$ denote the set of vehicle movements that form the itinerary of commodity k. Initially, the commodity itineraries are sorted based on their total empty miles $\sum_{a \in \mathcal{A}^k} \Gamma_a$. Based on this metric, a set of commodity itineraries, determined by the parameter λ^C, are selected. Their path and itinerary assignments are removed from the solution. Moreover, for each chosen commodity, a time corridor is established on the arc set $\mathcal{A}^k$ by considering the earliest and latest possible departure times on each arc of the path realization of the itinerary. In Figure 5.7, we illustrate the example of a commodity itinerary (bold arrows) associated with a commodity path ($A - B - C - D - E$). Given this path, one can identify a time window $[a_{\text{earliest}}, a_{\text{latest}}]$

Figure 5.7: Corridor-based Vehicle Movement Removal



for $a \in \mathcal{A}^k$, indicating any departure time along arc a that can possibly be part of a time feasible itinerary along the considered commodity path. These time intervals are shown using the dashed gray line in Figure 5.7, creating a corridor evolving around the current path of the selected commodity. Consequently, for all selected itineraries according to total empty miles metric, any vehicle movements falling within the corridor $[a_{\text{earliest}}, a_{\text{latest}}]$ for $a \in \mathcal{A}^k$ (dashed-line arrows) are also removed from the solution. Note that the set of itineraries not selected by the total empty-miles metric but associated with the removed vehicle movements are disrupted; their itineraries are not entirely removed, and their partial itineraries are retained. The intuition behind this operator is that by identifying commodity itineraries flowing on low-utilization vehicle movements and removing them and any vehicle movement that can possibly serve those itineraries, one may be able to re-route those movements more efficiently.

3. **Route-based vehicle-movement removals** This type of removal consists of removing the vehicle movements associated with a subset of the vehicle routes of the current complete solution, s^V. The commodity itineraries affected by the removal of the eliminated vehicle movements are also destroyed while preserving their assigned paths. The parameter λ^{VR} controls the percentage of vehicle routes to be eliminated from

the solution. The process of selecting vehicle routes for removal is made based on two criteria, each forming a destruction operator:

(a) **Maximum total empty miles:** Let $\mathcal{A}^r$ denote the set of arcs of vehicle route r in the current solution. The routes are sorted based on the sum of empty miles of the arcs in $\mathcal{A}^r$, i.e., $\sum_{a \in \mathcal{A}^r} \Gamma_a$. Then, the vehicle movements of the λ^{VR} percent of the routes with the highest total empty miles are destroyed. This operator aims to potentially decrease the number of vehicle routes by eliminating those with low utilization vehicle movements.

(b) **Ratio of route length with high empty miles:** Let $\hat{\mathcal{A}}^r$ be the set of arcs characterized by high empty miles, defined as $\Gamma_a \geq \epsilon$. After identifying $\hat{\mathcal{A}}^r$, the routes are sorted based on the metric capturing the percentage of route length with high empty miles, given by $\frac{\sum_{a \in \hat{\mathcal{A}}^r} d_a}{\sum_{a \in \mathcal{A}^r} d_a}$. Subsequently, the vehicle movements of the λ^{VR} percent of the routes with the highest ratio of route length with high empty miles are removed. Through this operator, our objective is to identify vehicle routes characterized by a higher proportion of low-utilization legs. Removal of such routes has a higher potential in reducing the required fleet size.

Repair operator. Following the destruction of the current FRP solution using one of the destruction operators, the partial solution undergoes a repair procedure to construct a valid FRP solution. To that end, we develop an IP formulation, Model 5.6, that is constructed and solved following each destruction procedure, allowing us to obtain a complete FRP solution, while ensuring adherence to both vehicle and hub capacity constraints.

Let $\mathcal{I}', \mathcal{K}', \mathcal{P}^{k'}$, and $\mathcal{A}'$ denote the set of hubs, commodities whose itineraries are destructed, the potential paths for such commodities, and the subset of arcs associated with the potential paths of the commodities in $\mathcal{K}'$, respectively. $\mathcal{W}'$ is the set of holding arcs as a subset of $\mathcal{A}'$. A time-space network is defined as $\tilde{\mathcal{G}} = (\mathcal{N}', \mathcal{A}')$ where $\mathcal{N}' = \{\mathcal{I}' \cup \{g\}\} \times \mathcal{T}$. It should be noted that $\mathcal{P}^{k'} \subseteq \mathcal{P}^k$, depending on the destruction operator. Let x_a be the number of existing

(non-destructed) vehicle movements, and F_a be the existing flow on $a \in \mathcal{A}'$, according to the partial solution at hand. Binary parameter θ_{apk} takes the value of 1 if commodity k can be assigned to the arc a based on its available paths in $\mathcal{P}^{k'}$. Similarly to Model 5.2, let the integer variable X_a denote the number of vehicle movements that are added to the partial solution, and let binary variable Y_{pk} denote the selection of path p for commodity k.

$$[\texttt{Repair}] \quad \text{Min} \quad \sum_{a\in\mathcal{A}'} c_a X_a + \sum_{k\in\mathcal{K}'}\sum_{p\in\mathcal{P}^{k'}} \gamma_p^P Y_{pk} + c_v B + \gamma^E \sum_{a\in\mathcal{I}} (QX_a - \sum_{k\in\mathcal{K}} q_k Z_{ak}) \tag{5.6a}$$

$$\text{s.t.} \quad (5.1b), (5.1c), (5.1h)$$

$$F_a + \sum_{kin\mathcal{K}'} q_k Z_{ak} \le Q(X_a + x_a), \qquad \forall a \in \mathcal{M}' \tag{5.6b}$$

$$F_a + \sum_{kin\mathcal{K}} q_k Z_{ak} \le s_{\underline{I(a)}}, \qquad \forall a \in \mathcal{W}' \tag{5.6c}$$

$$\sum_{ain\delta^-(i)} (X_a + x_a) \le p_i \qquad \forall i \in \mathcal{N}' \tag{5.6d}$$

$$\sum_{ain\delta^+(i)} (X_a + x_a) \le p_i \qquad \forall i \in \mathcal{N}' \tag{5.6e}$$

$$B \ge \sum_{\substack{a\in\mathcal{A} \\ \underline{t^a} \le t}} (1 - \beta_a^t)(X_a + x_a), \qquad \forall t \in \mathcal{T} \tag{5.6f}$$

$$X_a, B \in \mathbb{Z}^+, \qquad \forall a \in \mathcal{A}' \tag{5.6g}$$

$$Y_{pk} \in \{0,1\}, \qquad \forall p \in \mathcal{P}^{k'}, k \in \mathcal{K}' \tag{5.6h}$$

$$Z_{ak} \in \{0,1\}, \qquad \forall a \in \mathcal{A}', k \in \mathcal{K}'. \tag{5.6i}$$

Objective function (5.6a) minimizes the total variable cost of individual vehicle movements that are added to the destructed solution, the total penalty paid for path selections, an approximation (underestimator) of the fixed vehicle cost, and the total empty capacity penalty. Constraints (5.6b) ensure that the sum of vehicle capacity added during the repair process and the existing capacity on each arc is sufficient to cover the total flow on arc a.

Constraints (5.6c) to (5.6e) ensure that hub storage and parking capacities are respected. Constraints (5.6f) determine the lower bound of the number of vehicle movements.

5.5.3 A Tabu Search for the VSP

The VRPTW is a highly complex NP-hard combinatorial optimization problem that has been extensively studied in the literature. Heuristic and metaheuristic approaches are developed for a wide variety of problem characteristics since exact methods are not able to address most large-size instances. The hybrid genetic algorithm of (120; 156; 122; 168), the iterated local search of (85; 115; 22), the ALNS of (128; 100; 68; 125), and the Unified Tabu Search (UTS) of (28; 30; 29; 31; 69; 121; 166; 1) are several powerful solution approaches in the literature.

We note that the VSP is heavily constrained by the set of vehicle movements selected by the FRP. Therefore, solving the VSP may not necessitate a sophisticated algorithm involving a large neighborhood search. We develop a TS algorithm to address the VRPTW associated with the VSP, given its known strength despite its simple logic. Introduced by (76), TS is a metaheuristic local neighborhood search method mainly used in combinatorial optimization. (23; 60) provide an excellent literature survey on TS applications on VRPTW, together with other metaheuristics.

Our TS framework. The TS technique systematically explores the solution space through iterative steps, advancing from a given routing (VSP) solution s^V to the best solution within its defined neighborhood, denoted as $N(s^V)$. In contrast to conventional descent methods, the current solution is permitted to degrade between iterations. Acceptance of new, less favorable solutions is restricted to instances where it facilitates the exploration of unexplored regions. This strategy ensures the exploration of novel regions in the solution space, aiming to navigate around local minima and ultimately converge towards the optimal solution. To avoid repetitive cycles, solutions with attributes similar to recently examined solutions are temporarily marked as tabu or restricted. The duration of an attribute's tabu status, known as its *tabu tenure*, may vary across different time intervals. Under specific conditions, the tabu

status can be overridden, a concept referred to as the *aspiration criterion*. This criterion may come into play when a tabu solution demonstrates superiority over all previously encountered solutions. Various strategies are commonly employed to either diversify or intensify the search process, thereby enhancing the overall effectiveness of the TS technique.

Let $\mathcal{S}^V$ represent the set of solutions to the VSP, compliant with the route duration constraint, i.e., start after $\underline{t^g}$ and return to g by $\overline{t^g}$. Each routing solution $s^V \in \mathcal{S}^V$ is characterized by a set of vehicle routes, ensuring that each vehicle movement $\hat{v} \in \hat{\mathcal{V}}$ is served by exactly one vehicle. The time window constraints are relaxed, and instead of excluding infeasible solutions, they are penalized. A *time warp* tw_u incurs a cost for late arrival at a vertex u, calculated as $tw_u = A_u - l_u^V$, where A_u is the arrival time at vertex u. This approach is based on the relaxation method introduced by (120), allowing violations of time window constraints but using a penalized time warp to extend beyond the time window. Early arrivals are permitted, resulting in a waiting time $w_u = e_u^V - A_u$. The TS algorithm iteratively transitions from a current solution s^V to another solution $s^{V'} \in N(s^V)$ based on move evaluation criteria. Each routing solution s^V is associated with a utility function $C(s^V) = D(s^V) + \eta TW(s^V)$, where $D(s^V)$ and $TW(s^V)$ denote the total transport cost (represented by (5.3a)) and the total usage of time warp, respectively. Consequently, each *move* is assessed based on the change it induces in the utility function of the new solution. The parameter η is dynamically adjusted throughout the search, initially set to 1. After every set of $Iter^{adj}$ iterations, the value of η is doubled if the count of solutions with nonzero time warp in the most recent $Iter^{his}$ iterations exceeds δ_{max}. Conversely, it is halved if the count of such solutions falls below δ_{min}.

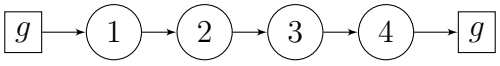
Each solution $s^V \in \mathcal{S}^V$ is associated with an attribute set $B(s^V) = (i, v)$, where (i, v) signifies that vertex i is served by vehicle v. The neighborhood $N(s^V)$ of a solution s is determined by the simple act of removing an attribute (i, v) from $B(s^V)$ and substituting it with an attribute (i, v'), where $v \neq v'$. When a vertex i is removed from the route of vehicle v, the reconnection of the route involves establishing links between the preceding

and succeeding vertices of the removed vertex. Subsequently, the vertex i is inserted into a new route v' between two successive vertices that minimize the utility function $C(s^V)$. After removing customer i from route v, preventing its reinsertion into the same route for the subsequent α iterations is achieved by labeling the attribute (i, v) as tabu. We set the tabu tenure for a newly executed move by drawing a random number from the predefined interval $[minTabu, maxTabu]$. The tabu status of an attribute can be lifted through an *aspiration criterion* if it facilitates the search process in finding a solution with a lower cost than the best solution found with that attribute.

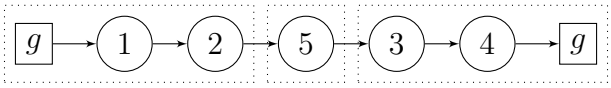
Move evaluation. As a computationally efficient method for performing move evaluations, for each move (removal and insertion), vehicle routes are divided into *partial routes* σ, which are then concatenated (155). For each partial path σ, the minimum duration $D(\sigma)$ to perform services, the minimum time warp usage $TW(\sigma)$, and the earliest $E(\sigma)$ and latest $L(\sigma)$ visit to the first vertex that allows for a schedule with the minimum duration and time-warp use are calculated. In a vehicle route with a single tour, the removal of a vertex can be represented by concatenation of at most 2 partial routes, and each insertion of a vertex can be represented by concatenation of at most 3 partial routes. Figure 5.8a provides an example of a vehicle route with 4 visited vertices. To update the information of the route following the insertion of vertex 5 between vertices 2 and 3, the partial paths $\sigma_1 - \sigma_3$ are concatenated. These 3 partial paths are illustrated in Figure 5.8b. Also, as in Figure 5.8c, 2 partial paths are concatenated in the case of the removal of vertex 5 from its position between vertex 2 and 3.

The calculation of the information of the routes by the concatenation operation $\oplus$ can be performed by the following Equations 5.7:

Figure 5.8: Concatenations to update route information after an insertion and a removal

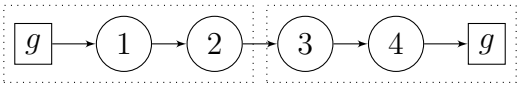


(a) Initial route



σ^1 σ^2 σ^3

(b) Route after inserting node 5



σ^1 σ^2

(c) Route after removing node 5

$$D(\sigma_1 \oplus \sigma_2) = D(\sigma_1) + D(\sigma_2) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + \Delta_{WT} \tag{5.7a}$$

$$TW(\sigma_1 \oplus \sigma_2) = TW(\sigma_1) + TW(\sigma_2) + \Delta_{TW} \tag{5.7b}$$

$$E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_2) - \Delta, E(\sigma_1)\} - \Delta_{WT} \tag{5.7c}$$

$$L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_2) - \Delta, L(\sigma_1)\} + \Delta_{TW} \tag{5.7d}$$

where $$\Delta = D(\sigma_1) - TW(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \tag{5.7e}$$

$$\Delta_{WT} = \max\{E(\sigma_2) - \Delta - L(\sigma_1), 0\} \tag{5.7f}$$

$$\Delta_{TW} = \max\{E(\sigma_1) + \Delta - L(\sigma_2), 0\} \tag{5.7g}$$

5.6 Computational Study

We conduct a comprehensive computational study to evaluate the performance of our proposed solution approaches. Our computational study is organized as follows. In Section 5.6.1, we introduce the set of test problems considered. We present our design of experiments

in Section 5.6.2, and parameter tuning and algorithm calibration are discussed in Section 5.6.3. We evaluate the results of the computational experiments for different variants of the proposed solution approach as well as a comparative analysis of the solutions approaches in terms of the fleet sizes and the number of vehicle movements in Section 5.6.4. In Section 5.6.4, we assess the effectiveness and contribution of each of the proposed destruction operators for the ALNS. Finally, in Section 5.6.4, we analyze the effect of hub constraints.

The tests were run in Python 3.6.5 on a machine with a 2×2.4 GHz Intel Xeon E5-2640 v4 processor and 512 GB of memory. The models were solved using Gurobi 9.0.3 with a maximum runtime of 24 hours. The progress of the performance of the models after different computational times 4, 8, 12, 16, 20 are reported.

5.6.1 Chicago Case Study Data from (142)

We test the proposed solution methods on a set of instances that reflect the characteristics of the considered problem. In this study, as in (142), the physical network of hubs is constructed with the 48 FedEx drop-off and pick-up stores in the city of Chicago, and a FedEx Ground DC defined as the central parking location. The Google API service is employed to obtain real road distances between hubs, assuming an approximate vehicle speed of 30 km/h within city limits, regardless of the time of day, to estimate travel times.

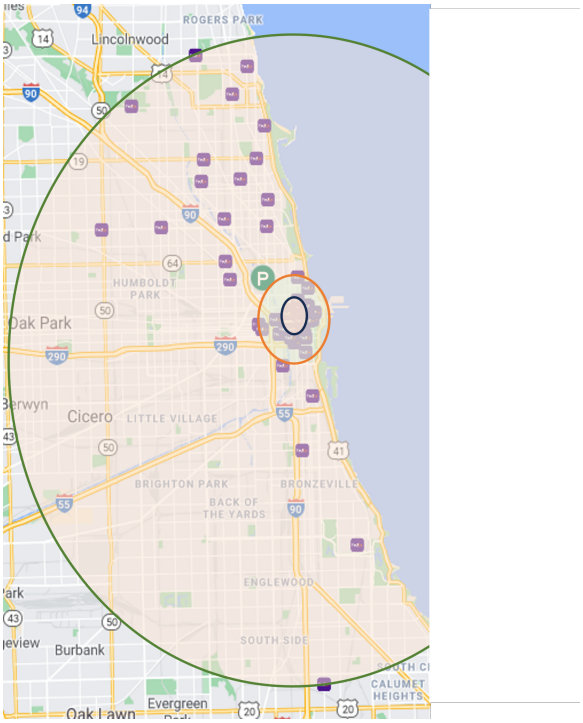
5.6.2 Design of Experiments

The computational experiments are designed based on the values considered for different attributes of the problem as given in Table 5.1. We consider an operational period of 12 hours with 8 hours of hub operational period. The time horizon is discretized into periods of 15 minutes.

For simplicity, the handling time is considered to be 0.2 hours for all hubs, and the service commitment, R, is 4 hours for all commodities. Vehicle capacity is fixed at 250 shipments, while hub parking capacity and hub storage capacity are determined for each hub based on the following perimeter rule: The hubs are separated into three equal-sized groups based on their distance to the gravity center of the hub network as in 5.9. From the low distance to

the longer distance from the gravity center of the hub network, the storage capacities and the parking capacities increase according to 1, 2, 3 and 250, 500, 750, respectively.

Figure 5.9: Illustration of the perimeter rule.



In (142), pairwise demand rates (number of shipments received per unit of time) served as the basis for generating demand instances. In contrast, our paper incorporates the temporal aspect of demand arrival. Thus, we sample $|\mathcal{K}|$ instances of type (i, j, t) $\forall i, j \in \mathcal{I}, t \in \mathcal{T}$: $t \leq t^{\times}$, where (i, j, t) denotes the origin, destination, and arrival time of a commodity. The volume associated with each instance is then sampled from a Poisson distribution with mean demand rates obtained from (142). For every instance class (combination of the number of hubs and the number of commodities), we generate five instances and report the average values in the results.

A fixed handling time of 0.2 hours is added to the travel time of each arc to incorporate handling activities upon arrival and before the departure of a vehicle from a hub. The handling time is treated as constant for all hubs, encompassing the duration for paperwork, parking, loading and unloading, and other activities. The path penalty γ_p^P for a path p is 0

Table 5.1: Parameter configurations considered in the design of experiments

Attribute	Description	Level Values
$\lvert\mathcal{I}\rvert$	Number of hubs	48
$\lvert\mathcal{K}\rvert$	Number of commodities	$\{25, 50, 100, 150, 200, 250, 300, 400, 500\}$
R	Service commitment	4 hrs
$[\underline{t^g} - \overline{t^g}]$	Operational period length	12 hrs
$[\underline{t^T} - \overline{t^T}]$	Hub operational period length	8 hrs
κ	Time granularity	0.25 hrs
Q	Vehicle capacity	250
c_v	Fixed cost per vehicle	\$574.32
c_a/d_a	Variable cost per mile	\$0.665
p_i	Hub parking capacity	$\{1, 2, 3\}$
s_i	Hub storage capacity	$\{250, 500, 750\}$
d_i^H	Handling time at hub i	0.2 hrs

if p is a p-path, otherwise, it is considered five times the difference between the variable cost of the p-path and the variable cost of the s-path, assuming that all hubs are visited.

5.6.3 Parameter Tuning

In line with the typical behavior observed in most metaheuristics, alterations in parameter values have the potential to impact the algorithm's performance without compromising its correctness. We set the initial temperature of the SA to ($T^{\text{init}} = 0.05\frac{f(s_0^V)}{\ln(0.5)}$), where ($f(s_0^V)$) represents the value of the initial routing solution. This choice of values is inspired by the tuning methodology conducted by (128). The final temperature is set to ($T^{\text{fin}} = T^{\text{init}}c^5$), ensuring a minimum of 15 iterations. In this application, the fine-tuning of parameters is accomplished through a trial-and-error approach, guided by comprehensive preliminary tests. An overview of the parameter values used in our SA and ALNS implementation is given in Table 5.2.

5.6.4 Experimental Study

In this section, we analyze the results of the computational experiments. Each instance of the problem is addressed by five solution approaches below:

Table 5.2: Parameter values used in ALNS application

Parameter	Description	Level Values
	ALNS	
n^{Ψ}	Number of best solutions in memory	5
c	Cooling rate for SA	0.9987
ϕ	Number of iterations for cooling	3
φ	Score update number of iterations	1
δ^{rep}	Number of cooling w/o new solutions to replace current solution	2
μ_1, μ_2, μ_3	Bonus factors for adaptive weight adjustment	1,1,2
λ^{VM}	Ratio of the independent vehicle movements to be removed	%20
λ^{C}	Ratio of the commodity itineraries to be removed	%15
λ^{VR}	Ratio of the vehicle routes to be removed	%15
ϵ	High empty miles threshold	0.25
	TS	
$Iter^{his}$	Number of past iterations to consider for adjusting η	100
$Iter^{adj}$	Number of infeasible solutions in the past $Iter^{his}$ to adjust η	30
δ_{max}	Upper threshold to increase η	5000
δ_{min}	Lower threshold to decrease η	1
$[minTabu, maxTabu]$	Tabu tenure random number interval	$[3, 7]$

1. **Comprehensive IP:** Solving the comprehensive IP associated with Model (5.1) with a commercial solver;

2. **Decomposition-based approach without the DBCs:** Addressing the FRP through solving Model (5.2) with Gurobi, while generating vehicle routes for VSP using the proposed TS;

3. **Decomposition-based approach with the DBCs:** Addressing the FRP through solving Model (5.2) augmented with DBCs (5.4) with Gurobi, while generation vehicle routes for VSP using the proposed TS;

4. **Decomposition-based approach with single-thread search:** Sequentially running five single-thread searches, without memory sharing mechanism;

5. **Decomposition-based approach with multi-thread search:** Running a five-thread search in parallel mode with the memory sharing mechanism.

Performance of solution approaches

In Table 5.3, we present a comprehensive comparison of the performance of the five solution approaches across a wide range of instance sizes. Each reported value represents the average

outcome over five instances ($I = 5$) within its respective class. Detailed instance-level results are reported in Tables C.2-C.5 in Appendix C.2. The total computational times are reported in hours. The column "Gap from BKS" reports the gap between the best upper bound associated with a given solution approach and the best-known solution obtained using any of the approaches. In the case of the decomposition-based approach with single-thread search, "Gap from BKS" is calculated based on the best solution obtained in the five runs of the metaheuristic. In the case of approaches that rely on solving an IP for the FRP, Gurobi may run out of memory, in which case we label the instance with "mem" or it may fail to find any feasible solution within the allotted time limit of 24 hours, in which case we label them as "nfs".

Analyzing the results in Table 5.3 indicates that the comprehensive IP when solved with Gurobi is not capable of generating any feasible solutions in the larger instances within the 24-hour time limit, and ends up with large optimality gaps for the smaller instances. We also observe that the decomposition-based approaches generally outperform the comprehensive IP in terms of generating high-quality solutions in a relatively shorter amount of time.

Next, we focus on the effect of DBCs. Interestingly, for small and medium-sized instances, the addition of DBCs significantly improves the speed of the solution process. Compared to the comprehensive IP, the DBC-mounted FRP is much simpler as it does not explicitly schedule vehicle routes. A DBC-mounted FRP is, however, much heavier than the FRP without DBCs in terms of the number of constraints, hence a larger computational time for the implementations with DBCs compared to those without. However, such additional computational time pays off by generating optimal solutions for instances with 25-200 commodities. Notice that the solution of the DBC-mounted FRP already guarantees vehicle movement continuity in the network; an extremely simplistic and fast procedure can generate a set of routes based on the vehicle movements of the DBC-mounted FRP in almost no time. For larger instances (with ≥ 250 commodities), the situation is different. In those instances, the DBC-mounted FRP often hits the time limit with an optimality gap ranging from 1% to

51%, depending on the size of the instance. As for the FRP without DBCs, the solution of the FRP + TS takes between 0.5 and 9 hours depending on the size of the instance. These times, compared to the computational times of the DBC-mounted FRP, show the added complexity of the DBCs. The decomposition without DBCs is, however, more vulnerable to decomposition-related suboptimality. Notice that in the decomposition with DBCs, most decisions are made in the FRP, and very little is left to the VSP, while in the decomposition without DBCs, decisions are more equally divided between the two subproblems, making the procedure more vulnerable to suboptimality.

Next, we focus on the performance of our metaheuristic. To better evaluate the contributions of different components of the multi-thread search, we also implemented a procedure in which the same number of single-thread searches are conducted separately. Compared to the multi-thread search, each single-thread search has access only to the information extracted from its own elite solutions. The threads in a multi-thread search, however, have access to all the elite solutions generated by all the threads as well as the opportunity to replace the current solution with an elite solution from another thread, in the case of a stagnated search.

Notice that, although each iteration of the ALNS+TS is relatively fast since multiple iterations of those two are run through the SA framework, the total runtimes appear longer. For the computational times to be comparable, in the case of the single- and multi-threads, we report the average computational times per thread. We observe that compared to the approaches where the FRP was solved using Gurobi (decomposition with and without DBC), the combination of the computational time and solution quality has improved. More specifically, the metaheuristics outperform the decomposition without DBC in terms of solution quality while requiring more computational effort. Compared to the decomposition with DBC, the metaheuristics provide an inferior solution quality, however, in a fraction of the time required for the decomposition with DBC.

Comparison of the single- and multi-thread metaheuristics reveals the benefit of collaborative search among the threads through sharing their elite solutions and the information extracted

from them to rank vehicle movements and to score destruction operators. With comparable computational times, the superiority of the multi-thread over a single-thread search is clear with an average improvement of the gap from BKS by 1% across all instances, and 2% in large instances (400 and 500 commodities).

An interesting observation from the metaheuristic results is that the gaps improve as the instances grow in terms of the number of commodities. Notice that the underlying network of all these instances is the same (the same 48 hubs), and therefore, an increase in the number of commodities translates into a larger flow density across the network and time. The latter increases the opportunities for consolidation and reduces the effect of any deviation from the optimal solution.

To evaluate the performance of the solution approaches under different runtime limits, in Table 5.5, we report the best solution approaches if we allow $\{4,\ 8,\ 12,\ 16,\ 20,\ 24\}$ hours of solution times for each instance. In general, we observe that the Multi-Thread solution approach produces the best solutions, especially with smaller allowed solution times and larger instances. In the meantime, the Decomposition w/DBC approach outperforms the other approaches for smaller instances with longer allowed solution times. The total costs and gaps from the best-known solution are reported in the Appendix C.2, in Table C.6.

Analysis of fleet size

In Table 5.4, we present the fleet sizes and the number of vehicle movements for all solution approaches. Fleet sizes indicate the number of vehicle routes, while vehicle movements are categorized into three columns: the total number of vehicle movements, the number of empty vehicle movements, and the percentage of empty vehicle movements relative to the total number.

An interesting observation emerges: as the demand density in the network (represented by the number of commodities) increases, the percentage of empty vehicle movements tends to decrease. However, this trend exhibits exceptions in the largest instances solved using the

decomposition approach with design balance constraints, where high optimality gaps lead to suboptimal vehicle routes. Moreover, the elimination of design balance constraints tends to elevate the percentage of empty vehicle movements.

The next insight is gained by observing the correlation between the percentage of empty vehicle movements and the total cost. This shows the value of approaches that more smartly distribute the capacity in the network to prevent or reduce the unused capacity of the vehicles. This also shows that the main vulnerability of a decomposed version is related to a loose connection between freight routing and vehicle routing, potentially resulting in a large number of empty vehicle movements. The DBCs, although computationally expensive, can effectively create a tight connection between the decisions made in the two subproblems.

Our metaheuristic on the other hand, by reiterating through the construction of a complete solution and partially destructing it using vehicle utilization-driven operators, develops a learning that allows it to achieve a percentage of empty vehicle movement that is comparable to the optimal solutions.

Evaluation of destruction operators

In this section, we conduct an assessment of the individual destruction operators. For these analyses, we use the multi-thread search and focus on the five instances with 100 commodities only. This is because we have access to the optimal solutions for these instances through the decomposition with DBCs. In Table 5.6, for each destruction operator, we report the deterioration in the gap from BKS of an implementation of the multi-thread search when that destruction operator is disabled. This analysis shows the contribution of each destruction operator. We report the average gap for the five instances in column $Avg\%$. Notably, the independent vehicle-movement removal with maximum empty miles followed by the route-based vehicle-movement removal with the maximum total empty miles emerge as the top-contributing operators. The comparison of the gaps shows that all these operators have comparable contributions.

Evaluation of the effect of hub capacity constraints

In this section, we study the effect of considering the hub capacity constraints in the problem. We relax the constraints related to hub parking, hub storage, and both hub parking and storage, subsequently assessing the total costs in each of these scenarios using the decomposition-based approach with multi-thread search. Table 5.7 reports the decrease in the total cost (reported as a percentage) if each of the hub capacity constraints (parking and/or hub storage capacity) is relaxed. For each instance class, we also report the percentage of the hub parking and storage capacity constraints that are active at the final solution. The results show that as the freight density increases (larger number of commodities), a larger number of capacity constraints become active, driving the costs up. In our experiments, the parking capacity showed to be more often active (71.2% vs 2.9% active constraints) and to have a larger effect in the total cost (6.9% vs 0.5% cost decrease) when applied to the largest size instances.

In Table 5.7, we present the average utilization of parking or storage spaces under scenarios where only parking or storage capacity is enforced, and when both parking and storage capacities are enforced. We note a substantial increase in parking space utilization from 1.8% to 85.9% as network demand density rises. Additionally, the introduction of storage capacity has minimal impact on parking utilization, as parking remains the more restrictive constraint. However, storage space utilization is significantly influenced by parking constraints, as tighter parking restrictions result in more packages waiting at storage for later dispatch.

Table 5.3: Performance of the solution approaches w.r.t. the best known solution (BKS) for each instance

	Comp. IP			**Dec. (w/ DBC)**			**Dec. (w/o DBC)**		**Single-Thread**		**Multi-Thread**		**BKS**	
$\|\mathcal{K}\|$	**Run time (hrs)**	**Opt. gap**	**Gap from BKS**	**Run time (hrs)**	**Opt. gap**	**Gap from BKS**	**Run time (hrs)**	**Gap from BKS**	**Run time (hrs)**	**Gap from BKS**	**Run time (hrs)**	**Gap from BKS**	**Method**	**Cost**
25	24	3%	2%	7	0%	0%	0.5	17%	1.0	9%	1.0	7%	Dec. (w/ DBC)	$8,973
50	24	16%	11%	10	0%	0%	1.1	15%	3.0	6%	3.2	5%	Dec. (w/ DBC)	$14,069
100	24	21%	10%	14	0%	0%	1.7	19%	4.2	7%	4.4	7%	Dec. (w/ DBC)	$21,685
150	24	49%	23%	14	0%	0%	2.3	11%	5.2	4%	5.6	4%	Dec. (w/ DBC)	$18,221
200	24	nfs	-	22	0%	0%	2.5	10%	5.9	3%	6.3	2%	Dec. (w/ DBC)	$30,032
250	24	nfs	-	24	1%	0%	3.5	9%	7.6	2%	8.2	2%	Dec. (w/ DBC)	$37,810
300	-	mem	-	24	13%	1%	4.3	9%	11.4	0%	11.5	0%	Multi-Thread	$39,025
400	-	mem	-	24	51%	22%	5.5	7%	16.0	2%	16.2	0%	Multi-Thread	$44,620
500	-	mem	-	-	nfs	-	9.0	12%	21.4	2%	21.1	0%	Multi-Thread	$59,493

Table 5.4: Comparison of fleet sizes and vehicle movements (total vs. empty) across various solution approaches

	Comprehensive IP				**Decomposition (w/ DBC)**				**Decomposition (w/o DBC)**				**Single-Thread**				**Multi-Thread**			
	Fleet	**# of veh. movements**			**Fleet**	**# of veh. movements**			**Fleet**	**# of veh. movements**			**Fleet**	**# of veh. movements**			**Fleet**	**# of veh. movements**		
$\|\mathcal{K}\|$	**Size**	**Total**	**Empty**	**Empty %**	**Size**	**Total**	**Empty**	**Empty %**	**Size**	**Total**	**Empty**	**Empty %**	**Size**	**Total**	**Empty**	**Empty %**	**Size**	**Total**	**Empty**	**Empty %**
25	13	44	6	14%	13	45	7	15%	15	107	34	32%	13	51	8	15%	13	51	7	14%
50	23	108	11	10%	20	108	13	12%	24	211	49	23%	20	115	15	13%	20	106	13	12%
100	31	205	21	10%	32	224	20	9%	36	290	66	23%	32	219	25	11%	32	212	26	12%
150	38	320	25	8%	35	299	26	9%	39	317	68	22%	35	286	31	11%	35	280	29	10%
200					39	304	24	8%	47	389	66	17%	39	356	31	9%	39	348	35	10%
250					48	377	24	6%	59	504	62	12%	50	426	29	7%	50	411	31	8%
300					58	453	43	9%	63	580	72	12%	58	495	34	7%	58	485	37	8%
400					72	581	62	11%	68	559	57	10%	64	523	39	7%	63	512	38	7%
500									82	658	63	10%	73	698	44	6%	72	649	37	6%

Table 5.5: Best solution approach at different solution time limits (hrs)

	Best solution approach at solution times (hrs)					
$\|\mathcal{K}\|$	**4 hrs**	**8 hrs**	**12 hrs**	**16 hrs**	**20 hrs**	**24 hrs**
25	Multi-Thread	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)
50	Multi-Thread	Multi-Thread	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)
100	Multi-Thread	Multi-Thread	Multi-Thread	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)
150	Multi-Thread	Multi-Thread	Multi-Thread	Dec. (w/ DBC)	Dec. (w/ DBC)	Dec. (w/ DBC)
200	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Dec. (w/ DBC)	Dec. (w/ DBC)
250	Dec. (w/o DBC)	Single-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Dec. (w/ DBC)
300	Single-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread
400	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread
500	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread	Multi-Thread

Table 5.6: Performance comparison of destruction operators in multi-thread search, highlighting the increase in gap from BKS when disabled, for 5 instances of $|\mathcal{K}| = 100$ (The best performer at each instance in bold)

Destruction Operators		Demand Instances					Avg %
		1	**2**	**3**	**4**	**5**	
Independent vehicle-movement removals	**Random selection**	**2.1%**	0.9%	0.7%	1.2%	0.8%	1.1%
	Max. empty-miles	1.5%	**2.4%**	1.6%	**2.3%**	1.2%	1.8%
Commodity itinerary-based vehicle-movement removals	**Max. slack time**	0.8%	1.4%	0.8%	1.4%	**1.2%**	1.1%
	Max. total empty-miles (corridor-based)	0.7%	0.6%	0.6%	1.2%	1.2%	0.9%
Route-based vehicle-movement removals	**Max. total empty miles**	1.9%	1.4%	**2.2%**	2.1%	0.9%	1.7%
	Ratio of route length with high empty miles	1.4%	0.8%	0.9%	1.5%	0.7%	1.1%

Table 5.7: The impact of relaxing hub capacity constraints on total costs and utilization rates

$\|\mathcal{K}\|$	**Storage %**	**Parking %**	**No Constr.%**	**Parking %**	**Storage %**	**Parking**	**Parking & Storage**	**Storage**	**Parking & Storage**
25	0.1%	0.0%	0.1%	1.2%	0.0%	1.8%	1.9%	1.2%	3.1%
50	0.1%	0.0%	0.1%	3.6%	0.0%	4.6%	4.8%	1.9%	3.9%
100	0.3%	0.0%	0.3%	5.5%	0.0%	7.1%	7.4%	5.1%	16.1%
150	1.1%	0.0%	1.1%	13.8%	0.2%	18.9%	19.6%	13.4%	35.0%
200	3.2%	0.1%	3.2%	23.2%	0.9%	27.9%	27.8%	23.0%	45.0%
250	3.4%	0.1%	3.4%	27.5%	1.0%	39.5%	41.0%	38.0%	51.0%
300	5.1%	0.2%	5.1%	50.5%	1.3%	60.0%	62.5%	41.5%	59.0%
400	6.8%	0.4%	6.8%	64.1%	2.1%	71.5%	73.6%	49.5%	63.5%
500	6.9%	0.5%	6.9%	71.2%	2.9%	85.9%	86.9%	51.4%	69.5%

5.7 Conclusion

In this paper, we addressed the critical challenges faced by major courier companies in the design of efficient service networks for intra-city express delivery, specifically focusing on the complexities introduced by hub capacity constraints. The objective was to devise an efficient transportation plan for the movement of freight within the network, utilizing an existing fleet of vehicles. Our study introduced and formalized a novel service network

design problem, explicitly incorporating practical constraints such as hub storage capacity and limitations on the number of vehicles being loaded or unloaded simultaneously at a hub. To tackle this complex problem, we proposed two modeling approaches: an integer programming formulation (comprehensive IP) that concurrently addresses freight routing and vehicle scheduling decisions, and a decomposition-based modeling that decomposes the problem into a freight routing problem (FRP) and a vehicle scheduling problem (VSP).

The comprehensive IP can be solved for smaller instances using a commercial solver, providing a foundation to identify the least cost solution considering both freight movements and vehicle routes, ensuring on-time delivery of shipments. For larger instances, we developed a metaheuristic approach leveraging a multi-threaded search strategy, incorporating a matheuristic with an adaptive large neighborhood search for the FRP and a tabu search for the VSP. The metaheuristic demonstrated its efficiency by generating solutions to smaller instances with comparable quality to the comprehensive IP in a shorter time and also exhibited the capability to produce "good" solutions for larger instances where the IP solver struggled either close the optimality gap within the time limit or even generate a feasible solution.

Our extensive computational study not only validated the quality and viability of our proposed approaches but also explored the use of IP solvers to solve the FRP and introduced enhancement strategies such as design balance constraints. The results underscored the practical applicability of our metaheuristic framework in addressing the intra-city express delivery challenges faced by courier service providers, offering a feasible and efficient alternative to traditional hierarchical network structures.

In conclusion, our contributions in modeling, formulation, and designing solution strategies provide a significant step forward in the optimization of intra-city express delivery service networks. Future work could explore additional refinements to the metaheuristic framework and consider dynamic elements to further enhance its adaptability to evolving urban logistics scenarios. The continuous development and integration of advanced optimization techniques

are imperative for meeting the increasing demands of modern urban logistics and express delivery services.

CHAPTER 6

CONCLUSION

This dissertation introduced a novel network design for intra-city courier services, focused on improving the operational efficiency and service quality for courier companies. Unlike traditional hierarchical network models which mandate that all shipments be aggregated at centralized distribution centers for sorting—thus creating bottlenecks and delays—this study proposed an alternative that utilized existing infrastructure of courier package drop-off and pick-up stores scattered throughout urban areas. This alternative network design reorganized these stores as mini sorting hubs, enabling the decentralization of the sorting process. The research was presented in three separate articles, each addressing different facets of the proposed network design under various conditions of demand, capacity, and operational constraints.

In the first article, we focused on tactical and operational planning of the proposed network structure. Tactical planning relied on high-level aggregated demand rates over long periods and took the form of a multi-commodity service network design. The goal was to identify one path per commodity while maximizing consolidation opportunities in the network. Commodities were transported on their paths by means of a series of continuously operating vehicle cycles, where the structure and number of such cycles were determined concurrently with commodity path assignment decisions in a mixed integer programming framework. A second model was designed to refine the time allocation along different steps of a commodity path allowing a potential reduction in the number of cycles required to meet the service guarantee. Operational planning then narrowed down the focus to a shorter

time period, and the baseline plan obtained from the tactical planning phase was adjusted to better fit potential deviations in observed demand patterns compared to the aggregate patterns. Through an extensive computational study designed on the topology of a major US city, we observed that the plans designed at the tactical level guaranteed high service levels, which were further improved at the operational level by customizing the plan to the special characteristics of a day of operation.

In the second article, we explored the stochastic service network design problem of an intra-city courier service provider. To efficiently fulfill delivery tasks, the courier company employed a hybrid fleet consisting of contracted drivers, crowdshippers, and third-party drivers within a planning framework that took into account uncertainty in terms of demand and transportation capacity offered by crowdshippers. At the tactical level, taking into account future demand and crowdshipper availability estimations, the courier company acquired transportation capacity through the forward market at a relatively low rate. At the operational level, however, once the demand and the available crowdshipper capacities were revealed, the courier company could supplement the existing transportation capacity by acquiring through the spot market and/or by employing crowdshippers. We modeled the problem at the tactical level as a two-stage stochastic problem with integer variables in both stages and developed a Branch-and-Benders-Cut with partial Benders Decomposition approach (PBBC) to solve the model. In our solution framework, we incorporated classical Benders Decomposition, Integer L-Shaped Method, and Benders Dual Decomposition to generate different types of optimality cuts. To improve the efficiency of our method, we employed accelerating strategies such as selective subproblems, parallel computing, and ϵ-optimality. Further, to study the effect of possible plan revisions, we proposed a partially adaptive stochastic programming approach that allowed for a limited number of adjustments to the tactical level plans given the extra information revealed at the operational phase. We quantified the benefits of such updates and evaluated the effect of the frequencies at which such updates were performed.

In the third article, we studied a service network design problem in the context of an intra-city express delivery system with several hub capacity constraint types. We explicitly accounted for the restrictions on the number of vehicles that could simultaneously load or unload, and the freight storage capacity at a hub. We modeled the problem on a time-space network and formulated it as a comprehensive integer program that could be solved using a commercial solver for smaller-size instances. To be able to address large instances, we developed a constructive metaheuristic that relied on decomposing the problem into two subproblems: (1) a freight routing problem (FRP), and (2) a vehicle scheduling problem (VSP), which were then solved sequentially and iteratively to generate and improve a complete solution. To address the FRP, we designed a matheuristic incorporating a memory-based adaptive large neighborhood search, while the VSP was tackled using a unified tabu search with a smart move evaluation mechanism. Our metaheuristic was capable of generating "high-quality" solutions for instances with 500 commodities. The results of our extensive computational study advocated for the quality and viability of the proposed approach for real-world applications with diverse geographies, market sizes, and service offerings.

REFERENCES

[1] Zakir Hussain Ahmed and Majid Yousefikhoshbakht. An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows. *Alexandria Engineering Journal*, 64:349–363, 2023.

[2] Aliaa Alnaggar, Fatma Gzara, and James H Bookbinder. Crowdsourced delivery: A review of platforms and academic literature. *Omega*, 98:102139, 2021.

[3] Sibel A Alumur, Bahar Y Kara, and Oya E Karasan. Multimodal hub location and hub network design. *Omega*, 40(6):927–939, 2012.

[4] J. Andersen, M. Christiansen, T. G. Crainic, and R. Grønhaug. Branch and price for service network design with asset management constraint. *Transportation Science*, 45(1):33–49, 2011.

[5] J. Andersen, T. G. Crainic, and M. Christiansen. Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C*, 17(2):197–207, 2009.

[6] Jardar Andersen, Teodor Gabriel Crainic, and Marielle Christiansen. Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies*, 17(2):197–207, 2009.

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

[8] Claudia Archetti, Martin Savelsbergh, and M Grazia Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016.

[9] A.P. Armacost, C. Barnhart, and K.A. Ware. Composite variable formulations for express shipment service network design. *Transportation Science*, 36(1):1–20, 2002.

[10] A.P. Armacost, C. Barnhart, K.A. Ware, and A.M. Wilson. Ups optimizes its air network. *INFORMS Journal on Applied Analytics*, 34(1):15–25, 2004.

[11] Alp M Arslan, Niels Agatz, Leo Kroon, and Rob Zuidwijk. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.

[12] Ruibin Bai, Stein W Wallace, Jingpeng Li, and Alain Yee-Loong Chong. Stochastic service network design with rerouting. *Transportation Research Part B: Methodological*, 60:50–65, 2014.

[13] Anantaram Balakrishnan, Thomas L Magnanti, and Richard T Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

[14] C. Barnhart, N. Krishnan, D. Kim, and K. Ware. Network design for express shipment delivery. *Computational Optimization and Applications*, 21(3):239–262, 2002.

[15] C. Barnhart and R. R. Schneur. Air network design for express shipment service. *Operations Research*, 44(6):852–863, 1996.

[16] Beste Basciftci, Shabbir Ahmed, and Nagi Gebraeel. Adaptive two-stage stochastic programming with an application to capacity expansion planning. *arXiv preprint arXiv:1906.03513*, 2019.

[17] Adam Behrendt, Martin Savelsbergh, and He Wang. Crowdsourced same-day delivery: Joint planning and coordination for centralized and decentralized couriers. *Available at Optimization-Online: https://optimization-online. org*, 2022.

[18] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[19] Silvio Binato, Mário Veiga F Pereira, and Sérgio Granville. A new benders decomposition approach to solve power transmission network design problems. *IEEE Transactions on Power Systems*, 16(2):235–240, 2001.

[20] Merve Bodur, Sanjeeb Dash, Oktay Günlük, and James Luedtke. Strengthened benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing*, 29(1):77–91, 2017.

[21] Natashia Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The continuous-time service network design problem. *Operations research*, 65(5):1303–1321, 2017.

[22] José Brandão. Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows. *Computers & Industrial Engineering*, 120:146–159, 2018.

[23] Olli Bräysy and Michel Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. *Top*, 10(2):211–237, 2002.

[24] US Census. 2020 census urban areas facts, 2020.

[25] Santiago Cerisola, Álvaro Baíllo, José M Fernández-López, Andrés Ramos, and Ralf Gollmer. Stochastic power generation unit commitment in electricity markets: A novel formulation and a comparison of solution methods. *Operations research*, 57(1):32–46, 2009.

[26] Mervat Chouman and Teodor Gabriel Crainic. Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, 49(1):99–113, 2015.

[27] Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. Maritime transportation. *Handbooks in operations research and management science*, 14:189–284, 2007.

[28] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal*, 30(2):105–119, 1997.

[29] Jean-François Cordeau and Gilbert Laporte. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 39(3):292–298, 2001.

[30] Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936, 2001.

[31] Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society*, 55(5):542–546, 2004.

[32] Jean-Francois Cordeau, Paolo Toth, and Daniele Vigo. A survey of optimization models for train routing and scheduling. *Transportation science*, 32(4):380–404, 1998.

[33] Teodor G Crainic and Jean-Marc Rousseau. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225–242, 1986.

[34] Teodor G Crainic and Jean-Marc Rousseau. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225–242, 1986.

[35] Teodor Gabriel Crainic. Service network design in freight transportation. *European journal of operational research*, 122(2):272–288, 2000.

[36] Teodor Gabriel Crainic. Service network design in freight transportation. *European journal of operational research*, 122(2):272–288, 2000.

[37] Teodor Gabriel Crainic. Long-haul freight transportation. In *Handbook of transportation science*, pages 451–516. Springer, 2003.

[38] Teodor Gabriel Crainic. City logistics. In *State-of-the-art decision-making tools in the information-intensive age*, pages 181–212. INFORMS, 2008.

[39] Teodor Gabriel Crainic, Michel Gendreau, and Bernard Gendron. Network design with applications to transportation and logistics, 2021.

[40] Teodor Gabriel Crainic and Mike Hewitt. *Service network design*. Springer, 2021.

[41] Teodor Gabriel Crainic, Mike Hewitt, Michel Toulouse, and Duc Minh Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2016.

[42] Teodor Gabriel Crainic, Mike Hewitt, Michel Toulouse, and Duc Minh Vu. Scheduled service network design with resource acquisition and management. *EURO Journal on Transportation and Logistics*, 7(3):277–309, 2018.

[43] Teodor Gabriel Crainic and Kap Hwan Kim. Intermodal transportation. *Handbooks in operations research and management science*, 14:467–537, 2007.

[44] Teodor Gabriel Crainic and Benoit Montreuil. Physical internet enabled hyperconnected city logistics. *Transportation Research Procedia*, 12:383–398, 2016.

[45] Teodor Gabriel Crainic, Walter Rei, Mike Hewitt, and Francesca Maggioni. *Partial Benders decomposition strategies for two-stage stochastic integer programs*, volume 37. CIRRELT, 2016.

[46] Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Advanced freight transportation systems for congested urban areas. *Transportation Research Part C: Emerging Technologies*, 12(2):119–137, 2004.

[47] Teodor Gabriel Crainic and Jacques Roy. Or tools for tactical freight transportation planning. *European Journal of Operational Research*, 33(3):290–297, 1988.

[48] Teodor Gabriel Crainic and Antonino Sgalambro. Service network design models for two-tier city logistics. *Optimization Letters*, 8(4):1375–1387, 2014.

[49] T.G. Crainic, M. Hewitt, and M. Toulouse. Scheduled service network design with resource acquisition and management. *EURO Journal on Transportation and Logistics*, 7:277–309, 2017.

[50] T.G. Crainic, N. Ricciardi, , and G. Storchi. Models for evaluating and planning city logistics systems. *Transportation Science*, 43:432–454, 2009.

[51] Ethan Cramer-Flood. Global ecommerce update 2021, 2021.

[52] Marcos Carneiro Da Silva, Paulo Morelato Franca, and Paulo D Bishop Da Silveira. Long-range planning of power distribution systems: secondary networks. *Computers & electrical engineering*, 22(3):179–191, 1996.

[53] Lars Dahle, Henrik Andersson, Marielle Christiansen, and M Grazia Speranza. The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122–133, 2019.

[54] Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research*, 55:167–184, 2015.

[55] Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95:95–123, 2016.

[56] Iman Dayarian, Adolfo Rocco, Alan Erera, and Martin Savelsbergh. Operations design for high-velocity intra-city package service. *Transportation Research Part B: Methodological*, 161:150–168, 2022.

[57] Iman Dayarian and Martin Savelsbergh. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174, 2020.

[58] Emrah Demir, Wolfgang Burgholzer, Martin Hrušovskỳ, Emel Arıkan, Werner Jammernegg, and Tom Van Woensel. A green intermodal service network design problem with travel time uncertainty. *Transportation Research Part B: Methodological*, 93:789–807, 2016.

[59] Xuefei Deng, Kshiti D Joshi, and Robert D Galliers. The duality of empowerment and marginalization in microtask crowdsourcing. *MIS quarterly*, 40(2):279–302, 2016.

[60] Guy Desaulniers, Oli BG Madsen, and Stefan Ropke. Chapter 5: The vehicle routing problem with time windows. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 119–159. SIAM, 2014.

[61] Émilie Dufour, Gilbert Laporte, Julie Paquette, and Marie-Ève Rancourt. Logistics service network design for humanitarian response in east africa. *Omega*, 74:1–14, 2018.

[62] A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang. Improved load plan design through integer programming based local search. *Operations Research*, 47(3):412–427, 2013.

[63] J. M. Farvolden and W. B. Powell. Subgradient methods for the service network design problem. *Transportation Science*, 28(3):177–272, 1994.

[64] Judith M Farvolden and Warren B Powell. Subgradient methods for the service network design problem. *Transportation Science*, 28(3):256–272, 1994.

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

[66] Olaf E Flippo and Alexander HG Rinnooy Kan. Decomposition in general mathematical programming. *Mathematical Programming*, 60(1):361–382, 1993.

[67] Pirmin Fontaine, Teodor Gabriel Crainic, Ola Jabali, and Walter Rei. Scheduled service network design with resource management for two-tier multimodal city logistics. *European Journal of Operational Research*, 294(2):558–570, 2021.

[68] Véronique François, Yasemin Arda, and Yves Crama. Adaptive large neighborhood search for multitrip vehicle routing with time windows. *Transportation Science*, 53(6):1706–1730, 2019.

[69] Zhuo Fu, Richard Eglese, and Leon YO Li. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society*, 59(5):663–673, 2008.

[70] Viviane Gascon, Abdelhamid Benchakroun, and Jacques A Ferland. Electricity distribution planning model: A network design approach for solving the master problem of the benders decomposition method. *INFOR: Information Systems and Operational Research*, 31(3):205–220, 1993.

[71] Bezalel Gavish. Formulations and algorithms for the capacitated minimal directed tree problem. *Journal of the ACM (JACM)*, 30(1):118–132, 1983.

[72] Arthur M Geoffrion. Elements of large-scale mathematical programming part i: Concepts. *Management Science*, 16(11):652–675, 1970.

[73] Arthur M Geoffrion. Elements of large scale mathematical programming part ii: Synthesis of algorithms and bibliography. *Management Science*, 16(11):676–691, 1970.

[74] Ilfat Ghamlouche, Teodor Gabriel Crainic, and Michel Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations research*, 51(4):655–667, 2003.

[75] Fred Glover. Future paths for integer programming and links to ar tifi cial intelli g en ce. *Computers operations research*, 13(5):533–549, 1986.

[76] Fred Glover and Manuel Laguna. *Tabu search*. Springer, 1998.

[77] Luis Gouveia. A 2n constraint formulation for the capacitated minimal spanning tree problem. *Operations research*, 43(1):130–141, 1995.

[78] A Grothey, S Leyffer, and KIM McKinnon. A note on feasibility in benders decomposition. *Numerical Analysis Report NA/188, Dundee University*, 1999.

[79] Gianfranco Guastaroba, Maria Grazia Speranza, and Daniele Vigo. Intermediate facilities in freight transportation planning: a survey. *Transportation Science*, 50(3):763–789, 2016.

[80] Edward He, Natashia Boland, George Nemhauser, and Martin Savelsbergh. An exact algorithm for the service network design problem with hub capacity constraints. *Networks*, 80(4):572–596, 2022.

[81] Mike Hewitt. Enhanced dynamic discretization discovery for the continuous time load plan design problem. *Transportation Science*, 53(6):1731–1750, 2019.

[82] Mike Hewitt. The flexible scheduled service network design problem. *Transportation Science*, 2022.

[83] Mike Hewitt, Teodor Gabriel Crainic, Maciek Nowak, and Walter Rei. Scheduled service network design with resource acquisition and management under uncertainty. *Transportation Research Part B: Methodological*, 128:324–343, 2019.

[84] Arild Hoff, Arnt-Gunnar Lium, Arne Løkketangen, and Teodor Gabriel Crainic. A metaheuristic for stochastic service network design. *Journal of Heuristics*, 16(5):653–679, 2010.

[85] Toshihide Ibaraki, Shinji Imahori, Koji Nonobe, Kensuke Sobue, Takeaki Uno, and Mutsunori Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics*, 156(11):2050–2069, 2008.

[86] A. I. Jarrah, E. Johnson, and L. C. Neubert. Large-scale, less-than-truckload service network design. *Operations Research*, 57(3):609–625, 2009.

[87] Xiaoping Jiang, Ruibin Bai, Jason Atkin, and Graham Kendall. A scheme for determining vehicle routes based on arc-based service network design. *INFOR: Information Systems and Operational Research*, 55(1):16–37, 2017.

[88] Xiaoping Jiang, Ruibin Bai, Stein W Wallace, Graham Kendall, and Dario Landa-Silva. Soft clustering-based scenario bundling for a progressive hedging heuristic in stochastic service network design. *Computers & Operations Research*, 128:105182, 2021.

[89] Christian Vad Karsten, Berit Dangaard Brouer, Guy Desaulniers, and David Pisinger. Time constrained liner shipping network design. *Transportation Research Part E: Logistics and Transportation Review*, 105:152–162, 2017.

[90] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[91] Min Jung Kim. Benefits and concerns of the sharing economy: economic analysis and policy implications. *KDI Journal of Economic Policy*, 41(1):15–41, 2019.

[92] Nayeon Kim, Benoit Montreuil, Walid Klibi, and Nitish Kholgade. Hyperconnected urban fulfillment and delivery. *Transportation Research Part E: Logistics and Transportation Review*, 145:102104, 2021.

[93] Kerim U Kızıl and Barış Yıldız. Public transport-based crowd-shipping with backup transfers. *Transportation Science*, 2022.

[94] Walid Klibi, Alain Martel, and Adel Guitouni. The design of robust value-creating supply chain networks: a critical review. *European Journal of Operational Research*, 203(2):283–293, 2010.

[95] Attila A Kovacs, Bruce L Golden, Richard F Hartl, and Sophie N Parragh. Vehicle routing problems in which consistency considerations are important: A survey. *Networks*, 64(3):192–213, 2014.

[96] Giacomo Lanza, Teodor Gabriel Crainic, Walter Rei, and Nicoletta Ricciardi. Service network design problem with quality targets and stochastic travel times. Technical report, CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise . . . , 2017.

[97] Giacomo Lanza, Teodor Gabriel Crainic, Walter Rei, and Nicoletta Ricciardi. Scheduled service network design with quality targets and stochastic travel times. *European Journal of Operational Research*, 288(1):30–46, 2021.

[98] Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

[99] Gilbert Laporte and Francois V. Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133–142, 1993.

[100] Hongtao Lei, Gilbert Laporte, and Bo Guo. The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research*, 38(12):1775–1783, 2011.

[101] Xiangyong Li, Yi Ding, Kai Pan, Dapei Jiang, and YP Aneja. Single-path service network design problem with resource constraints. *Transportation Research Part E: Logistics and Transportation Review*, 140:101945, 2020.

[102] Xiangyong Li, Kai Wei, YP Aneja, and Peng Tian. Design-balanced capacitated multicommodity network design with heterogeneous assets. *Omega*, 67:145–159, 2017.

[103] Xiangyong Li, Kai Wei, Zhaoxia Guo, Wei Wang, and YP Aneja. An exact approach for the service network design problem with heterogeneous resource constraints. *Omega*, page 102376, 2020.

[104] Xiangyong Li, Kai Wei, Zhaoxia Guo, Wei Wang, and YP Aneja. An exact approach for the service network design problem with heterogeneous resource constraints. *Omega*, 102:102376, 2021.

[105] Chuanju Liu, Shaochong Lin, Zuo-Jun Max Shen, and Junlong Zhang. Stochastic service network design: The value of fixed routes. *Transportation Research Part E: Logistics and Transportation Review*, 174:103118, 2023.

[106] Ying Liu and Yongmei Liu. The effect of workers' justice perception on continuance participation intention in the crowdsourcing market. *Internet Research*, 2019.

[107] A. Lium, T. G. Crainic, and S. W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43:144–157, 2009.

[108] Arnt-Gunnar Lium, Teodor Gabriel Crainic, and Stein W Wallace. Correlations in stochastic programming: A case from stochastic service network design. *Asia-Pacific Journal of Operational Research*, 24(02):161–179, 2007.

[109] Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Demetrio Laganà. The vehicle routing problem with occasional drivers and time windows. In *International Conference on Optimization and Decision Science*, pages 577–587. Springer, 2017.

[110] Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, 113:104806, 2020.

[111] Thomas L Magnanti and Richard T Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations research*, 29(3):464–484, 1981.

[112] Thomas L Magnanti and Richard T Wong. Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55, 1984.

[113] Florian Martin, Vera C Hemmelmayr, and Tina Wakolbinger. Integrated express shipment service network design with customer choice and endogenous delivery time restrictions. *European Journal of Operational Research*, 2021.

[114] Dale McDaniel and Mike Devine. A modified benders' partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319, 1977.

[115] Julien Michallet, Christian Prins, Lionel Amodeo, Farouk Yalaoui, and Grégoire Vitry. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & operations research*, 41:196–207, 2014.

[116] Michel Minoux. Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360, 1989.

[117] B. Montreuil. Physical internet manifesto: globally transforming the way physical objects are handled, moved, stored, realized, supplied and used, versions 1.1 to 1.11. 2009.

[118] Benoit Montreuil. Toward a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2):71–87, 2011.

[119] Kianoush Mousavi, Merve Bodur, and Matthew J Roorda. Stochastic last-mile delivery with crowd-shipping and mobile depots. *Transportation Science*, 56(3):612–630, 2022.

[120] Yuichi Nagata, Olli Bräysy, and Wout Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & operations research*, 37(4):724–737, 2010.

[121] Phuong Khanh Nguyen, Teodor Gabriel Crainic, and Michel Toulouse. A tabu search for time-dependent multi-zone multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 231(1):43–56, 2013.

[122] Phuong Khanh Nguyen, Teodor Gabriel Crainic, and Michel Toulouse. A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. *Journal of Heuristics*, 20:383–416, 2014.

[123] Santiago Nieto-Isaza, Pirmin Fontaine, and Stefan Minner. The value of stochastic crowd resources and strategic location of mini-depots for last-mile delivery: A benders decomposition approach. *Transportation Research Part B: Methodological*, 157:62–79, 2022.

[124] Morton E O'Kelly and Harvey J Miller. The hub network design problem: a review and synthesis. *Journal of Transport Geography*, 2(1):31–40, 1994.

[125] Binbin Pan, Zhenzhen Zhang, and Andrew Lim. Multi-trip time-dependent vehicle routing problem with time windows. *European Journal of Operational Research*, 291(1):218–231, 2021.

[126] Shenle Pan, Eric Ballot, George Q Huang, and Benoit Montreuil. Physical internet and interconnected logistics services: research and applications, 2017.

[127] Michael Berliner Pedersen, Teodor Gabriel Crainic, and Oli BG Madsen. Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43(2):158–177, 2009.

[128] David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.

[129] Aymeric Punel, Alireza Ermagun, and Amanda Stathopoulos. Studying determinants of crowd-shipping use. *Travel Behaviour and Society*, 12:30–40, 2018.

[130] Thomas Puschmann and Rainer Alt. Sharing economy. *Business & Information Systems Engineering*, 58(1):93–99, 2016.

[131] Ragheb Rahmaniani, Shabbir Ahmed, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders dual decomposition method. *Operations Research*, 68(3):878–895, 2020.

[132] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[133] Ragheb Rahmaniani, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. Accelerating the benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28(1):875–903, 2018.

[134] A. Rajagopal. *Crowd-based business models—Using collective intelligence for market competitiveness*. Palgrave Macmillan, 2021.

[135] CD Randazzo and Henrique Pacca Loureiro Luna. A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research*, 106(1):263–286, 2001.

[136] CD Randazzo, Henrique Pacca Loureiro Luna, and Philippe Mahey. Benders decomposition for local access network design with two technologies. *Discrete mathematics and theoretical computer science*, 4(2):235–246, 2001.

[137] R Romero and A Monticelli. A hierarchical decomposition approach for transmission network expansion planning. *IEEE transactions on power systems*, 9(1):373–380, 1994.

[138] Ann-Kathrin Rothenbächer, Michael Drexl, and Stefan Irnich. Branch-and-price-and-cut for a service network design and hub location problem. *European Journal of Operational Research*, 255(3):935–947, 2016.

[139] Jacques Roy and Teodor Gabriel Crainic. Improving intercity freight routing with a tactical planning model. *Interfaces*, 22(3):31–44, 1992.

[140] Jacques Roy and Louis Delorme. Netplan: A network optimization model for tactical planning in the less-than-truckload motor-carrier industry. *INFOR: Information Systems and Operational Research*, 27(1):22–35, 1989.

[141] Ozgur Satici and Iman Dayarian. Tactical and operational planning of express intra-city package services. *Omega*, page 102940, 2023.

[142] Ozgur Satici and Iman Dayarian. Tactical and operational planning of express intra-city package services. *Omega*, 122:102940, 2024.

[143] Yannick Oskar Scherr, Mike Hewitt, Bruno Albert Neumann Saavedra, and Dirk Christian Mattfeld. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195, 2020.

[144] Yannick Oskar Scherr, Bruno Albert Neumann-Saavedra, Mike Hewitt, and Dirk Christian Mattfeld. Service network design for same day delivery with mixed autonomous fleets. *Transportation research procedia*, 30:23–32, 2018.

[145] Juliet Schor et al. Debating the sharing economy. *Journal of self-governance and management economics*, 4(3):7–22, 2016.

[146] Xiaochuan Song, Graham H Lowman, and Peter Harms. Justice for the crowd: Organizational justice and turnover in crowd-based labor. *Administrative Sciences*, 10(4):93, 2020.

[147] David Soto Setzke, Christoph Pflügler, Maximilian Schreieck, Sven Fröhlich, Manuel Wiesche, and Helmut Krcmar. Matching drivers and transportation requests in crowdsourced delivery systems. 2017.

[148] Statista. Pitney bowes parcel shipping index, 2022.

[149] Nicolas Teypaz, Susann Schrenk, and Van-Dat Cung. A decomposition scheme for large-scale service network design with asset management. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):156–170, 2010.

[150] Fabian Torres, Michel Gendreau, and Walter Rei. Vehicle routing with stochastic supply of crowd vehicles and time windows. *Transportation Science*, 56(3):631–653, 2022.

[151] Horst Treiblmaier, Kristijan Mirkovski, and Paul Benjamin Lowry. Conceptualizing the physical internet: Literature review, implications and directions for future research. In *11th CSCMP Annual European Research Seminar, Vienna, Austria, May*, 2016.

[152] UN. 2018 revision of world urbanization prospects, 2019.

[153] Bart van Riessen, Rudy Negenborn, Rommert Dekker, and Gabriel Lodewijks. Service network design for an intermodal container network with flexible due dates/times and the possibility of using subcontracted transport. Technical report, 2013.

[154] Richard M Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM journal on applied mathematics*, 17(4):638–663, 1969.

[155] T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. Time-window relaxations in vehicle routing heuristics. *Journal of Heuristics*, 21(3):329–358, 2015.

[156] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research*, 40(1):475–489, 2013.

[157] Duc Minh Vu, Teodor Gabriel Crainic, and Michel Toulouse. A three-phase matheuristic for capacitated multi-commodity fixed-cost network design with design-balance constraints. *Journal of heuristics*, 19(5):757–795, 2013.

[158] Xin Wang, Teodor Gabriel Crainic, and Stein W Wallace. Stochastic network design for planning scheduled transportation services: the value of deterministic solutions. *INFORMS Journal on Computing*, 31(1):153–170, 2019.

[159] Zujian Wang and Mingyao Qi. Robust service network design under demand uncertainty. *Transportation Science*, 54(3):676–689, 2020.

[160] Zujian Wang, Mingyao Qi, Chun Cheng, and Canrong Zhang. A hybrid algorithm for large-scale service network design considering a heterogeneous fleet. *European Journal of Operational Research*, 276(2):483–494, 2019.

[161] Christopher Watson. Trends in world urbanisation. In *Proceedings of ICUP-1st International Conference on Urban Pests. Cambridge, England*, 1993.

[162] Nicole Wieberneit. Service network design for freight transportation: a review. *OR spectrum*, 30(1):77–112, 2008.

[163] Nicole Wieberneit. Service network design for freight transportation: a review. *OR spectrum*, 30(1):77–112, 2008.

[164] Haotian Wu, Ian Herszterg, Martin Savelsbergh, and Yixiao Huang. Service network design for same-day delivery with hub capacity constraints. *Transportation Science*, 57(1):273–287, 2023.

[165] Haotian Wu, Martin Savelsbergh, and Yixiao Huang. Planning the city operations of a parcel express company. *Omega*, 107:102539, 2022.

[166] Yangkun Xia and Zhuo Fu. Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate. *Cluster Computing*, 22(Suppl 4):8725–8733, 2019.

[167] Barış Yıldız and Martin Savelsbergh. Optimizing package express operations in china. *European Journal of Operational Research*, 300(1):320–335, 2022.

[168] Lu Zhen, Chengle Ma, Kai Wang, Liyang Xiao, and Wei Zhang. Multi-depot multi-trip vehicle routing problem with time windows and release dates. *Transportation Research Part E: Logistics and Transportation Review*, 135:101866, 2020.

[169] Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175:461–502, 2019.

APPENDIX A

CHAPTER 3 APPENDIX

A.1 Hub Clustering

Ideally in the network, to achieve the best economies of scale, priority should be given to creating short channels between hub pairs with the highest traffic. Thus, in our clustering approach, grouping node pairs with the highest traffic and the least distance is favored. This can be achieved by using a distance metric that is weighted by the inverse of inter-node traffic volume. To guarantee connectivity of the network, clusters must be created with overlaps. That is, the created clusters are not disjoint and a hub may be a member of multiple clusters so that all hub pairs with non-zero demand rates are connected through at least one path of length less than S.

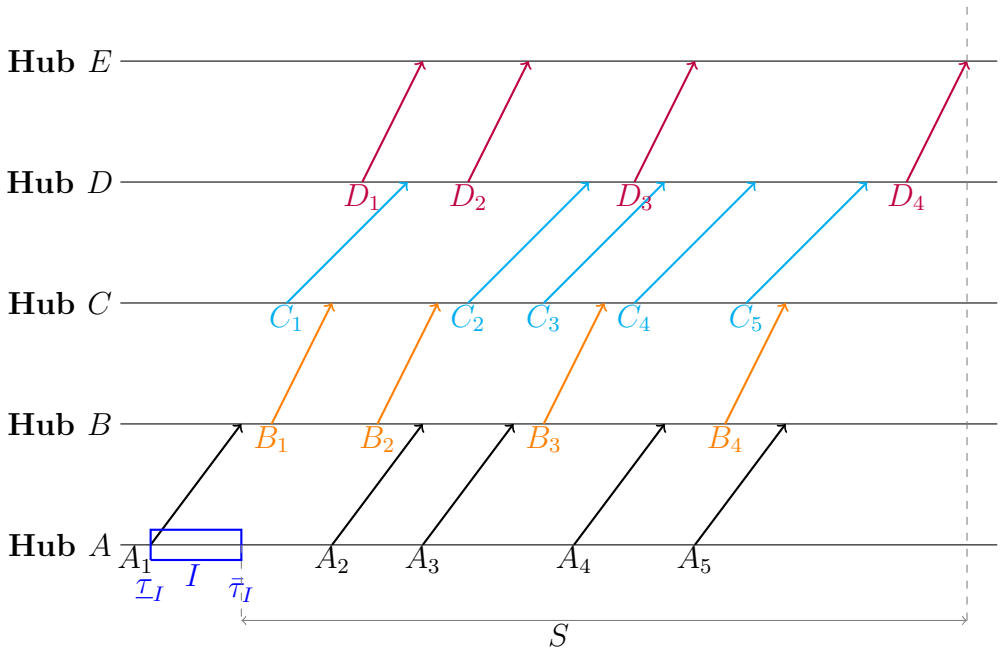
We manage hub clustering decisions using two parameters k and m, respectively referring to the number of clusters and maximum membership, i.e., the maximum number of clusters that a hub can be a member of. The membership limit of a hub can be influenced by operational limitations such as its throughput restrictions. Obviously, hubs at the center of the network tend to be members of multiple clusters to enable connectivity. A membership limit for each hub disperses the traffic among them, ensuring that all of the hubs can fulfill the process without getting overwhelmed.

Given set values for m and k, we follow two main steps to construct the clusters. In the first step, k distinct clusters of the same size are created, while in the second step, cluster overlaps are enforced to the point the desired connectivity is achieved.

Step 1: We apply the partitioning around medoids (PAM) algorithm introduced by (90) to solve a k-medoids problem to obtain k distinct clusters. The k-medoids method is similar to the well-known k-means clustering method. A problem with the k-means clustering is that the final centroids are not interpretable, i.e., centroids are not necessarily actual nodes of the graph. This is mainly problematic in our context as our weighted distance metric takes into account not only the pairwise distances but also the pairwise traffic, while the latter is only relevant between actual nodes of the graph. In the PAM algorithm, first, k random nodes are selected as medoids. Then, each remaining node is associated with the closest medoid. Then, the swap of each medoid and non-medoid node is considered and the corresponding cost is calculated and the best swap is performed if it decreases the cost of the network. This operation is repeated until no further improvement can be made. We enforce the cluster sizes to be the same in the PAM algorithm by associating the nodes to the closest cluster only if the cluster is not "full".

Step 2: Once we have the k same-sized, distinct and non-overlapping clusters, to create connectivity the overlapping procedure is followed. For each node-cluster pair (clusters other than the one the node is currently assigned to), a relationship function that calculates the total traffic based on the weighted distance metric between the node and the cluster members is constructed. Next, for each node, the difference between its top two relationships is calculated as the regret value. The node with the highest regret value is added to the cluster with the highest relationship with the node, incrementing the size of the cluster and membership count of the node by one. Once a node reaches its maximum membership limit m, it is no longer considered to join other clusters. The procedure is repeated until the desired level of connectivity is reached, i.e., all hub pairs with a non-zero demand rate are connected through at least one path with a length less than S. In the resulting network, all the hubs within the same cluster are connected through direct arcs while hubs in two different clusters are connected through paths via one or more intermediate hubs.

Figure A.1: Time-space network, commodity A-E

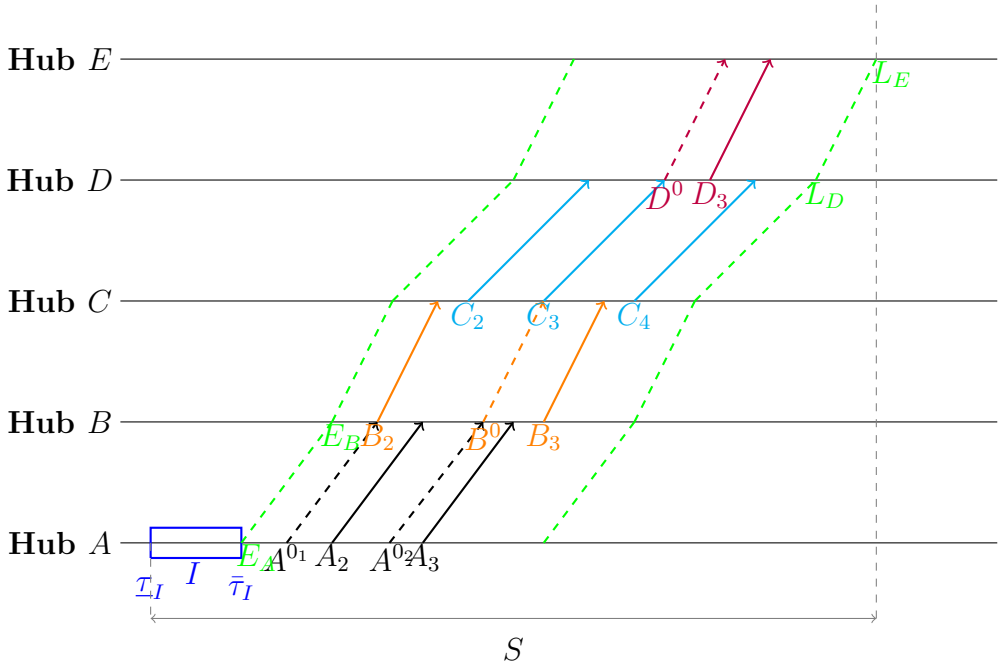


A.2 Enumerating all possible lateness recovery alternatives

We propose an approach that relies on enumerating all possibilities implemented through a search tree structure that can be described as follows using Figure A.1. Consider commodity A-E with a commodity path $A{-}B{-}C{-}D{-}E$. Figure A.1 shows the set of existing dispatches along the arcs of this commodity path. Shipments received between dispatches A_1 and A_2 at the origin hub A would be loaded on dispatches, A_2, B_3, C_4, and D_4 (vehicle capacity permitting). This sequence of dispatches represents the earliest dispatches along the arcs of the commodity path that can move shipments received between A_1 and A_2 toward their destination. However, given the service commitment S, it can be observed that shipments received within interval $I := [\underline{\tau}_I, \bar{\tau}_I]$ will reach hub E late (violating service commitment S). One can identify extra dispatches to be added during specific time windows along one or potentially multiple arcs of the commodity path such that the lateness associated with shipments received during time interval I is prevented.

The earliest time an extra dispatch along arc AB can be added to recover the lateness of I is right at the end of the interval I, denoted by $E_A := \bar{\tau}_I$. Similarly, one can identify the

Figure A.2: Recovery plan for interval I



earliest times dispatches along each subsequent arc of the commodity path that can be added to potentially be part of a path to recover lateness of I. For example, the earliest dispatch along BC would be at time $E_B := E_A + t_{AB}$ (for the sake of simplicity of presentation, loading/unloading times are ignored). The latest possible dispatches can be determined considering the fact that a shipment arrived at the origin hub A at the beginning of I, must reach hub E no later than $L_E := \underline{\tau}_I + S$. The latest dispatch times along the arcs of the path can be backtracked considering the arc lengths. For example, $L_D := L_E - t_{DE}$. These time boundaries for dispatches along the arcs of the commodity path are depicted by green dashed lines in Figure A.2, while existing dispatches falling outside of these boundaries are disregarded. Any dispatch along the arcs of the commodity path that falls within these boundaries can potentially be part of a path, referred to as an *alternative* that recovers the lateness of shipments arrived during I.

By sweeping the existing dispatches along the arcs of the commodity path in chronological order of its arcs and in ascending order of their dispatch times, a series of alternatives

incorporating each existing dispatch is identified. The special ordering of processing dispatches guarantees that no alternative is generated more than once.

For each existing dispatch along an arc, a series of forward and backward partial paths are generated such that the concatenation of a forward and a backward partial path forms a complete path recovering the current lateness related to I. Let $a(1), \ldots, a(i), \ldots, a(n)$ be the sequence of arcs along a commodity path. Let $d_{a(i)}$ be an existing dispatch along arc $a(i)$.

Forward Partial Paths: We construct the set of forward partial paths using a tree structure (see Figure A.3a). The root of such a tree would be $d_{a(i)}$. Each node of the tree has at least one child node. The child nodes of a given node associated with dispatch $d_{a(j)}$ are:

- $d^0_{a(j+1)}$: An extra dispatch along arc $a(j+1)$ with a departure time that coincides the arrival time of $d_{a(j)}$, if no dispatch is currently scheduled along $a(j+1)$ at that time. This represents the earliest dispatch along $a(j+1)$ that can follow $d_{a(j)}$.

- One child node per existing dispatch along $a(j+1)$ such that its departure time is between the arrival of $d_{a(j)}$ and L_{j+1}.

Once all the child nodes associated with $d_{a(i)}$ are generated, we iterate through such nodes one by one to identify their child nodes based on the same logic explained above. This procedure is repeated until the last arc of the path, namely $a(n)$. Accordingly, such a tree will have a depth of $n - i + 1$. Each branch of this tree starting from the root node all the way down to one of the leaves of the tree represents a partial forward path based on an existing dispatch $d_{a(i)}$.

Backward Partial Path: We construct one backward partial path per exiting dispatch using a separate tree structure (see Figure A.3b). Similar to the case of the forward partial path, the root of the backward tree is $d_{a(i)}$. Each node of the tree has exactly one child node. The child node of a given node associated with dispatch $d_{a(j)}$ is:

Figure A.3: Forward and Backward Partial Paths Starting from C_3

(a) Forward Partial Paths

(b) Backward Partial Path

- $d^0_{a(j-1)}$: An extra dispatch along arc $a(j-1)$ such that its arrival time coincides the departure time of $d_{a(j)}$, if no existing dispatch is currently scheduled along $a(j-1)$ at that time. This represents the latest dispatch along $a(j-1)$ that can precede $d_{a(j)}$.

For the child of $d_{a(i)}$, identify its child node based on the same logic explained above. This procedure is repeated until the first arc of the path, namely $a(1)$. Accordingly, such a tree will have a depth i. The only branch of this tree starting from the root node all the way down to the leaf of the tree represents a partial path starting from the origin of the commodity and ending at existing dispatch $d_{a(i)}$.

Each combination of the backward partial path and a forward partial path associated with an existing dispatch $d_{a(i)}$ forms an alternative recovery plan. The same procedure is repeated for all existing dispatches within the boundaries of $[E_a, L_a]$ associated with a given interval I in the aforementioned chronological order of dispatches.

Figure A.3 shows the forward and backward trees constructed based on the existing dispatch C_3. The set of such alternatives identified are:

1. $A^{0_2} - B^0 - C_3 - D^0$, incorporating three extra dispatches;

2. $A^{0_2} - B^0 - C_3 - D_3$, incorporating two extra dispatches;

The extra dispatches incorporated into the alternatives have some flexibility in terms of time so that the time continuity of the dispatches in the alternative is not disrupted. Such

time flexibility can be expressed as a time window $[\underline{\pi}, \bar{\pi}]$ as $\underline{\pi}$ being the start time and $\bar{\pi}$ end time of the time window. While finding the time window of an extra dispatch, two factors must be considered: (1) The position of the extra dispatch with respect to the alternative path, i.e. first arc, last arc, or middle arcs, and (2) Whether in the selected alternative, the dispatches in the previous or next arcs are existing ones or extra dispatches. Suppose we are interested in determining the time window associated with an extra dispatch on a given arc a. If dispatches preceding and succeeding the extra dispatch along the alternative are existing dispatches, then we can consider 3 cases for $\underline{\pi}$ and $\bar{\pi}$ values.

1. If the extra dispatch is on a middle arc of the path, then the earliest departure time is bounded by the arrival of the dispatch in the preceding arc: $\underline{\pi}$ is set to the arrival time of the dispatch in the preceding arc. On the other hand, the latest departure time is set such that its arrival at the end of the arc coincides with the departure time of the existing dispatch on the succeeding arc.

2. If the extra dispatch is the first arc on its path, the earliest departure time is $\underline{\pi} = E_a$, while the latest departure time $\bar{\pi}$ is determined in the same manner as for a middle arc.

3. If the extra dispatch is the last arc on its path, $\underline{\pi}$ is set to the arrival time of the dispatch in the preceding arc, and $\bar{\pi} = L_{a+1} - t_a$.

If in the selected alternative, there are consecutive extra dispatches, then their time windows constitute boundaries for one another. Therefore, the total time flexibility of the consecutive extra dispatches will be shared among them equally.

A.3 Time Redistribution Model Results

In Table A.1, the results of Model 3.4 are shown for different values of parameters q, S, κ, and when $m = 5$. For each instance, two values are reported. The first value is the number of vehicles removed $\bar{Y}$ and the second value is the CPU time (in seconds) of the model.

Table A.1: Results of Model 3.4: Number of vehicles removed and CPU time (seconds)

q	**50**		**100**		**150**		**200**		**250**	
S	$\bar{Y}$	CPU	$\bar{Y}$	CPU	$\bar{Y}$	CPU	$\bar{Y}$	CPU	$\bar{Y}$	CPU
	1	0.2	11	0.2	21	0.2	24	0.2	29	0.3
8	1	0.2	12	0.1	26	0.2	38	0.1	44	0.2
	1	0.2	12	0.1	29	0.2	39	0.2	46	0.2
	0	0.1	2	0.1	4	0.1	4	0.1	5	0.1
10	0	0.1	3	0.1	4	0.2	4	0.1	8	0.1
	0	0.1	3	0.1	4	0.1	4	0.1	8	0.1
	0	0	0	0.1	1	0.1	0	0.1	2	0.1
12	0	0	0	0.1	1	0.1	0	0.1	2	0.1
	0	0	0	0.1	1	0.1	0	0.1	2	0.1

A.4 Total Number of vehicles with Approaches G, M3, and M3-G

The total fleet sizes for all instances are reported in Table A.2. The three values for each instance correspond to the three alternative selection approaches G, M3, and M3-G.

Table A.2: Total # of vehicles (Tactical & Operational) with approaches G, M3 and M3-G, when $\kappa = 2$

q	50	100	150	200	250
S	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G
8	642.5 | 577.5 | 585.5	411.6 | 349.2 | 356.1	335.5 | 275.4 | 282.3	229.0 | 229.0 | 229.0	214.0 | 214.0 | 214.0
10	534.0 | 534.0 | 534.0	312.8 | 308.0 | 308.1	259.2 | 242.8 | 242.2	246.2 | 210.5 | 209.8	191.0 | 191.0 | 191.0
12	522.5 | 522.5 | 522.5	296.1 | 295.0 | 295.0	222.6 | 220.4 | 220.4	193.9 | 190.9 | 190.9	182.3 | 173.8 | 173.8

A.5 Operational Planning Result Tables

A.5.1 Earliness and Lateness

Other evaluation metrics when putting the tactical plans to the test with the demand realizations are the average earliness per early shipment and average lateness per late shipment. These metrics are presented in Table A.3. For each instance, the first value is the average earliness per early package and the second value is the average lateness per late package in hours. If there are no packages delivered late in that instance, then the average lateness per late package is reported as zero.

Table A.3: Average earliness per early shipment and average lateness per late shipment, when $\kappa = 2$ (hours)

q	50	100	150	200	250
S	Early | Late	Early | Late	Early | Late	Early | Late	Early | Late
8	3.07 | 0.71	4.20 | 0.28	4.06 | 0.39	4.13 | 0	4.10 | 0
10	4.48 | 0	4.44 | 0.02	4.41 | 0.05	4.31 | 0.13	4.35 | 0
12	4.48 | 0	4.47 | 0	4.46 | 0	4.45 | 0	4.42 | 0.01

A.5.2 Identifying the Minimal Set of Extra Dispatches

The alternative selection process for identifying the minimal set of extra dispatches is performed by using the greedy approach, the IP-based alternative selection model, or the hybrid approach. Table A.4 shows the results of each approach where G represents the greedy approach, M3 represents the IP-based alternative selection model and the M3-G represents the hybrid approach, in which the IP model is warmstarted with the solution of the greedy.

Table A.4: Number of extra dispatches with approaches G, M3 and M3-G when $\kappa = 2$

q	50	100	150	200	250
S	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G
8	212.9 | 26.8 | 45.1	225.5 | 23.8 | 52.5	232.1 | 27.5 | 56.6	0 | 0 | 0	0 | 0 | 0
10	0 | 0 | 0	16.6 | 5.3 | 5.7	51.2 | 18.6 | 19.3	121.2 | 26.8 | 22.3	0 | 0 | 0
12	0.5 | 0.5 | 0.5	6.1 | 5.3 | 5.3	8.5 | 2.0 | 2.0	10.9 | 3.3 | 3.3	27.5 | 5.0 | 5.0

A.5.3 Vehicle Routing to Cover Extra Dispatches

The minimal set of extra dispatches obtained from Model (3.5) with or without warmstart is covered by a fleet of vehicles, where its size and vehicle routes are obtained by modeling and

solving a variant of VRPTW using tabu search. In Table A.5, for each instance, the number of additional vehicles needed to cover all extra dispatches obtained from three approaches G, M3, and M3-G are reported. Each value is the average of the number of extra vehicles required to cover the extra dispatches of the 10 demand realizations. The difference in the number of extra dispatches obtained from the three approaches also affects the number of additional vehicles.

Considering the number of extra dispatches and the number of extra vehicles in Tables A.4 and A.5, a similar pattern to those of service levels shown in Table 3.2 is observed. When both the service commitment and vehicle capacity constraints are tight, we have lower service levels and a higher number of late time intervals, hence, the highest required number of extra dispatches and vehicles.

Table A.5: Number of extra vehicles with approaches G, M3 and M3-G, when $\kappa = 2$

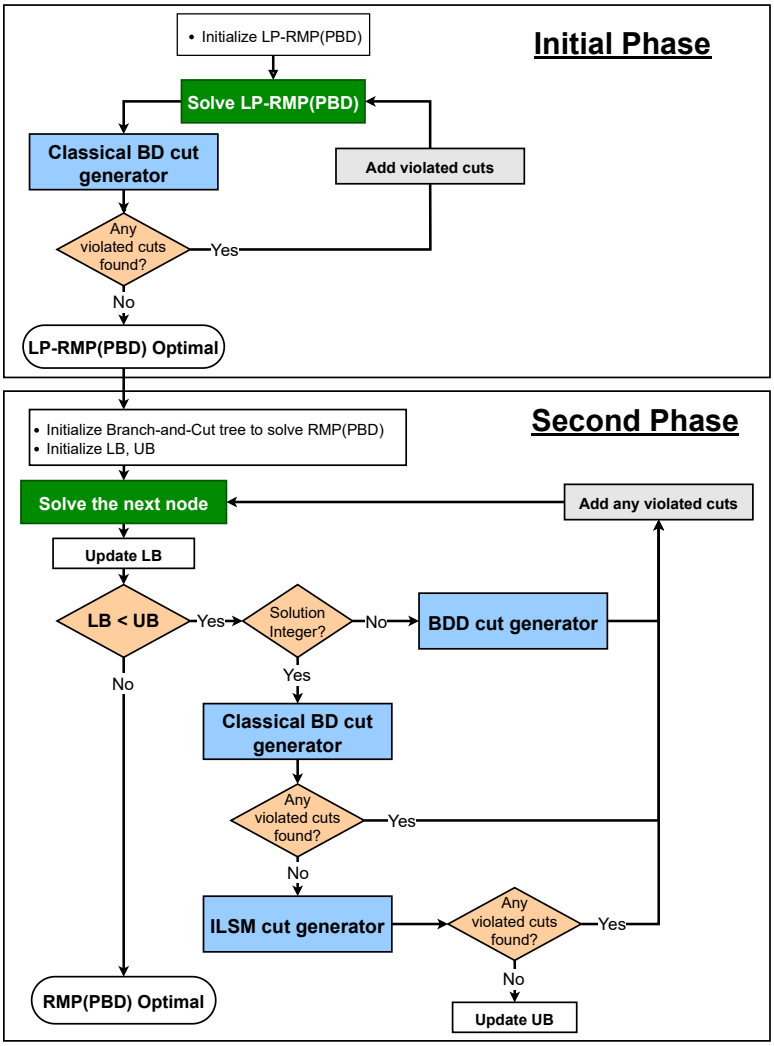
q	50	100	150	200	250
S	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G	G | M3 | M3-G
8	83.5 | 18.5 | 26.5	79.6 | 17.2 | 24.1	77.5 | 17.4 | 24.3	0 | 0 | 0	0 | 0 | 0
10	0 | 0 | 0	8.8 | 4.0 | 4.1	26.2 | 9.8 | 9.2	46.2 | 10.5 | 9.8	0 | 0 | 0
12	0.5 | 0.5 | 0.5	3.1 | 2.0 | 2.0	3.6 | 1.4 | 1.4	4.9 | 1.9 | 1.9	11.3 | 2.8 | 2.8

APPENDIX B

CHAPTER 4 APPENDIX

B.1 Proposed Branch-and-Benders Cut Algorithm Flowchart

Figure B.1: Algorithm flowchart



APPENDIX C

CHAPTER 5 APPENDIX

C.1 Construction Heuristics

Each of the construction heuristics developed is capable of generating a solution to the FRP either from scratch or by reconstructing partially destructed solutions. Each heuristic incorporates a phase dedicated to the addition of vehicle movements, followed by the subsequent implementation of the TS algorithm to establish vehicle routes.

1. **Myopic FIFO commodity assignment**

   In this method, commodities are sorted according to arrival times, and itineraries are constructed using a FIFO approach starting from the commodity with the earliest arrival. For each commodity, we examine each potential path realization and choose the least-cost itinerary. The itinerary may utilize the existing capacity on the network or may add capacity to certain arcs by adding new vehicle movements. To identify the least-cost itinerary in each potential path realization, we consider a temporal window based on the earliest and latest possible departure times on each arc of the path realization. Within that temporal window, we carry out both a forward traversal and a backward traversal to identify the itinerary with the lowest cost.

   The forward traversal initiates at the first arc of the path realization and assigns the commodity to the earliest available existing vehicle movement without incurring any additional cost. If there are no available vehicle movements within the designated time

window, we add a vehicle movement at the earliest time, incurring the full cost. With each subsequent arc, the time window is adjusted to start from the arrival time at that segment. This process is repeated until the commodity reaches its destination. The backward traversal mirrors the forward traversal, but starts at the destination and assigns the commodity to the most recent existing vehicle movement. The iteration continues until the itinerary connects the destination to the origin.

2. **Myopic urgency-based commodity assignment**

   This strategy closely resembles the FIFO approach, with the key difference being that the order of commodities is determined by their respective slack times. The slack time for a commodity is calculated as the disparity between its service commitment and the average total distance of its path realizations.

3. **Load-based planning**

   In this approach, commodities are iteratively allocated to arcs, whereby the collection of assigned arcs for each commodity constitutes an itinerary. We attach a numerical score to each potential commodity-arc assignment, and in each iteration, we implement the assignment with the highest score. Initial scores are calculated by dividing the total volume of commodities that could be assigned to a specific arc by the associated cost of that arc. After each assignment, any overlapping assignments associated with that commodity, such as assignments to the same arc, the preceding arc with a later arrival, the succeeding arc with an earlier arrival, path realizations that do not include that arc, or situations where parking/storage capacity is at maximum, are assigned a score of 0 to preclude assignments to those arcs. This iterative process is repeated until further assignments become unattainable.

4. **Arc and path realization assignments**

   In this method, during each iteration, we calculate the marginal cost for each commodity-arc assignment using the formula: $\overline{c_a}(f) = (\lceil \frac{F_a + f}{Q} \rceil - \lceil \frac{F_a}{Q} \rceil) c_a$ where F_a is the existing

volume on a. In each iteration, we opt for the assignment with the minimum cost and subsequently update the marginal costs following a procedure similar to the prior approach.

5. **Arc and path realization assignments with the depth-first approach**
   Building upon the preceding method, we incorporate an additional step in which we compute the overall cost for each path realization. In each iteration, subsequent to determining the commodity-arc assignment, we additionally specify a particular path realization for the commodity. Within that iteration, a subprocess is executed, during which an arc on each path realization is chosen. Subsequent to this selection, the costs are updated and the iteration is carried out until no further assignments can be made.

Construction Heuristic Enhancement Approaches

Similarly to the Model 5.2, we implement a set of approaches to increase the performance of the construction heuristics. This is done by applying discounts or penalties to the cost/score of each arc in the heuristics. To improve the even distribution of vehicle movements over time, we count the number of vehicle movements at each time period, and apply a cost penalty of $\%\omega^{time}$ to all arcs in the time period t with the highest number of vehicle movements. To improve the balance of vehicle movements, in each iteration, the arcs that can reduce the imbalance at the nodes are discounted by $\%\omega^{balance}$.

C.2 Detailed Instance-level Results

Table C.1: Instance-level results of Comprehensive IP approach

		Comprehensive IP							
		Time (hrs)	Opt gap	Total Cost		Fleet Size	# of vehicle mov		
$\|\mathcal{K}\|$	I			LB	UB		Total	Empty	Empty %
	1	24	2%	$9,237	$9,435	14	43	6	14%
	2	24	3%	$8,926	$9,246	14	39	5	13%
25	**3**	24	3%	$8,822	$9,061	12	39	6	15%
	4	24	3%	$8,702	$8,971	13	56	7	13%
	5	24	5%	$8,534	$8,971	14	45	7	16%
	1	24	12%	$13,606	$15,444	23	115	12	10%
	2	24	19%	$12,651	$15,599	25	125	14	11%
50	**3**	24	20%	$12,120	$15,131	22	100	8	8%
	4	24	15%	$13,391	$15,736	24	98	10	10%
	5	24	16%	$13,631	$16,208	22	100	9	9%
	1	24	20%	$19,667	$24,645	33	231	27	12%
	2	24	24%	$18,307	$24,152	31	192	21	11%
100	**3**	24	20%	$19,080	$23,910	30	192	19	10%
	4	24	22%	$17,858	$22,954	30	224	22	10%
	5	24	19%	$19,289	$23,872	33	186	18	10%
	1	24	52%	$10,776	$22,437	36	273	26	10%
	2	24	50%	$11,562	$23,110	35	324	24	7%
150	**3**	24	46%	$12,112	$22,417	39	320	26	8%
	4	24	49%	$10,982	$21,520	40	360	23	6%
	5	24	48%	$11,757	$22,596	39	324	25	8%
	1	24	nfs	$21,943	-	-	-	-	-
	2	24	nfs	$21,943	-	-	-	-	-
200	**3**	24	nfs	$21,943	-	-	-	-	-
	4	24	nfs	$22,382	-	-	-	-	-
	5	24	nfs	$21,935	-	-	-	-	-
	1	24	nfs	$23,616	-	-	-	-	-
	2	24	nfs	$24,088	-	-	-	-	-
250	**3**	24	nfs	$24,088	-	-	-	-	-
	4	24	nfs	$25,293	-	-	-	-	-
	5	24	nfs	$25,799	-	-	-	-	-
	1	-	mem	-	-	-	-	-	-
	2	-	mem	-	-	-	-	-	-
300	**3**	-	mem	-	-	-	-	-	-
	4	-	mem	-	-	-	-	-	-
	5	-	mem	-	-	-	-	-	-
	1	-	mem	-	-	-	-	-	-
	2	-	mem	-	-	-	-	-	-
400	**3**	-	mem	-	-	-	-	-	-
	4	-	mem	-	-	-	-	-	-
	5	-	mem	-	-	-	-	-	-
	1	-	mem	-	-	-	-	-	-
	2	-	mem	-	-	-	-	-	-
500	**3**	-	mem	-	-	-	-	-	-
	4	-	mem	-	-	-	-	-	-
	5	-	mem	-	-	-	-	-	-

Table C.2: Instance-level results of decomposition-based approach with the DBCs

		Decomposition (w/ DBC)									
		Time (hrs)			Opt gap	Total Cost		Fleet Size	# of vehicle mov		
$\|\mathcal{K}\|$	I	FRP	VSP	Total		LB	UB		Total	Empty	Empty %
	1	7	0	7	0%	$9,334	$9,334	14	44	7	16%
	2	7	0	7	0%	$8,857	$8,857	12	42	5	12%
25	**3**	7	0	7	0%	$9,010	$9,010	14	42	5	12%
	4	7	0	7	0%	$8,874	$8,874	12	53	9	17%
	5	7	0	7	0%	$8,790	$8,790	15	42	7	17%
	1	9	0	9	0%	$13,965	$13,965	21	118	13	11%
	2	10	0	10	0%	$14,292	$14,292	21	124	15	12%
50	**3**	10	0	10	0%	$13,299	$13,299	20	103	13	13%
	4	11	0	11	0%	$14,200	$14,200	20	94	11	12%
	5	11	0	11	0%	$14,590	$14,590	20	99	11	11%
	1	14	0	14	0%	$21,053	$21,053	29	248	28	11%
	2	14	0	14	0%	$21,852	$21,852	32	214	22	10%
100	**3**	15	0	15	0%	$21,819	$21,819	35	209	16	8%
	4	14	0	14	0%	$20,240	$20,240	32	242	21	9%
	5	13	0	13	0%	$23,460	$23,460	34	207	16	8%
	1	15	0	15	0%	$18,197	$18,197	35	247	25	10%
	2	16	0	16	0%	$18,015	$18,015	36	308	26	9%
150	**3**	15	0	15	0%	$17,475	$17,475	36	291	26	9%
	4	14	0	14	0%	$18,523	$18,523	34	339	27	8%
	5	13	0	13	0%	$18,894	$18,894	35	309	26	9%
	1	20	0	20	0%	$30,680	$30,680	41	326	25	8%
	2	21	0	21	0%	$30,373	$30,373	37	315	24	8%
200	**3**	23	0	23	0%	$28,551	$28,551	40	312	25	8%
	4	21	0	21	0%	$28,837	$28,837	36	292	24	8%
	5	23	0	23	0%	$31,720	$31,720	39	275	24	9%
	1	24	0	24	1%	$34,611	$34,996	47	361	25	7%
	2	24	0	24	1%	$34,580	$34,996	51	363	24	6%
250	**3**	24	0	24	1%	$37,997	$38,496	47	368	24	6%
	4	24	0	24	1%	$41,417	$41,960	47	390	24	6%
	5	24	0	24	1%	$38,054	$38,604	50	402	26	6%
	1	24	0	24	15%	$32,420	$37,919	53	421	45	11%
	2	24	0	24	14%	$32,344	$37,539	56	447	38	9%
300	**3**	24	0	24	13%	$34,726	$39,792	58	447	41	9%
	4	24	0	24	12%	$34,376	$38,996	62	463	42	9%
	5	24	0	24	11%	$37,646	$42,505	62	486	48	10%
	1	24	0	24	54%	$25,124	$54,317	61	607	71	12%
	2	24	0	24	55%	$23,064	$51,783	63	594	58	10%
400	**3**	24	0	24	50%	$26,283	$52,300	62	558	58	10%
	4	24	0	24	46%	$29,962	$55,961	64	584	62	11%
	5	24	0	24	49%	$29,412	$58,200	68	563	57	10%
	1	24	-	-	nfs	$36,302	-	-	-	-	-
	2	24	-	-	nfs	$35,184	-	-	-	-	-
500	**3**	24	-	-	nfs	$38,159	-	-	-	-	-
	4	24	-	-	nfs	$38,013	-	-	-	-	-
	5	24	-	-	nfs	$42,662	-	-	-	-	-

Table C.3: Instance-level results of decomposition-based approach without the DBCs

		Decomposition (w/o DBC)								
		Time(hrs)			FRP	Total	Fleet	# of vehicle movements		
$\|\mathcal{K}\|$	I	FRP	VSP	Total	opt gap	cost	Size	Total	Empty	Empty %
	1	0	0	1	0%	$10,735	16	109	30	28%
	2	0	0	1	0%	$10,008	16	98	29	30%
25	**3**	0	0	1	0%	$10,333	16	112	38	34%
	4	0	0	1	0%	$10,412	15	119	41	34%
	5	0	0	1	0%	$11,047	14	99	34	34%
	1	1	0	1	0%	$15,501	26	206	52	25%
	2	1	0	1	0%	$16,150	25	210	51	24%
50	**3**	1	0	1	0%	$14,899	23	203	49	24%
	4	1	0	1	0%	$15,866	23	212	47	22%
	5	1	0	1	0%	$18,375	22	226	46	20%
	1	1	1	2	0%	$25,819	41	295	66	22%
	2	1	1	2	0%	$27,834	38	309	64	21%
100	**3**	1	1	2	0%	$26,439	35	295	66	22%
	4	1	1	2	0%	$22,264	34	277	68	24%
	5	1	1	2	0%	$26,510	32	274	70	25%
	1	1	1	2	0%	$20,199	42	350	68	19%
	2	1	1	2	0%	$19,817	39	330	65	20%
150	**3**	2	1	2	0%	$19,572	38	314	73	23%
	4	2	1	2	0%	$20,561	37	301	70	23%
	5	2	1	2	0%	$20,972	39	290	66	23%
	1	2	1	3	0%	$33,748	45	414	73	18%
	2	2	1	2	0%	$33,715	46	404	67	17%
200	**3**	2	1	2	0%	$31,121	47	396	66	17%
	4	2	1	2	0%	$31,720	45	376	62	17%
	5	2	1	2	0%	$34,892	50	358	61	17%
	1	2	1	4	0%	$38,846	50	491	60	12%
	2	2	1	3	0%	$38,496	55	491	63	13%
250	**3**	2	1	3	0%	$41,960	60	494	64	13%
	4	2	1	4	0%	$44,898	63	512	64	13%
	5	2	1	3	0%	$42,078	68	531	58	11%
	1	3	1	4	0%	$40,990	65	547	70	13%
	2	3	1	4	0%	$40,993	65	579	67	12%
300	**3**	3	1	4	0%	$42,935	67	575	70	12%
	4	3	1	4	0%	$41,687	61	589	72	12%
	5	3	1	4	0%	$45,183	58	611	80	13%
	1	4	2	5	0%	$45,633	66	559	63	11%
	2	3	2	5	0%	$47,196	65	577	57	10%
400	**3**	4	2	5	0%	$49,694	67	559	59	11%
	4	4	2	6	0%	$46,252	72	534	54	10%
	5	4	2	6	0%	$49,699	71	565	52	9%
	1	6	2	9	0%	$61,762	85	687	66	10%
	2	7	2	9	0%	$63,945	83	687	65	9%
500	**3**	6	2	9	0%	$65,863	81	657	58	9%
	4	7	2	9	0%	$72,450	76	645	64	10%
	5	7	2	10	0%	$68,827	83	618	63	10%

Table C.4: Instance-level results of decomposition-based approach with single-thread search

		Single-Thread										
		Avg total time	Avg #iter	Total cost			Fleet sizes			Best # vehicle movements		
$\|\mathcal{K}\|$	I			Mean	Best	Std	Mean	Best	Std	Total	Empty	Empty %
	1	(hrs)	30	$9,908	$9,820	20	15	14	4	49	7	14%
	2	1	34	$9,493	$9,380	29	12	12	0	49	6	12%
25	**3**	1	31	$9,756	$9,565	22	15	14	3	47	7	14%
	4	1	30	$9,700	$9,613	6	12	12	4	61	11	18%
	5	1	30	$10,485	$10,320	29	16	15	1	47	7	15%
	1	3	34	$14,902	$14,624	9	22	21	2	129	16	13%
	2	3	34	$15,277	$14,992	30	22	21	2	131	18	14%
50	**3**	3	34	$14,150	$14,038	22	21	20	3	108	16	15%
	4	4	34	$15,632	$15,340	12	21	20	2	98	13	13%
	5	3	32	$15,496	$15,267	22	21	20	1	109	13	12%
	1	5	45	$23,214	$23,053	18	30	29	4	239	35	15%
	2	5	38	$24,296	$24,008	6	33	32	5	205	28	14%
100	**3**	5	43	$23,996	$23,805	19	37	35	2	206	18	9%
	4	3	44	$21,207	$20,976	15	34	32	5	239	26	11%
	5	3	35	$24,601	$24,333	9	35	34	4	205	18	9%
	1	4	51	$19,079	$18,965	28	36	35	2	238	30	13%
	2	6	54	$19,025	$18,781	11	37	36	5	301	31	10%
150	**3**	6	54	$18,333	$18,133	28	38	36	2	271	32	12%
	4	5	51	$19,569	$19,223	9	35	34	1	324	33	10%
	5	6	54	$19,770	$19,594	22	36	35	3	296	31	11%
	1	5	61	$32,043	$31,663	12	43	41	3	382	33	9%
	2	7	63	$31,617	$31,211	6	38	37	5	366	27	7%
200	**3**	6	61	$29,818	$29,377	26	42	40	1	365	27	7%
	4	6	64	$30,028	$29,672	10	37	36	5	344	34	10%
	5	6	65	$32,818	$32,655	16	41	39	1	325	34	10%
	1	7	72	$36,234	$35,805	29	50	49	3	412	25	6%
	2	7	74	$36,106	$35,926	19	56	53	1	412	24	6%
250	**3**	10	70	$40,258	$39,468	11	51	49	1	417	27	7%
	4	7	70	$43,358	$42,802	15	51	49	5	441	34	8%
	5	7	71	$40,184	$39,512	17	54	52	3	448	35	8%
	1	9	78	$37,749	$37,561	11	55	54	3	477	32	7%
	2	11	78	$37,926	$37,365	22	57	55	1	475	33	7%
300	**3**	12	70	$39,674	$39,477	15	58	57	1	499	34	7%
	4	13	72	$39,008	$38,814	22	63	63	5	491	35	7%
	5	13	76	$42,747	$42,450	24	63	62	3	533	36	7%
	1	11	70	$42,589	$42,293	28	67	65	0	486	37	8%
	2	13	76	$42,888	$42,255	21	65	63	2	486	38	8%
400	**3**	13	73	$45,142	$44,387	8	64	63	2	511	39	8%
	4	10	74	$49,479	$48,989	8	66	65	3	564	40	7%
	5	13	70	$49,554	$49,258	26	65	65	0	567	41	7%
	1	23	81	$54,445	$54,066	9	76	73	1	623	42	7%
	2	23	81	$57,183	$56,617	21	73	72	1	653	43	7%
500	**3**	20	80	$60,276	$59,620	26	73	72	3	688	44	6%
	4	21	81	$63,879	$63,372	28	74	73	2	731	45	6%
	5	19	84	$70,135	$68,827	8	76	75	1	794	46	6%

Table C.5: Instance-level results of decomposition-based approach with multi-thread search

		Multi-Thread						
		Total time	Avg #iter per thread	Total cost	Fleet Size	# of vehicle movements		
$\|\mathcal{K}\|$	I					Total	Empty	Empty %
	1	1	31	$9,795	14	48	7	7%
	2	1	34	$9,251	14	46	5	5%
25	**3**	1	31	$9,400	12	46	6	7%
	4	1	32	$9,345	13	63	8	3%
	5	1	31	$10,094	14	51	7	9%
	1	3	33	$14,747	20	113	14	4%
	2	3	34	$14,805	21	127	14	4%
50	**3**	3	33	$13,665	20	100	12	3%
	4	4	35	$14,762	19	87	11	2%
	5	3	34	$16,155	20	103	12	3%
	1	5	44	$22,587	32	239	28	3%
	2	5	38	$23,619	32	195	24	2%
100	**3**	5	43	$23,384	31	196	24	2%
	4	3	42	$20,884	35	234	24	1%
	5	4	37	$25,111	31	198	29	4%
	1	5	52	$18,593	32	242	29	2%
	2	6	51	$18,505	38	292	29	2%
150	**3**	6	52	$17,920	35	269	32	3%
	4	5	52	$19,010	36	330	30	2%
	5	6	53	$20,342	33	269	27	1%
	1	6	60	$31,176	60	356	35	2%
	2	7	61	$30,841	33	352	33	1%
200	**3**	6	64	$29,013	34	331	38	3%
	4	6	61	$29,295	35	334	34	2%
	5	6	63	$32,430	32	369	34	2%
	1	8	74	$35,570	41	378	27	2%
	2	8	78	$35,601	52	378	32	3%
250	**3**	11	70	$39,106	53	419	36	4%
	4	7	72	$42,645	52	460	30	2%
	5	8	70	$39,248	52	421	32	3%
	1	9	79	$37,382	50	466	34	7%
	2	11	78	$37,278	59	465	35	7%
300	**3**	12	67	$39,319	60	488	37	7%
	4	13	68	$38,724	59	481	39	7%
	5	13	75	$42,422	62	524	41	7%
	1	11	71	$41,211	60	472	35	8%
	2	13	77	$41,397	60	474	36	8%
400	**3**	12	76	$43,675	62	501	38	8%
	4	10	76	$48,361	66	555	40	8%
	5	14	68	$48,453	66	556	42	8%
	1	21	83	$53,234	66	576	34	8%
	2	23	78	$55,541	69	604	35	7%
500	**3**	20	77	$58,526	71	638	37	7%
	4	21	82	$62,392	74	683	39	7%
	5	18	81	$67,771	79	745	41	7%

Table C.6: Performance of solution approaches w.r.t. the best known solution (BKS) for each instance at different solution time points (hrs). (Bold values show the first time obtaining the best solution)

Method	$\|\mathcal{K}\|$	Total costs at solution times (hrs)						Gap from BKS					
		4	**8**	**12**	**16**	**20**	**24**	**4**	**8**	**12**	**16**	**20**	**24**
IP	**25**	nfs	$27,819	$19,716	$15,823	$14,984	**$9,137**	nfs	68%	54%	43%	40%	2%
	50	nfs	$45,888	$32,089	$26,630	$24,841	**$15,624**	nfs	68%	56%	47%	43%	10%
	100	nfs	nfs	$52,222	$42,319	$38,968	**$23,907**	nfs	nfs	56%	49%	44%	9%
	150	nfs	nfs	$57,422	$46,421	$31,262	**$22,416**	nfs	nfs	67%	61%	42%	19%
	200	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs
	250	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs
	300	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem
	400	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem
	500	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem	mem
Dec. (w/ DBC)	**25**	$12,437	**$8,973**	$8,973	$8,973	$8,973	$8,973	23%	0%	0%	0%	0%	0%
	50	$26,516	$19,022	**$14,069**	$14,069	$14,069	$14,069	44%	22%	0%	0%	0%	0%
	100	$50,323	$36,572	$26,520	**$21,685**	$21,685	$21,685	54%	37%	13%	0%	0%	0%
	150	$41,461	$30,110	$21,100	**$18,221**	$18,221	$18,221	54%	37%	11%	0%	0%	0%
	200	$72,075	$51,299	$37,281	$31,594	**$30,032**	$30,032	57%	40%	18%	3%	0%	0%
	250	nfs	$104,231	$76,137	$63,660	$57,094	**$37,810**	nfs	63%	50%	40%	33%	0%
	300	nfs	nfs	nfs	$65,023	$58,632	**$39,350**	nfs	nfs	nfs	40%	33%	1%
	400	nfs	nfs	nfs	$113,385	$104,118	**$54,512**	nfs	nfs	nfs	61%	57%	18%
	500	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs	nfs
Dec. (w/o DBC)	**25**	**$10,507**	$10,507	$10,507	$10,507	$10,507	$10,507	9%	15%	15%	15%	15%	15%
	50	**$16,158**	$16,158	$16,158	$16,158	$16,158	$16,158	8%	8%	13%	13%	13%	13%
	100	**$25,773**	$25,773	$25,773	$25,773	$25,773	$25,773	10%	10%	10%	16%	16%	16%
	150	**$20,224**	$20,224	$20,224	$20,224	$20,224	$20,224	6%	7%	7%	10%	10%	10%
	200	**$33,039**	$33,039	$33,039	$33,039	$33,039	$33,039	5%	8%	8%	8%	9%	9%
	250	**$41,256**	$41,256	$41,256	$41,256	$41,256	$41,256	0%	6%	7%	7%	7%	8%
	300	$59,682	**$42,358**	$42,358	$42,358	$42,358	$42,358	27%	5%	8%	8%	8%	8%
	400	$65,056	**$47,695**	$47,695	$47,695	$47,695	$47,695	24%	2%	5%	6%	6%	6%
	500	$125,121	$91,866	**$66,570**	$66,570	$66,570	$66,570	43%	31%	7%	10%	11%	11%
Single-Thread	**25**	**$9,740**	$9,740	$9,740	$9,740	$9,740	$9,740	2%	8%	8%	8%	8%	8%
	50	**$14,852**	$14,852	$14,852	$14,852	$14,852	$14,852	0%	0%	5%	5%	5%	5%
	100	$23,697	**$23,235**	$23,235	$23,235	$23,235	$23,235	2%	1%	1%	7%	7%	7%
	150	$19,362	**$18,939**	$18,939	$18,939	$18,939	$18,939	2%	0%	0%	4%	4%	4%
	200	$32,065	**$30,915**	$30,915	$30,915	$30,915	$30,915	2%	1%	1%	1%	3%	3%
	250	$43,764	**$38,703**	$38,703	$38,703	$38,703	$38,703	6%	0%	1%	1%	1%	2%
	300	$43,504	$41,891	**$39,133**	$39,133	$39,133	$39,133	0%	4%	0%	0%	0%	0%
	400	$51,682	$47,568	$46,757	**$45,436**	$45,436	$45,436	4%	2%	3%	2%	2%	2%
	500	$72,199	$65,170	$61,765	$60,274	**$60,501**	$60,501	1%	3%	0%	1%	2%	2%
Multi-Thread	**25**	**$9,577**	$9,577	$9,577	$9,577	$9,577	$9,577	0%	6%	6%	6%	6%	6%
	50	**$14,827**	$14,827	$14,827	$14,827	$14,827	$14,827	0%	0%	5%	5%	5%	5%
	100	$23,232	**$23,117**	$23,117	$23,117	$23,117	$23,117	0%	0%	0%	6%	6%	6%
	150	$18,926	**$18,874**	$18,874	$18,874	$18,874	$18,874	0%	0%	0%	3%	3%	3%
	200	$31,313	**$30,551**	$30,551	$30,551	$30,551	$30,551	0%	0%	0%	0%	2%	2%
	250	$42,243	$39,127	**$38,434**	$38,434	$38,434	$38,434	2%	1%	0%	0%	0%	2%
	300	$43,943	$40,279	**$39,025**	$39,025	$39,025	$39,025	1%	0%	0%	0%	0%	0%
	400	$49,694	$46,614	$45,132	$44,658	**$44,620**	$44,620	0%	0%	0%	0%	0%	0%
	500	$71,413	$63,442	$61,642	$59,914	$59,499	**$59,493**	0%	0%	0%	0%	0%	0%