

TACTICAL DESIGN OF LAST MILE LOGISTICAL SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Alexander M. Stroh

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Operations Research
School of Industrial and Systems Engineering

Georgia Institute of Technology

August 2021

Copyright © Alexander M. Stroh 2021

TACTICAL DESIGN OF LAST MILE LOGISTICAL SYSTEMS

Thesis committee:

Dr. Alan Erera
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Dr. Joel Sokol
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Dr. Alejandro Toriello
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Dr. John Carlsson
School of Industrial and Systems
Engineering
University of Southern California

Dr. Martin Savelsbergh
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Date approved: July 21, 2021

All models are wrong, but some are useful.

George Box

For my father, Patrick J. Stroh

ACKNOWLEDGMENTS

I am extremely grateful to have been able to work with Dr. Alejandro Toriello and Dr. Alan Erera during my time at Georgia Tech. Those late morning discussions in front of the whiteboard were some of my most fond memories of graduate school. They were always available for a healthy academic discussion and we left no variable unturned. I am thankful for how eagerly they brought me along on projects, allowed me to experiment with new ideas, and really tailored my graduate experience to what I needed. They kept me honest, inquisitive, and challenged me to grow. It was a pleasure being their student.

I would also like to thank the other members of my committee as they have all individually contributed to my success as an academic. Dr. Martin Savelsbergh continues to be an inspiration for me and all who strive to make an impact in the world of Operations Research. I am grateful that for my first year at Georgia Tech, Dr. Joel Sokol was my mentor and really helped me transition into becoming a successful researcher. Lastly, Dr. John Carlsson's passion for his work has been infectious for many of us and as he started me on this journey I am glad he could be here to help finish it.

I would like to give a special acknowledgement to my father, Patrick Stroh. My dad has been my north star for my entire life and I am extremely proud to be his son. He was truly a Renaissance man, and everyone he crossed paths with were all the better for having met him. I continue to strive to live my life as he lived his.

Lastly and most importantly, I want to thank the person whom I share this doctorate with. My wife Bridget has been by my side semester after semester, coffee cup after coffee cup, sacrificing her time, energy, and granting me all the patience in the world. Thank you for always being my number one superfan, through the good and the bad days.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	xi
List of Figures	xii
Summary	xiv
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Contributions	4
1.3.1 Chapter 2: Tactical Design of Same-Day Delivery Systems	4
1.3.2 Chapter 3: Time-Varying Service Regions in Same-Day Delivery	5
1.3.3 Chapter 4: Cycle Time Considerations for Capacitated Vehicle Routing Problems	6
Chapter 2: Tactical Design of Same-Day Delivery Systems	7
2.1 Introduction	7
2.1.1 Literature Review	10
2.2 Model Formulation	13

2.3	Optimal Policies	15
2.3.1	Many Vehicles	16
2.3.2	One Vehicle	17
2.3.3	General Fleet Size	22
2.4	Model Applications	24
2.4.1	Serving the Entire Region versus Partitioning	25
2.4.2	Orders at the Start of the Service Day	26
2.4.3	Capacitated Vehicles	28
2.4.4	Choosing Order Cutoff Time	29
2.5	Computational Study	32
2.5.1	Many-Vehicle Policy	35
2.5.2	Single-Vehicle Policy	37
2.5.3	Many Capacitated Vehicles	39
2.6	Conclusions	41
Chapter 3: Time-Varying Service Regions in Same-Day Delivery		43
3.1	Introduction	43
3.1.1	Literature Review	46
3.2	Model Formulation	48
3.3	One vehicle, one dispatch systems	50
3.4	Multiple vehicles, one dispatch each	53
3.4.1	Model formulation	53
3.4.2	Model analysis and tactical design properties	55

3.4.3	Value of varying service regions	58
3.4.4	Impacts of constrained service regions	63
3.5	One vehicle, multiple dispatch case	64
3.5.1	Model formulation	64
3.5.2	Model analysis and tactical design properties	69
3.5.3	Value of varying service regions	70
3.6	Computational Experiments	72
3.6.1	Two vehicles, one dispatch each simulation	74
3.6.2	One vehicle, two dispatches simulation	77
3.7	Conclusions	79
 Chapter 4: Cycle Time Considerations for Capacitated Vehicle Routing Problems		81
4.1	Introduction	81
4.1.1	Literature Review	85
4.2	Problem Description and Formulation	90
4.3	Model Properties and Observations	95
4.4	Column Generation Techniques	99
4.5	Computational Examples	104
4.5.1	Phase 1 Optimization	105
4.5.2	Phase 2 Optimization	106
4.5.3	Column Generation Applications	109
4.6	Conclusions	113
 Chapter 5: Conclusion		115

5.1	Summary	115
5.2	Conclusion	117
5.3	Future Works	117
Appendices		118
Chapter A: Technical Supplementation for Chapter 2		119
A.1	Proof of Theorem 1	119
A.2	Proof of Lemma 3	122
A.3	Proof of Theorem 4	123
A.4	Proof of Corollary 5	125
A.5	Proof of Theorem 6	126
A.6	Proof of optimality for the Generalized MVP	128
A.7	Proof of Theorem 8	129
A.8	Proof of Proposition 9	129
A.9	<i>A Posteriori</i> Formulations	130
Chapter B: Technical Supplementation for Chapter 3		132
B.1	Proof of Property 10	132
B.2	Proof of Property 11	132
B.3	Proof of Property 12	133
B.4	Proof of Theorem 13	133
B.5	Proof of Property 14	135
B.6	Proof of Property 15	137

B.7 Proof of Property 16	138
B.8 Proof of Property 17	139
B.9 Proof of Theorem 18	140
B.10 Proof of Property 19	147
B.11 Proof of Property 20	147
B.12 Proof of Theorem 21	148
Chapter C: Technical Supplementation for Chapter 4	150
C.1 Proof of Theorem 23	150
C.2 Proof of Theorem 24	150
C.3 Details regarding the example illustrated in Figure 4.1	151
C.4 Details regarding the generated inputs in Section 4.5	152
References	153
Vita	161

LIST OF TABLES

2.1	Summary of tactical SDD optimality results.	15
2.2	Computational study results, many-vehicle policy.	36
2.3	Computational study results, single-vehicle policy.	39
2.4	Computational study results, capacitated vehicles.	41
3.1	Optimal dispatching policies, <i>mv1d</i> model.	57
3.2	Optimal dispatching policies, fixed-area <i>mv1d</i> model.	61
3.3	Optimal dispatching policies, <i>1vDd</i> model.	68
3.4	Optimal dispatching policies, fixed-area <i>1vDd</i> model.	71
3.5	Computational study results, realistic <i>2v1d</i> design.	76
3.6	Computational study results, realistic fixed-area <i>2v1d</i> design.	76
3.7	Computational study results, realistic <i>1v1d</i> design.	78
3.8	Computational study results, realistic <i>1v2d</i> design.	79
4.1	Computational study results, column generation for CTC-CVRPs.	112
C.1	Supporting information for Figure 4.1.	152

LIST OF FIGURES

2.1	Visual representation of an MVP requiring four vehicles.	17
2.2	Visual representation of optimal dispatch policy for Example 2.	22
2.3	Comparing the MVP with three vehicles to a hybrid policy with two.	24
2.4	Profit function with respect to cutoff time, many-vehicle case	32
2.5	Profit function with respect to cutoff time, single-vehicle case	32
2.6	Computational study, Northeast Atlanta service region.	33
3.1	Quantity served versus accumulation time, 1v1d system.	52
3.2	An illustration of a 3v1d policy.	53
3.3	Quantity served versus fixed service area size, fixed-area <i>mv1d</i> variant.	62
3.4	An illustration of a 1v2d policy.	65
3.5	Calibrating the routing function constant.	74
4.1	An illustration of dominated route permutations.	99
4.2	Computational study results, Phase 1 optimization.	106
4.3	Computational study results, Phase 2 optimization.	107
4.4	Computational study results, average cycle time versus fleet size.	108
4.5	Computational study results, direct routes versus fleet size.	109
A.1	An illustration of shifting volumes between consecutive dispatches.	121

A.2 Proving Theorem 6, hybrid policy illustration.	127
A.3 Proving Theorem 6, MVP illustration.	128

SUMMARY

This dissertation consists of three distinct logistical topics, unified by a focus on the intelligent design of last mile logistical systems at a tactical level. The three design problems all arise within package delivery supply chains, though the mathematical models and solution techniques developed in these studies can be applied to other logistics systems. We propose models that do not attempt to capture granular minute by minute operational decision making, but rather, system behavior on average so that we may approximate the impact of various design choices.

In Chapter 2, we study tactical models for the design of same-day delivery (SDD) systems. While previous literature includes operational models to study SDD, they tend to be detailed, complex, and computationally difficult to solve. Thus, such models may not provide any insight into tactical SDD design variables and their impact on the average performance of the system. We propose a simplified vehicle dispatching model that captures the average behavior of an SDD system from a single depot location by utilizing continuous approximation techniques. We analyze the structure of vehicle dispatching policies given by our model for various families of problem instances and develop techniques to find optimal dispatching policies that require only simple computations. Our models can help answer various tactical design questions including how to select a fleet size, determine an order cutoff time, and combine SDD and overnight order delivery operations.

In Chapter 3, we study the tactical optimization of SDD systems under the assumption that service regions are allowed to vary over the course of each day. In most existing studies of last mile logistics problems, service regions are assumed to be static. Service regions which are designed too small or cutoff SDD availability too soon may potentially lose SDD market share, while regions which are designed too large or accept orders too late may result in costly operations or failed deliveries, resulting in a loss of customer goodwill. We use a continuous approximation approach to capture average system behavior

and derive optimal dynamic service region areas and tactical vehicle dispatching policies which maximize the expected number of SDD orders served per day. Furthermore, we compare such designs to fixed service region designs or capacitated service region designs.

In Chapter 4, we introduce the concept of cycle time considering capacitated vehicle routing problems, which are motivated by the desire to decrease the average time packages spent within a delivery network. Traditional vehicle routing models focus on the resource usage of the system whereas our models instead consider the impact of routing policies on the units being served. We explicitly consider pre-routing waiting times at a depot, total demand-weighted accumulated routing times, vehicle capacity constraints, and designing repeatable delivery routes in our models. We present two set partitioning formulations for such problems and derive efficient solution techniques so that the impact of various design parameters can be assessed.

CHAPTER 1

INTRODUCTION

1.1 Motivation

As a population, we have become increasingly reliant on the e-commerce industry. In 2016, the United States saw year over year annual retail sales grow by an estimated 2.76%, while electronic shopping by itself grew by 12.40% [1]. More recently, changes in travel behavior due to the COVID-19 pandemic [2, 3] have accelerated this growth even further: total e-retail volume in the U.S. between April 2020 and March 2021 surpassed \$817 billion, representing an increase of over 30% from the prior year [4, 5]. Not only does the demand for electronic shopping continue to grow at staggering rates, the last mile fulfillment systems which support this growth also continue to progress.

In the last decade, we have started to see a rapid movement into the next-day and same-day delivery (SDD) space. As a service offering, SDD allows e-commerce firms to directly compete with brick-and-mortar retail by providing the customer with near-instant gratification. Amazon, one of the current SDD industry leaders, began offering an SDD fulfillment option in October 2009 across seven major U.S. cities [6]. By November 2018, Amazon had grown to offering the service to over 8,000 cities and towns [7]. Another recent trend in last mile supply chains is for service providers to consider more service oriented constraints and objectives such as fairness considerations for service or decreasing average time to receive services. A recent survey suggests that 82% of companies believe that customer retention is cheaper than customer acquisition [8] and a study from the Harvard Business School expresses that increasing customer retention rates by 5% can increase profits by over 25% [9]. Overall, both retailers and the customers continue to evolve their mindset regarding last mile supply chains.

While industry leaders in e-commerce and consumers have started to pay more attention to what is offered in the last mile space, academics have begun to study so-called “operational” models which attempt to aid in the day to day decision making of a business for these systems. For example, managing an SDD system is potentially problematic for retailers with already thin profit margins. SDD systems inherit and exacerbate many of the issues faced by more traditional two-day or next-day last mile logistics systems, including tight deadlines, low order volumes, and a high level of order variability and dynamism. It is not surprising that the logistics research community over the past few years had proposed and studied operational policies for distribution systems using a variety of models and assumptions, e.g. [10, 11, 12, 13, 14, 15, 16]. These works seek to optimize day-to-day operations, including vehicle routing and order acceptance mechanisms. While these studies are paramount to understanding how to efficiently manage a pre-defined system, they do not focus on optimally designing aspects of the system topology itself.

In contrast, the logistics research community has not focused its analysis on the tactical design decisions (i.e., regarding decisions which are made and implemented every few weeks or months) important for last-mile distribution: How large should the vehicle fleet be? How late in the day should services be offered to customers? How large should the service area be? Is there a more customer-centric method for pricing a delivery tour rather by cost or completion time alone? To our knowledge, no papers in the literature address these and other important questions.

1.2 Objectives

The primary focus of this dissertation is to aid in the tactical level decision making that goes into designing and developing of various last mile logistical systems. Our research objectives are as follows:

- (1) To produce models of SDD systems which capture system behavior on an aggregate level.

- (2) To illustrate that our SDD models retain a high level of fidelity, so that they be used for their intended purposes.
- (3) To study the effects of various tactical design parameters on SDD systems.
- (4) To show value in a novel service-orientated last mile system, where the average time packages spend in the network is minimized.
- (5) To model such service-orientated systems so that their design considerations can be easily compared and contrasted.

In this first chapter we started by motivating the need for intelligently designed last mile logistics systems. Statistics from general e-commerce growth to specific fulfillment option expansion were given and discussed. The primary research objectives of the dissertation were stated, which are now followed by the specific chapter by chapter contributions which work towards the completion of these objectives.

Chapter 2 introduces the concept of continuous approximations in SDD modeling in order to simplify various operational decisions. While some design parameters remain fixed for the specific use-cases presented in the chapter, the use of such approximations for SDD modeling are to our understanding the first of their kind. The chapter focuses on a few families problem instances and solves the associated models to optimality via underlying structured dispatching policies. Such results lead to readily accessible system design comparisons. A computational section demonstrates the power and accuracy of such models.

Chapter 3 dives deeper into the modeling of SDD system design by studying the effects of time-varying service regions and variable order cutoff times. Results of the previous chapter are leveraged into the tactical modeling of these systems, which is most useful to simplify the otherwise complicated granular decision making that such systems require. The pros and cons of being able to dynamically change a service region throughout a day are studied. Various other consequences from our models are observed and discussed.

Chapter 4 introduces a novel objective function for evaluating repeatable dispatches for vehicle routing problems. Unlike existing objectives and constraints which are resource or cost minimizing, the systems presented in this chapter are concerned with the effects on the delivery units themselves. The primary motivation for this work is to minimize the total time units spend in a delivery network. The associated models are formulated with tactical design in mind, as they can be readily compared and contrasted for a variety of design considerations.

Chapter 5 finishes the dissertation with a summary of contributions and concluding remarks, including possible future research opportunities.

1.3 Contributions

1.3.1 Chapter 2: Tactical Design of Same-Day Delivery Systems

We consider the following to be our primary contributions from Chapter 2:

- (1) We propose a simple model for SDD dispatching that captures aggregate SDD system behavior by leveraging the elegant structure of continuous approximations. To our knowledge, this is the first such use of this methodology in SDD applications.
- (2) We use the dispatching model to analyze two important operational cases for SDD fleets. The first case is when a single vehicle is assigned to a service area and is dispatched multiple times during the operating day, and the second case is when the fleet is large enough that each vehicle is dispatched once per day. We characterize the structure of optimal dispatching policies for these two cases using our model, and show that the optimal policies can be determined using very simple computational techniques, such as finding the roots of equations with single unknowns. Although the case in which multiple vehicles must make multiple dispatches is significantly more difficult to optimize, we propose a heuristic policy for this case with a worst-case performance guarantee.

- (3) We use the simple dispatching approximation model and the optimal policies that result to answer various tactical system design questions, including fleet sizing, length of service window, and whether SDD orders should be combined with overnight orders. In all cases, our conclusions rely on simple and transparent analytical properties. The model predictions are validated in a computational study against a detailed operational model using realistic data.

1.3.2 Chapter 3: Time-Varying Service Regions in Same-Day Delivery

We consider the following to be our primary contributions from Chapter 3:

- (1) Using continuous approximations on order arrivals and vehicle routing times, we propose a mathematical optimization model for maximizing order quantity served when the service region is allowed to vary between vehicle dispatches. The decision space for the model includes choosing the order accumulation time between successive dispatches as well as determining the size of time-varying service regions from which orders accrue.
- (2) We isolate and perform an in-depth theoretical analysis for a few important SDD system variations. Specifically, we study a setting in which multiple vehicles each dispatch once per day to analyze the marginal benefits of increasing fleet size. We also study a setting in which one vehicle dispatches multiple times per day to analyze the marginal benefits of re-using a particular vehicle for multiple dispatches. We use our theoretical results to design efficient solution procedures.
- (3) We study the quantifiable effects of allowing time-varying service regions compared to traditional designs which use a fixed service region.
- (4) We conduct various computational experiments using a set of realistic modeling parameters to better understand the implications of our tactical design model for retailers in a realistic setting.

1.3.3 Chapter 4: Cycle Time Considerations for Capacitated Vehicle Routing Problems

We consider the following to be our primary contributions from Chapter 4:

- (1) We formulate two novel VRPs with cycle time considerations. We focus our efforts on package delivery logistics stemming from a single depot node leveraging a fleet of capacitated service vehicles. Using a set-partitioning formulation we construct our so-called “Phase 1” problem, where we determine the minimal fleet size necessary to satisfy package demand over time. Then we formulate our “Phase 2” model which minimizes package cycle time dependent on a given fleet size parameter.
- (2) We prove and subsequently discuss a variety of key theoretical insights into our models. Two of our derived theorems quantify the impact of fleet sizing on any given routing tour, while a third property analyzes the importance of pre-processing varying permutations of route stops for a given subset of demand nodes.
- (3) We discuss a methodology for column generation for our CTC-CVRPs. Such procedures can be used to derive LP relaxation bounds, in the optimal solving of IP solutions via branch-price-and-cut methods, and for heuristical improvements on top of existing IP solutions.
- (4) Lastly, we illustrate some of our findings with a set of computational exercises in an attempt to demonstrate the utility and flexibility of our novel models.

CHAPTER 2

TACTICAL DESIGN OF SAME-DAY DELIVERY SYSTEMS

2.1 Introduction

Total annual retail sales in the United States grew by an estimated 2.76% from 2015 to 2016, in part via electronic shopping, which increased by 12.40% [1]. Within the growing space of e-commerce, same-day delivery (SDD) services are commonly offered by large retailers and logistics providers. A survey of over 500 North American retailers found that 51% claimed to provide SDD fulfillment options in 2017, up from the 16% reported in 2016 [17]. Amazon, one of the current SDD industry leaders, began offering an SDD fulfillment option in October 2009 across seven major U.S. cities [6]. By April 2016, Amazon offered the service to over 1,000 cities and towns, a figure which rose to over 8,000 by November 2018 [18, 7]. These statistics highlight recent demand increases in the e-commerce space, as well as the adoption of SDD systems by many retailers.

Managing an SDD system is potentially problematic for retailers with already thin profit margins. SDD systems inherit and exacerbate many of the issues faced by more traditional two-day or next-day last mile logistics systems, including tight deadlines, low order volumes, and a high level of order variability and dynamism; in general, the uncertainty increases and the time to react decreases [11, 12]. Therefore, dispatching and routing orders may be costly and inefficient if not planned carefully.

Like more traditional e-retail delivery services, SDD requires two core logistics processes: order management at the stocking location, including receiving, picking, and packing; and order distribution from the stocking location to customer delivery addresses. We focus here on the second of these processes. Order distribution requires operational decisions, such as when to dispatch a delivery vehicle (timing), and which subset of awaiting

customers it will serve (composition). There are clear trade-offs between the timing and composition of SDD dispatches. In some cases, it may be best for a delivery vehicle to wait as long as possible at the stocking location, allowing the accumulation of orders and greater routing efficiency upon dispatch. Alternatively, shorter, more time-inefficient trips could be made in order to reduce the workload within the system, leaving more flexibility to serve orders later in the service day. Such decisions are made numerous times, across a fleet of vehicles, during each service day. Even when the orders to be loaded onto each vehicle for a dispatch are known, deciding the sequence in which to visit delivery locations is a traveling salesman problem (TSP) with possible side constraints; the problem of order assignment and routing in e-commerce has only recently attracted attention from the research community, e.g. [19]. Additionally for SDD, retailers often constrain themselves to serve all SDD demand in a given service day, and thus restrict the latest possible time SDD orders can be placed [20, 21]. These order cutoff times can be static or determined dynamically.

Over the past few years, the logistics research community has proposed and studied operational policies for SDD distribution systems using a variety of models and assumptions, e.g. [10, 11, 12, 15, 16]. The models considered in the literature to date typically assume a fixed SDD system design, including service area, delivery vehicle fleet size, service time window, etc., and then perform a detailed analysis, optimization and/or simulation of operating policies.

In contrast, the logistics research community has not focused its analysis on the tactical design decisions important for SDD distribution: How large should the SDD delivery vehicle fleet be? How late in the day should SDD service be offered to customers? How large should the service area be? To our knowledge, no papers in the literature address these and other important questions, and our goal is to offer a first attempt. While detailed operational models can in principle offer some insights about such tactical decisions, their granularity implies significant complexity, which in turn renders tactical analysis difficult

and less transparent – the models have “too many moving parts”. One goal of this chapter is to develop simplified models of operational decisions while maintaining fidelity at the aggregate level; we propose models that do not attempt to capture each order realization and operational decision, but rather to capture the system behavior “on average” so that we may approximate the impact of various design choices on day-to-day operations.

We develop a distribution modeling approach for a single SDD stocking location or dispatch facility, where orders are packed for last-mile delivery and dispatched on delivery vehicles. As in most e-retail settings, we assume orders are customer-specific and cannot be packed or dispatched preemptively, but rather only after they are placed. We also assume a common delivery deadline, e.g. the end of the business day, rather than order-specific deadlines more common in food delivery services [22]. Since SDD systems face tight delivery deadlines and comparatively low order volume, time (not vehicle capacity) tends to be the limiting resource and one of the primary constraints in our models. We initially assume uncapacitated vehicles, and later show that our results extend to the capacitated case with only slight modifications.

To build a simplified SDD dispatch model that still accurately captures system performance, we use a continuous approximation approach in which the expected durations of vehicle routing tours are approximated using a concave, increasing function of the number of orders served. The use of such approximations is well established in logistics [23], with some canonical results dating back several decades [24, 25, 26]. When order locations are randomly distributed in the service region according to a continuous distribution, continuous approximations are known to be quite accurate. Such approximations have recently been successfully applied in a last-mile operational context [14], and can also be calibrated with empirical observations (see, e.g., [27]). We provide our own computational validation of the approximation model and dispatching policies we develop, and we show that they are remarkably accurate when compared to much more detailed operational models.

Furthermore, although our model is motivated by SDD, our main results rely only on

the concavity and monotonicity of the expected dispatch time required to serve a number of orders. Our model could therefore also have applications in other areas where expected processing time is concave and increasing, including batched queueing and warehousing systems, see e.g. [28].

The remainder of the chapter is organized as follows. Section 2.1 concludes with a literature review. In Section 2.2 we formulate a continuous approximation model of vehicle dispatch operations from a single stocking location and justify our model assumptions. We then describe optimal dispatch policies for specific instances of the proposed model in Section 2.3. Section 2.4 provides a managerial analysis of tactical SDD system design using our model and its solutions. We detail a realistic computational experiment using the model and policies in Section 2.5, and conclude in Section 2.6. An appendix contains material omitted from the main body.

2.1.1 Literature Review

SDD models can be classified within the rich family of vehicle routing problems (VRPs). The defining features of an SDD model include stochastic order arrivals, order cutoff times and/or a delivery deadline, and perhaps most salient, the overlap in time of dispatching and order arrivals. Examples of model objectives are maximizing expected orders served in a service day, minimizing penalties from undelivered orders, and/or minimizing total routing distance or time given that most or all orders are served. For these reasons, we reference the VRP with probabilistic customer arrivals as studied in [29, 30, 31]. Additionally, dynamic vehicle routing problems such as [32, 13] broadly encompass SDD modeling.

We now survey some operational SDD models from the literature. One such problem is the dynamic dispatch waves problem (DDWP) [11, 12]. The DDWP discretizes the dispatch decision epochs, or “waves”, for an operator managing an SDD vehicle fleet. Customer orders arrive according to a known stochastic process and must be served by the end of the service day, or the operator will receive an order-based penalty. Additionally,

all delivery vehicles are constrained to return back to the depot before the end of the service day. The objective of the DDWP is to minimize the sum of expected routing cost and penalty cost. In [12] a deterministic variant of the DDWP is solved, where order arrival locations and times are known exactly, over a 1-dimensional service region using the optimal policy structure found in a dynamic programming formulation. In [11] the same authors model the DDWP in 2-dimensions using an integer-programming approach to solve a deterministic variant. Using these deterministic solutions, the authors compute *a priori* dispatch policies for stochastic variants. These policies are further expanded to dynamic policies in their respective papers. In [12], optimal dispatch policies for an SDD variant (one-dimensional service region, one vehicle) were found to have the property that once the first dispatch occurred, the vehicle never waited at the depot again. Additionally, the durations of successive dispatches are decreasing.

Another SDD model found in literature is the same-day delivery problem for online purchases (SDDP) [16]. The authors provide a general framework for SDD modeling, using a fleet of delivery vehicles of known size, a fixed cutoff time for SDD orders, a known arrival rate and distribution of orders, and a service time and a delivery time window on each order. Like the DDWP, all vehicles operate from a single depot. Unlike the DDWP, the objective of the SDDP is to maximize the expected number of SDD requests that are fulfilled in a service day. A model of the SDDP as a Markov decision process (MDP) is proposed, and dispatch policies are found via a sample-scenario approach with orienteering subproblems. The authors discuss delaying delivery vehicles at the depot for as long as possible without violating delivery time-window constraints or altering vehicle return times. Such properties were shown to exist in optimal dispatching solutions, allowing restriction of the search space.

The DDWP and the SDDP, as well as other works in the literature [33, 34, 35, 15], tend to study the daily operations of SDD logistics, while assuming implicit or explicit knowledge of tactical system design features. It is possible to use these more complicated models

to gain tactical-level insight. In [33] the author performs an analysis of the relationship between fleet size and delivery capacity of the system. The authors in [16] observe how the number of fulfilled orders can increase with fleet size by re-solving their operational SDD model multiple times. Such procedures can be useful for determining managerial decisions. However, they require repeatedly solving complex models, often with heuristic methods that do not guarantee optimal solutions; in contrast, our approach will be to exactly optimize a simplified approximation model.

We use continuous approximations to model an SDD system. Specifically, we assume orders arrive at a constant rate, and allow a non-integer number of orders to be served in a dispatch. We approximate the expected time of a dispatch's duration using a non-decreasing, concave function of the number of orders served; a typical example would be the square root of the number of orders, scaled by an appropriate constant [24]. For a recent survey on continuous approximation models in freight logistics, see [23]. The use of such approximations in the field goes back to the BHH theorem [24], a formula for the expected length of a TSP tour as a function of the number of stops visited when locations are drawn from a continuous distribution over the service region. [25, 26] then expanded upon this approximation in an analysis of vehicle routing problems with specified dispatch depots for logistics distribution and collection problems, and studied how different zone shapes affect tour lengths and how to select best zone shapes. The work in [27] considers similar approximation ideas for the Held-Karp TSP bound, and [36] calculates empirical constants for TSP length as a function of the number of stops in a tour; see also [37]. Although our study is the first to apply continuous approximations in SDD, there are other applications of these techniques in urban logistics. For example, the work in [38] considers the efficiency of urban commercial vehicles using continuous approximations, and the techniques are used to study drones in last-mile delivery in [39]. Furthermore, a continuous approximation approach is used in [40] to partition a service region for vehicle routing. Operational models for urban last-mile delivery are considered in [14] and continuous approximations

are deployed within an approximate dynamic programming framework for optimization. Continuous approximations are frequently applied in facility location; see [41] for a recent example with applications in SDD and the last mile.

2.2 Model Formulation

We consider an SDD system in which a single depot serves as a stocking location, and its vehicle fleet serves all orders placed from a defined service region. Orders accumulate throughout the service region over the course of the day, and the dispatcher must ensure that all orders are served by the end of the service day at minimum cost. We next formally define our problem by describing the relevant notation and model elements.

Service Day: The first time any vehicle can leave the depot is time $t = 0$, and the end of the service day is $t = T$. For convenience, we refer to the service day as having T units of time.

Customer Orders and Geography: Demand for SDD in the service region continuously accumulates over time at a constant rate of λ orders per time unit. This demand is served by a fleet of m vehicles, each departing from the single depot. For convenience, we assume without loss of generality that $\lambda = 1$ and all other parameters are appropriately scaled.

Order Cutoff Time: Customer orders become ready for dispatch starting at time $t = 0$ and continue until time $t = N < T$ in the service day. The service day begins with no orders requiring delivery, and thus the total number of orders that accumulate over the day is $\lambda N = N$. Depending on the context, we may refer to N as the order cutoff time or the number of orders to serve. We also discuss the problem extension when an initial set of orders is ready at the start of the service day.

Vehicle Restrictions: All vehicles must return to the depot by time $t = T$. We do not initially constrain the capacity of any vehicle, nor do we restrict any vehicle to carry

an integer number of orders. Vehicles may be dispatched more than once during the service day. We later discuss the model extension with capacitated vehicles.

Dispatch Time Function: The time it takes for a vehicle to serve $n \in \mathbb{R}_{\geq 0}$ orders and return to the depot is given by a function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ that is concave and increasing. This property indicates that serving more orders should take more time, but that there will be a gain in marginal time efficiencies when consolidating orders. An example dispatch time function is of the form $f(n) = a + bn + c\sqrt{n}$. This function includes a constant setup time at the depot, a service time per order, and a BHH routing time [24] between orders; [27] showed computationally that these approximations work well in practice even for small order numbers, assuming order locations are independently drawn from a continuous distribution over the service region.

The objective of our model is to choose a set of feasible dispatches that serves all of the orders while minimizing total dispatch time incurred by all vehicles. We define the d -th dispatch as a tuple (t_d, q_d, i_d) , where t_d indicates the time when vehicle i_d leaves the depot with an order quantity, q_d . We assume without loss of generality that dispatches are ordered by time of dispatch, then if necessary by vehicle index. A set of dispatches $\{(t_d, q_d, i_d)\}_{d=1}^D$ is feasible for our model if the following conditions are satisfied:

$$\sum_{d=1}^D q_d = N, \tag{2.1a}$$

$$q_d \geq 0 \quad \forall d, \tag{2.1b}$$

$$t_d + f(q_d) \leq T \quad \forall d, \tag{2.1c}$$

$$t_d + f(q_d) \leq t_\delta \quad \forall d, \delta \text{ s.t. } i_d = i_\delta, d < \delta, \tag{2.1d}$$

$$t_d \geq 0 \quad \forall d, \tag{2.1e}$$

$$\sum_{\delta=1}^d q_\delta \leq t_d \quad \forall d. \tag{2.1f}$$

Our problem is to choose dispatches that minimize $\sum_{d=1}^D f(q_d)$, subject to (2.1a)-(2.1f),

over all $D \geq 1$. Constraints (2.1a)-(2.1b) guarantee that all dispatches serve a non-negative order number, and that these sum to the total number of orders. Constraint (2.1c) requires all vehicles to return to the depot by the end of the service day. Constraint (2.1d) guarantees that any vehicle performs one dispatch at a time. Constraint (2.1e) ensures the vehicles dispatch only after the service day begins. Finally, constraint (2.1f) guarantees that orders are served only after they realize.

2.3 Optimal Policies

We initially focus on two important families of instances of our SDD model, the many-vehicle case and the single-vehicle case. In the former, we assume any number of vehicles can be added to the delivery fleet at negligible cost; this situation applies, for example, when we can allocate vehicles for SDD from other resources. In the latter, we focus on the simplest case in which the fleet is constrained, as this case is already of significant interest operationally [11, 12]. For the general case with a finite fleet greater than one, we leverage these results to construct a heuristic dispatching policy that combines the policies used at the two extremes. Table 2.1 summarizes results in this section, including the type of result we obtain and additional conditions required for the result to hold.

Table 2.1: Summary of Section 2.3 results.

Section	Fleet	Result	Additional Conditions?
§2.3.1	unlimited	optimal policy	no
§2.3.2	single vehicle	optimal policy	sufficient processing speed sufficient gap time minimum dispatch size
§2.3.3	finite	heuristic policy with approximation guarantee	single-vehicle conditions $f(n) = bn + c\sqrt{n}$

2.3.1 Many Vehicles

We first assume the fleet consists of as many vehicles as we like. Consider the following dispatch policy, and the accompanying theorem.

Many-Vehicle Policy (MVP) Starting at time $t = 0$, dispatch a delivery vehicle at the moment when it can take all of the realized orders waiting at the depot and return at exactly time $t = T$. Repeat this process until only a single vehicle is needed to deliver all remaining orders and return to the depot before the end of the service day. Dispatch this last vehicle at time $t = N$.

Theorem 1. *The MVP is an optimal dispatch policy for the many-vehicle case. Furthermore, if the number of vehicles used by MVP is m^* , the total dispatch time used by this policy provides a lower bound for the model objective with fleet size $m < m^*$.*

Proof. See Appendix A.1. □

The optimal times given by the policy are easy to solve for in practice, since all that is required is to solve equations of the form, $t + f(t) = C$, for different values of C . If more than one vehicle is required, then $t_1 + f(t_1) = T$, and if more than two vehicles are required then $t_2 + f(t_2 - t_1) = T$, and so on. For problems with realistic parameters, computational results show that the MVP will often require a reasonably sized fleet. Figure 2.1 depicts an example MVP dispatch plan with four vehicles. Each curved arc corresponds to a vehicle dispatch, while the preceding horizontal straight line represents the time in which that dispatch's orders accumulate while the vehicle waits. Line styles are alternated for visual clarity.

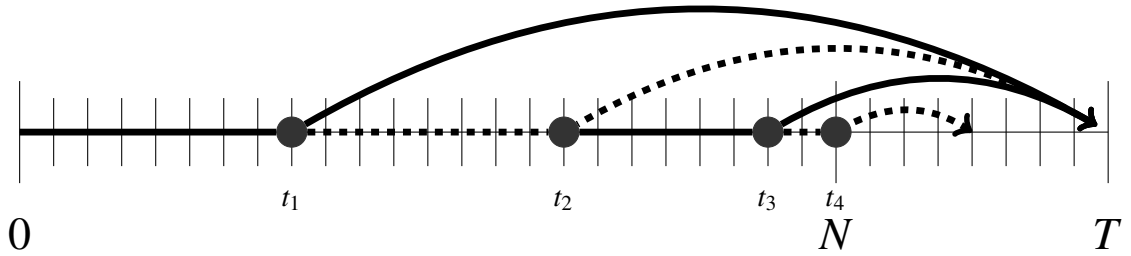


Figure 2.1: Visual representation of an MVP requiring four vehicles.

Example 1. A retailer provides SDD service for an 8 mile by 8 mile service region, with an average of 75 orders placed over a 10-hour cutoff time. The retailer operates over a 12-hour service day and has an unrestricted fleet size. We scale time to 8 minutes per time unit, and the model parameters are set to $N = 75$ and $T = 90$. Additionally, suppose that the dispatch time function is $f(n) = 2.15\sqrt{n} + .13n$, roughly equivalent to a routing time approximation (Manhattan distances [27], with vehicles traveling at 25 miles per hour), plus a service time of 1 minute per order. The MVP returns the optimal solution,

$$\begin{array}{lll}
 t_1 = 64.38, & q_1 = 64.38, & i_1 = 1, \\
 t_2 = 75, & q_2 = 10.62, & i_2 = 2,
 \end{array}$$

which uses two vehicles and 272.06 total minutes of dispatch time.

2.3.2 One Vehicle

Now assume the fleet consists of a single delivery vehicle. If $N + f(N) \leq T$, this vehicle can (optimally) wait until time $t = N$ to dispatch once with all N orders. More generally, we use the following lemma in our analysis. Qualitatively, this lemma suggests that given a fixed amount of orders to be served, it would be better to split the orders between two dispatches as unevenly as possible to maximize routing efficiencies.

Lemma 2. For any $0 \leq a \leq b \leq Y$, the optimal solution of

$$\min_{a \leq y \leq b} \{f(y) + f(Y - y)\}$$

is $y^* = a$ if $a \leq Y - b$, and $y^* = b$ otherwise.

Proof. This result follows directly from the concavity of f . □

Before stating our main results for a single vehicle, we must address a pathology arising from Lemma 2. Consider an instance with $N = 9, T = 11.99, f(n) = \sqrt{n}$; clearly, a single dispatch is infeasible, since the vehicle would return to the depot too late by 0.01 time units. It can be shown that any feasible two-dispatch policy satisfies $0.06 \leq q_1 \leq 8.25$ and $q_2 = 9 - q_1$. By the lemma, the best solution using two dispatches is given by $q_1^* = 0.06$. Furthermore, this policy is in fact optimal for a single vehicle over any number of dispatches. There are two characteristics of this optimal policy worth noting. First, the first time of dispatch can be adjusted to any time in the range of $q_1^* \leq t_1^* \leq N - f(q_1^*)$, while $t_2^* = N$. Second, the first dispatch size is very small, which is likely to be unreasonable since the square root approximation of routing time tends to be inaccurate for small numbers.

This example shows that the model requires additional assumptions to produce meaningful answers. We now introduce three additional conditions that address these concerns.

Sufficient processing speed: There exists $q_{\min} < \infty$ such that $f(x) \leq x/\lambda = x$, for all $x \geq q_{\min}$.

Sufficient gap time: The parameters T, N, q_{\min} satisfy $T - N \geq f(2q_{\min})$.

Minimum dispatch size: Any feasible solution $\{(t_d, q_d, i_d)\}_{d=1}^D$ satisfies $q_d \geq q_{\min}$ for all $d < D$.

The first condition ensures that the system can “keep up” with orders; that is, whenever the delivery vehicle is dispatched with a large enough quantity, it is guaranteed to arrive

back at the depot to find a lesser number of unserved orders. The second condition ensures that there is enough time between the last order arrival and the end of service day. Finally, from a modeling perspective, a constraint on the minimum size of a dispatch is justifiable economically. The last dispatch is not subject to this constraint, since it must serve all remaining orders, regardless of their number. The sufficient processing speed and sufficient gap time conditions imply feasibility, including satisfying the minimum dispatch size condition; the next lemma formalizes this argument.

Lemma 3. *An SDD problem instance with a single vehicle and parameters satisfying the sufficient processing speed and sufficient gap time conditions has a feasible solution satisfying the minimum dispatch size constraints.*

Proof. See Appendix A.2. □

We next state our main results on optimal dispatch policies for a single vehicle.

Theorem 4. *A single-vehicle SDD instance satisfying the sufficient processing speed and sufficient gap time conditions, and with the additional constraints imposed by the minimum dispatch size condition, has an optimal dispatch policy such that*

- (C1) *each dispatch takes all available unserved orders at the depot at the time of dispatch,*
- (C2) *after the first dispatch, the vehicle never waits at the depot again, and*
- (C3) *if the vehicle is dispatched more than once, the last dispatch arrives back at the depot exactly at time $t = T$.*

Proof. See Appendix A.3. □

Theorem 4 implies that there exists an optimal policy $\{(t_d^*, q_d^*)\}_{d=1}^{D^*}$ that can be described completely by t_1^* . By (C1), we have $q_1^* = \lambda t_1^* = t_1^*$, and then (C2) yields $t_2^* = t_1^* + f(q_1^*)$. Applying this reasoning recursively, $q_2^* = \lambda(t_2^* - t_1^*) = t_2^* - t_1^*$ and so on, continuing until the last dispatch covers all the remaining orders, leaving at or after time $t = N$.

In contrast, the example after Lemma 2 shows that when an instance does not satisfy the three conditions, the optimal solution need not have the structure outlined in the theorem. Specifically, the example's optimal solution does not satisfy at least one of (C1) or (C2).

Algorithmically, we apply Theorem 4 to restrict our search for an optimal policy to feasible policies that satisfy (C1) and (C2). Among such policies, we search for one that satisfies (C3) and has a first dispatch time $t_1 = \alpha$ as large as possible; (C2) and (C3) guarantee that maximizing α is equivalent to minimizing the vehicle's total dispatch time.

Formally, for positive integers δ, d , let

$$f^1 := f, \quad f^\delta(\cdot) := f(f^{\delta-1}(\cdot)), \quad \delta \geq 2, \quad h_d(\alpha) := \alpha + \sum_{\delta=1}^{d-1} f^\delta(\alpha).$$

Intuitively, f^δ is the δ -th composition of f , and we use f^δ to define $h_d(\alpha)$, the number of orders served by the first d dispatches for a policy satisfying (C1) and (C2) with first dispatch at time α . If a feasible policy satisfying (C1) and (C2) has a first dispatch at time α and a total of $D \geq 2$ dispatches, it serves $h_{D-1}(\alpha)$ orders with its first $D-1$ dispatches, while the last dispatch serves the remaining $N - h_{D-1}(\alpha)$ orders, where $h_{D-1}(\alpha) \leq N \leq h_{D-1}(\alpha) + f^{D-1}(\alpha) = h_D(\alpha)$. This implies natural bounds on the first dispatch time when using exactly D dispatches: If we define $\alpha_1 = N$ and α_2 such that $\alpha_2 + f(\alpha_2) = N$, then all feasible dispatch policies satisfying (C1), (C2) and using exactly two dispatches have a time of first dispatch in $[\alpha_2, \alpha_1)$. Generalizing, if we define α_D as the unique α satisfying $h_D(\alpha) = N$, all feasible policies that satisfy (C1), (C2) and use exactly D dispatches have a time of first dispatch in $[\alpha_D, \alpha_{D-1})$.

Because the α_D values are decreasing in D and we want to maximize α , we can iteratively search for this time of first dispatch by fixing the policy's total number of dispatches D , starting with $D = 1$. For each D , we attempt to find the largest $\alpha \in [\alpha_D, \alpha_{D-1})$ satisfying $h_D(\alpha) + f(N - h_{D-1}(\alpha)) = T$. If such an α exists, it must correspond to the first dispatch time of an optimal policy satisfying (C1), (C2) and (C3); if no such α exists, we increase

D by one and repeat. Algorithm 1 states this procedure in pseudo-code, and Corollary 5 describes its utility.

Corollary 5. *Suppose a single-vehicle SDD instance satisfies the sufficient processing speed and sufficient gap time conditions, and we impose the additional constraints of the minimum dispatch size condition. Algorithm 1 determines the time of first dispatch for an optimal policy satisfying (C1), (C2), and (C3) from Theorem 4.*

Proof. See Appendix A.4. □

Algorithm 1 Calculating the optimal time of first dispatch in the single-vehicle case.

```

1:  $D \leftarrow 1, \alpha^* \leftarrow 0$ 
2: if  $N + f(N) \leq T$  then
3:    $\alpha^* \leftarrow N$ 
4: else
5:   while  $\alpha^* = 0$  do
6:      $D \leftarrow D + 1$ 
7:      $A \leftarrow \{\alpha \in [\alpha_D, \alpha_{D-1}) : h_D(\alpha) + f(N - h_{D-1}(\alpha)) = T\}$ 
8:     if  $A \neq \emptyset$  then
9:        $\alpha^* \leftarrow \max_{\alpha \in A} \alpha$ 
10:    end if
11:  end while
12: end if
13: return  $\alpha^*$ 

```

Example 2. Consider the same instance as in Example 1, that is, $N = 75, T = 90, f(n) = 2.15\sqrt{n} + .13n$. Now suppose the fleet has a single delivery vehicle, and $q_{\min} = 12$. This set of model parameters satisfy the sufficient gap time, and sufficient processing speed

conditions, so we can use Theorem 4 and our root finding algorithm to compute an optimal dispatch policy also satisfying the minimal dispatch size condition.

We first set $D \leftarrow 1$, and $\alpha_1 \leftarrow 75$. As $N + f(N) = 103.37 > 90 = T$, a single dispatch is insufficient and we must consider using two or more dispatches. For two dispatches, we calculate that $\alpha_2 = 52.58$, and thus consider all policies with two dispatches where $\alpha \in [52.58, 75)$. From here we wish to determine if there is an α in this range satisfying $\alpha + f(\alpha) + f(N - \alpha) = T$. Indeed, $\alpha = 54.65$ solves this expression, and the algorithm terminates with $\alpha^* = 54.65$ and $D^* = 2$. The calculated optimal policy is

$$\begin{array}{lll} t_1 = 54.65, & q_1 = 54.65, & i_1 = 1, \\ t_2 = 77.65, & q_2 = 20.35, & i_2 = 1, \end{array}$$

with a total dispatch time of 282.74 minutes; see Figure 2.2. By comparing this example to the previous one, we can see that by decreasing the fleet from two to one vehicles we would increase the total dispatch time by less than 4% (272.06 minutes to 282.74 minutes).

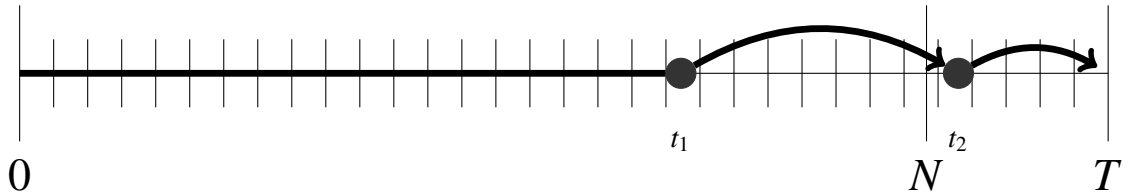


Figure 2.2: Visual representation of optimal dispatch policy for Example 2.

2.3.3 General Fleet Size

We now consider the more complex case in which the fleet is finite but greater than one, and thus many vehicles may need to be dispatched more than once. Unfortunately, it is no longer possible to show that optimal policies satisfy simple structural properties in this case. For example, consider a family of instances where the fleet has two vehicles, but three dispatches are required to serve the orders feasibly. Depending on the parameters T , N and

f , it is possible to construct both cases where it is optimal for the vehicle dispatched first to return before T and to make a second dispatch while the second vehicle is used only once, but it is also possible to find cases where it is instead optimal for the vehicle dispatched second to make the additional dispatch.

Although the optimization of this case is significantly more complex, we can leverage our analysis of the previous cases to construct a heuristic policy, described next.

Hybrid Policy For a fleet with m vehicles, the first $m - 1$ are dispatched according to the MVP. The final vehicle serves all remaining orders according to the single-vehicle policy computed with Algorithm 1.

The next result shows that this heuristic policy produces solutions within a worst-case factor of optimality for an important class of dispatch time functions.

Theorem 6. *Assuming the sufficient processing speed and sufficient gap time conditions, the hybrid policy is feasible, including satisfying the minimum dispatch size condition constraints. Furthermore, suppose $f(n) = bn + c\sqrt{n}$, where $b \geq 0$ and $c \geq 0$. If the hybrid policy dispatches the last vehicle D_m times, its total dispatch time is guaranteed to be within a factor $\frac{m-1+D_m\sqrt{D_m}}{m-1+D_m}$ of the Many-Vehicle Policy's dispatch time, which is itself a lower bound for any m -vehicle solution.*

Proof. See Appendix A.5. □

See Figure 2.3 below for an example of the hybrid policy compared to the MVP.

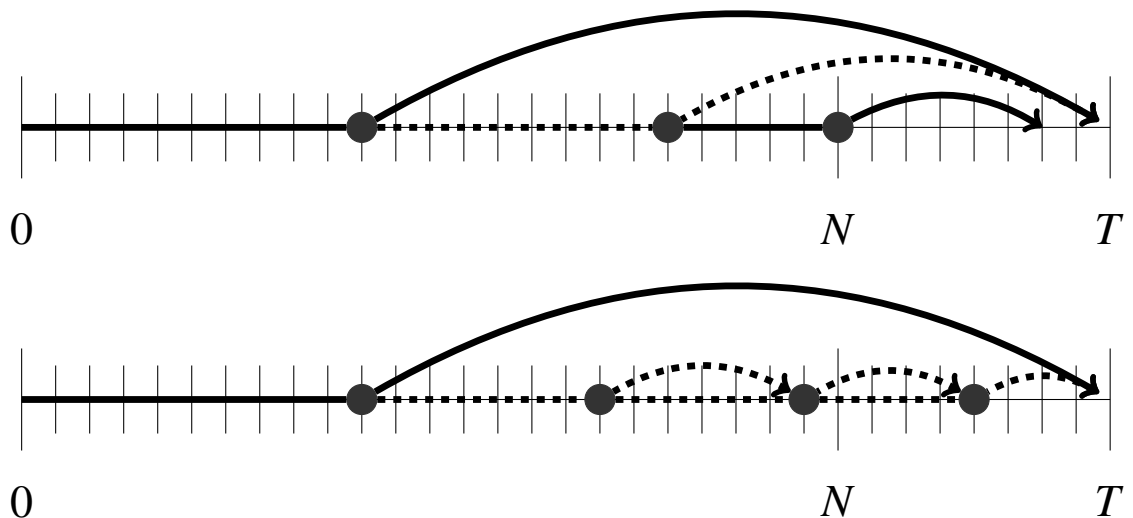


Figure 2.3: In this example, the MVP uses three vehicles. If only two are available, the hybrid policy stipulates that the first vehicle behaves as in the MVP, while the second performs an optimal single-vehicle dispatch policy on the remaining orders, which in this case requires three dispatches.

Example 3. Consider again the problem instance from Examples 1 and 2. The single-vehicle policy is a special case of the hybrid policy when $m = 1$ (and thus a single vehicle must make all dispatches even if the MVP uses multiple vehicles). Because the single-vehicle policy uses two dispatches in this instance, we conclude from Theorem 6 that its cost is at most a factor $\sqrt{2} \approx 1.41$ larger than the MVP cost. In this case, we know from direct calculation that this cost difference is much smaller, only around 4%.

The guarantee provided by Theorem 6 improves as m grows, since more of the hybrid policy's dispatches exactly mimic what the MVP does; for example, with $m = 2$ and $D_2 = 2$, the guarantee improves to $(1 + 2\sqrt{2})/3 \approx 1.28$.

2.4 Model Applications

The discussion in Examples 1 and 2 demonstrate how our model can be applied for tactical design, specifically in fleet sizing. We next discuss other potential uses of the model.

2.4.1 Serving the Entire Region versus Partitioning

Location analysis and customer assignment are important strategic and tactical questions in logistics, and continuous approximation models have been successfully applied for service region design, e.g. [40]. We can similarly ask in an SDD context whether partitioning the service region offers advantages over simply having every vehicle serve the entire region.

Consider a dispatch time function $f(n) = a + bn + c\sqrt{n}$ consisting of a setup time at the depot, a service time per order, and a BHH [24] routing time approximation, which depends on the size of the service region. Suppose we partition this region into m sub-regions of equal size, so that the demand arrival rate in each is $1/m$; each sub-region would then have a dispatch time function of the form $\hat{f}(n) = a + bn + c\sqrt{n/m}$, since the area the vehicle serves is scaled down by a factor of $1/m$. At time $t = N$, if a single vehicle can serve each sub-region with a single dispatch, the total dispatch time for all vehicles would be

$$m \times \hat{f}(N/m) = m \left(a + b(N/m) + c\sqrt{N/m^2} \right) = am + bN + c\sqrt{N};$$

the last two terms correspond exactly to the service and routing time a single vehicle would need to serve all N orders in a single dispatch. Therefore, if the MVP policy uses m^* vehicles and it is feasible to partition the region into m^* sub-regions and serve each with a single dispatch, partitioning is preferable. However, in general, the number of required vehicles for a partitioning strategy with a single dispatch per vehicle may differ from m^* and be either larger or smaller.

Example 4. A retailer provides SDD service for an 8 mile by 8 mile service region, with an average of 75 orders placed over a 10-hour cutoff time. The retailer operates over an 11 hour and 20 minute service day. We scale time to 8 minutes per time unit, and the model parameters are set to $N = 75$ and $T = 85$. Additionally, take the dispatch time function as $f(n) = 1.88 + .25n + 2.15\sqrt{n}$, roughly equivalent to a routing time approximation (Manhattan distances [27], with vehicles traveling at 25 miles per hour), plus a service time of 2

minutes per order and a setup time of 15 minutes. The MVP returns the optimal solution

$$\begin{array}{lll}
 t_1 = 53.87, & q_1 = 53.87, & i_1 = 1, \\
 t_2 = 70.30, & q_2 = 16.43, & i_2 = 2, \\
 t_3 = 75, & q_3 = 4.70, & i_3 = 3,
 \end{array}$$

with 428.37 total minutes of dispatch time. In contrast, the minimum number of vehicles needed for a partition strategy as described above is five, with each delivering 15 orders in a total of 374.16 minutes. The manager must then decide whether saving 54.21 minutes per service day is worth an additional two vehicles in the SDD fleet. We can similarly use our single-vehicle policy to develop partitioning strategies with a single vehicle serving each sub-region but performing multiple dispatches.

2.4.2 Orders at the Start of the Service Day

Thus far, our model assumes no orders are ready for dispatch at the start of the service day. It may be that the SDD system is also required to serve some next-day or overnight orders. In the model, this translates to a number $N' \geq 0$ of orders that are ready at the start of the service day.

In the many-vehicle case, the optimal policy is similar to the MVP, with one modification. Let $Q = f^{-1}(T)$, i.e. Q is the unique number satisfying $f(Q) = T$ (the inverse exists and Q is unique because f is increasing); this number is implicitly a capacity on the number of orders a vehicle can carry during the service day to remain time-feasible. We can now define a generalized MVP, which returns an optimal policy when $N' \geq 0$ orders are ready at the depot at the start of the service day. The proof of this claim is found in the appendix, A.6.

Generalized MVP At time $t = 0$, dispatch as many vehicles as possible each carrying exactly Q orders. The subsequent dispatches are calculated via the MVP.

If the number of orders available at the start of the day is large, the generalized MVP may not capture additional opportunities for routing efficiency stemming from directly optimizing a vehicle routing problem for these orders; however, such opportunities do not relate to the SDD system and would rely on more established routing models.

In the single vehicle case, it is possible that a problem instance previously defined by N, T , and f is still feasible for $N' > 0$. Define an augmented problem with $\bar{N} = N' + N$, $\bar{T} = N' + T$, and $\bar{N}' = 0$. If the solution to this problem has an initial dispatch quantity of $\bar{q}_1^* \geq N'$, then the optimal quantities for the original instance are identical to those of the augmented one, with dispatch times moved up by N' .

More generally, by relaxing the gap time condition from section 2.3.2, we can solve the single-vehicle problem for any instance of $N' > 0$. Practically, this involves increasing the gap between the order cutoff time and the end of the service day.

Generalized gap time condition: Parameters T, N, N', q_{\min} satisfy $T - N \geq N' + f(2q_{\min})$.

Assume the generalized gap time condition holds for parameters T, N, q_{\min}, N' . As before, define and solve an augmented problem with $\bar{N} = N' + N$, $\bar{T} = N' + T$, and $\bar{N}' = 0$. If one dispatch is optimal, it follows that $\bar{q}_1^* = \bar{N} \geq N'$. In the case of a multiple-dispatch optimal solution, the vehicle will only ever be idle at the depot at the start of the service day and will be done serving orders exactly at time \bar{T} . Thus, the total dispatching time is equal to $\bar{T} - \bar{q}_1^* = T + N' - \bar{q}_1^*$. Because the original problem over N, N' , and T is feasible, we know that the total dispatch time for any optimal policy is less than or equal to T units of time. Additionally, any feasible solution to the original problem can be implemented in the augmented problem. Therefore the optimal solution to the augmented problem must use less than or equal to T units of dispatch time. It follows that $\bar{q}_1^* \geq N'$. Therefore, in all cases it is true that $\bar{q}_1^* \geq N'$, which implies that the optimal quantities for the original instance are identical to those of the augmented one, with dispatch times moved up by N' .

2.4.3 Capacitated Vehicles

Compared to traditional delivery settings, SDD systems operate in an environment with reduced order volume and much tighter time constraints. Therefore, in many cases the number of orders that can be delivered while satisfying time constraints is relatively small, and thus vehicle capacity is not a binding constraint. Nevertheless, there may be situations in which capacities must also be considered; we next discuss how our results extend to this case. Extending the notation we use in Section 2.4.2, suppose that each vehicle has capacity to serve at most Q orders. Consider first the many-vehicle case, and the following natural extension to the MVP.

Capacitated MVP Compute the MVP; if the first (and largest) dispatch serves Q or fewer orders, implement the policy. Otherwise, dispatch the first vehicle with Q orders, update T and N by subtracting Q , and recompute the MVP on the updated instance. Repeat until the computed MVP is feasible.

Intuitively, the MVP tries to serve as many orders as possible with each successive dispatch while still having the corresponding vehicle return by the end of the service day. The capacitated version of the policy does the same, but must also respect the additional vehicle capacity. As with the Generalized MVP in the previous section, this policy is also an optimal dispatching policy. The proof of this is identical to that of the Generalized MVP found in the appendix.

We can implement a similar policy modification in the single-vehicle case, where we additionally assume $Q \geq q_{\min}$. As in the many-vehicle case, Algorithm 1 constructs a solution in which the first dispatch is the largest; when this dispatch is too large for the capacity, the following modification replaces it with the maximum possible quantity and iterates on the remaining, smaller instance.

Capacitated Single-Vehicle Policy Run Algorithm 1; if $\alpha^* \leq Q$, implement this solution. Otherwise, update T and N by subtracting Q , and run Algorithm 1 on the smaller instance. Repeat until $\alpha^* \leq Q$ in Algorithm 1. If this process occurs k times before α^* is feasible, the solution to the instance has k dispatches of size Q before the dispatches given by Algorithm 1.

As a simple example, suppose $N + f(N) \leq T$ but $Q < N \leq 2Q$, i.e. a single dispatch is time-feasible but capacity-infeasible, and two dispatches suffice (although this is not what we expect in SDD, the setting serves for illustration purposes). In this case, it is optimal for the first dispatch to take Q orders, and the second dispatch takes the rest; the two dispatches can take place consecutively, returning at time T . More generally, we are able to give the following guarantee.

Corollary 7. *Suppose $Q \geq 2q_{\min}$. The capacitated single-vehicle policy produces an optimal solution to the single-vehicle instance with vehicle capacity Q .*

The proof of Theorem 4 can be modified slightly to prove Corollary 7. The only additional consideration needed is that each vehicle takes all unserved orders at the depot at the time of dispatch, *up to quantity Q* . The requirement $Q \geq 2q_{\min}$ is a technical condition necessary to use that same proof; however, in practice it is reasonable to expect that the capacity of a delivery vehicle is at least twice its minimum quantity.

2.4.4 Choosing Order Cutoff Time

Consider again the instance in Example 1. In an optimal solution, the second vehicle has almost 53 minutes of slack between its earliest possible arrival back to the depot and the end of the service day T . A system designer could consider either reducing N so the system requires only one vehicle, or increasing N to serve more orders with this second vehicle and increase its utilization.

For this discussion, we fix T, f, q_{\min} and allow N to vary in $[0, U]$, where U is an upper

bound chosen by the system manager. We assume that earned revenue from orders served is proportional to N with constant β , and operating costs are proportional to the dispatch time of the optimal policy, denoted by $g(N)$. Without loss of generality, we scale β so we can compare revenue against cost. Suppose the SDD manager wishes to choose a cutoff time that maximizes system profit as measured by earned revenue minus operating costs. Equivalently, this cutoff time would also minimize the cost of serving orders that occur before the cutoff plus the opportunity cost of not serving orders after the cutoff. The profit maximizing cutoff time is then given by

$$\max_{0 \leq N \leq U} \pi(N) = \beta N - g(N). \quad (2.2)$$

First, we analyze the many-vehicle case. Recall that in the MVP, if the solution uses m vehicles, the first $m - 1$ return exactly at time T . If the cutoff time is chosen carefully, the last dispatch also returns precisely at this time. Specifically, let N_i be the cutoff time at which the MVP uses exactly i vehicles, with all vehicles returning exactly at time T . Letting, $N_0 = 0$, we have the recursion $N_i = N_{i-1} + \Delta_i$, where Δ_i uniquely solves $\Delta_i + f(\Delta_i) = T - N_{i-1}$. Define N_U as the largest such value such that $N_U \leq U$. The following theorem leverages these values to search for an optimal cutoff time for (2.2) in the many-vehicle case.

Theorem 8. *In the many-vehicle case, an optimal solution for (2.2) satisfies*

$$N^* \in \{N_0, N_1, \dots, N_U, U\}.$$

Proof. See Appendix A.7. □

Now we analyze the single-vehicle case, where we impose the upper bound $U \leq T - f(2q_{\min})$ on N . Define i_{\max} as the number of dispatches in the optimal dispatch policy when $N = U$, and suppose $i_{\max} \leq 2$. Define \bar{N}_1 such that for $N \in (0, \bar{N}_1]$, the optimal dispatch policy determined via Theorem 4 uses exactly one dispatch.

Proposition 9. *In the single vehicle case, for any U such that $0 \leq U \leq T - f(2q_{\min})$ and $i_{\max} \leq 2$, the optimal solution of (2.2) satisfies $N^* \in \{0, \bar{N}_1, U\}$.*

Proof. See Appendix A.8. □

The proof for the general case of i_{\max} would follow from showing that $g(N)$ is piecewise concave with breakpoints at each \bar{N}_i . We have empirical evidence that this is indeed the case but have not been able to prove it generally.

We now return to the instance in Examples 1 and 2, and calculate an optimal value for the cutoff time N .

Example 5. Consider the same instance as in the previous examples with $T = 90$, $f(n) = 2.15\sqrt{n} + .13n$, and $q_{\min} = 12$. Suppose $\beta = 0.80$ and profit is given in scaled monetary units.

For the many-vehicle case, we calculated the first four values of N_i to be $N_0 = 0, N_1 = 64.38, N_2 = 79.62$, and $N_3 = 84.57$. The associated dispatch times are 0, 25.62, 36.00, and 41.43, which result in profits of $\pi(N_0) = 0$, $\pi(N_1) = 25.88$, $\pi(N_2) = 27.70$, and $\pi(N_3) = 26.22$. By Theorem 8, the optimal order cutoff time is $N^* = N_2$, with an optimal dispatch policy of $\{(t_1 = 64.38, q_1 = 64.38, i_1 = 1), (t_2 = 79.62, q_2 = 15.24, i_2 = 2)\}$.

For the single-vehicle case, we let $U = T - f(2q_{\min}) = 76.35$, and it follows that $i_{\max} = 2$. Note that, $\bar{N}_1 = 64.38$. The associated costs of $N^* \in \{0, \bar{N}_1, U\}$ are 0, 25.62, and 36.04, which result in profits of $\pi(0) = 0$, $\pi(\bar{N}_1) = 25.88$, and $\pi(U) = 25.38$. Thus, the optimal cutoff time is $N^* = \bar{N}_1$ with the corresponding policy $\{(t_1 = 64.38, q_1 = 64.38, i_1 = 1)\}$.

Figures 2.4 and 2.5 plot $\pi(N)$ for this instance in the many and single vehicle cases, respectively.

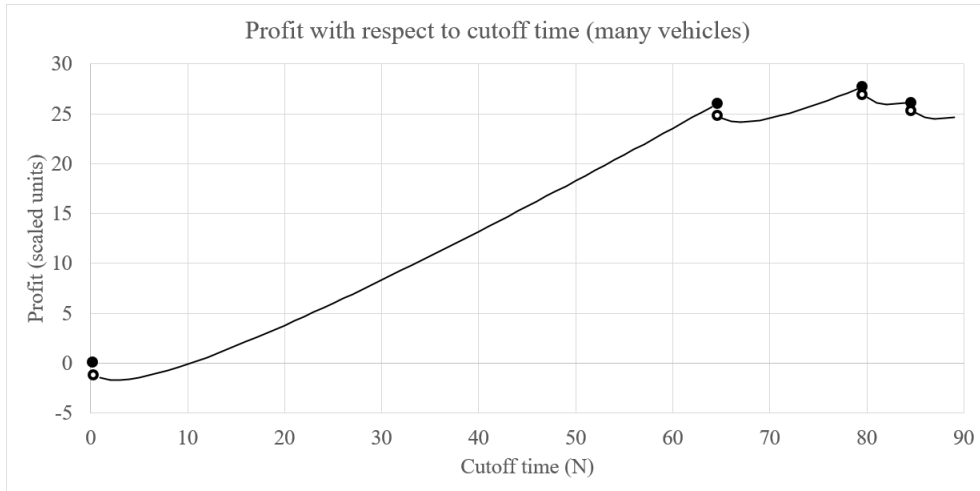


Figure 2.4: Profit function with respect to cutoff time, many-vehicle case

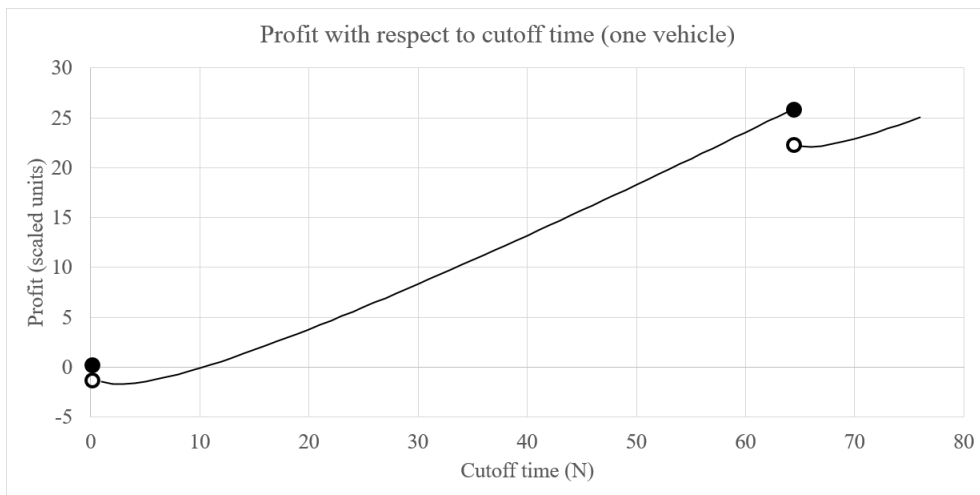


Figure 2.5: Profit function with respect to cutoff time, single-vehicle case

2.5 Computational Study

Our SDD planning model uses continuous approximations to preserve simplicity and transparency of analysis, and the previous sections discuss various ways in which the model can be used to inform managerial decisions about an SDD system, including fleet sizing, the choice of service cutoff time, and so forth. When making such decisions, we naturally depend on the accuracy and fidelity of the model when compared to more granular opera-

tional models. We now present a computational case study that empirically demonstrates the accuracy of the model and its potential practical use.

Our study considers a hypothetical SDD system where one service area comprises roughly 26 square miles in northeastern metro Atlanta. Specifically, this service region consists of the 22 census tracts north of Interstate 85, south of Interstate 285 and east of Georgia Highway 400, with a population of 92,198 as measured by the U.S. Census Bureau [42]. For the study, we chose five representative addresses within each tract, for a total of 110 potential customer locations, plus a depot location on the northeast border, on Interstate 285. Figure 2.6 depicts the service region and the 110 representative customer locations.

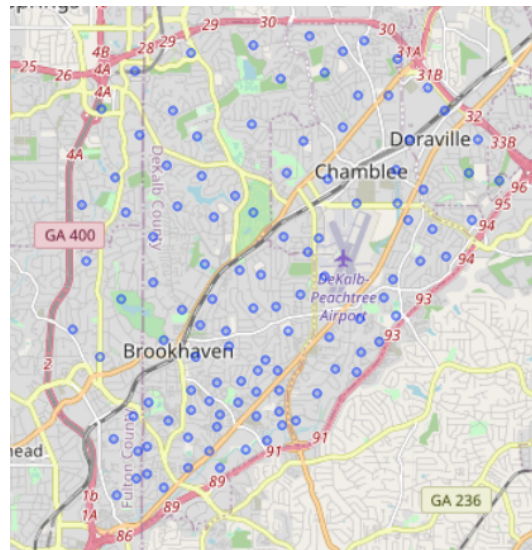


Figure 2.6: Service region in northeast Atlanta, Georgia, USA. Image generated in RStudio with the library “leaflet”, containing all 110 possible customer locations taken from 22 census tracts [42].

We assume SDD orders begin at 9 am, and the service day ends at 6 pm; we evaluate two different order cutoff times below in different experiments. Assuming 5% of the working population in the region would like to use the SDD service once every other month within the service day, an order would arrive on average approximately once every six minutes over this time, and we use this rate in the model. However, we do not assume orders are

equally likely to appear anywhere in the region: We assign each tract a weight proportional to the product of its median household income and its population (both available from [42]); assuming an order originates in a tract, one of its five representative locations is chosen uniformly at random.

To construct the routing component of our dispatch time function, we sampled customer locations with replacement according to the weighted distribution described above, with sample sizes ranging from 10 to 75 locations and a total of 1,980 samples. We used the Google Maps API [43] to query driving time between every pair of locations, and for each sample calculated the optimal TSP route time for a vehicle route visiting these locations from the depot. From this, we obtained via linear regression a routing time approximation of $24\sqrt{n}$ minutes for n orders, with an R-squared value of 0.94. Furthermore, we include 1.5 minutes of service time per order and a fixed setup time of 10 minutes per dispatch. After re-scaling the instance parameters to be measured in increments of six minutes, this results in an order arrival rate $\lambda = 1$, service day length $T = 90$ and dispatch time function

$$f(n) = 1.67 + 0.25n + 4\sqrt{n}.$$

For our operational simulation, we replace the constant arrival rate with a Poisson arrival process that uses the same rate; as in our calibration experiment, an order's tract is chosen randomly with weights proportional to the product of median household income and population, and its location within the tract is chosen uniformly at random. We use actual driving times between pairs of locations given by the Google API and determine the total routing time for a dispatch by solving a TSP using a standard integer programming formulation implemented in Gurobi 7.5.2. Our experiments are coded in Python 3.6.3 and run on a Linux computing cluster, which employs HTCondor 8.8.4.

2.5.1 Many-Vehicle Policy

First, we assume that this service region is served by two delivery vehicles each performing a single daily dispatch. Following the reasoning from Section 2.4.4, we choose a cutoff time to fully utilize both vehicles. With these parameters, the MVP prescribes that the first dispatch takes 48.40 orders and its route requires a duration of just under 250 minutes, while the second dispatch takes 18.26 orders with a duration of approximately 140 minutes; the total dispatch time sums to about 389 minutes. The predicted dispatch time for the second vehicle translates to an order cutoff time of 3:40 PM ($N = 66.66$).

We now describe our simulated operational benchmark for the many-vehicle policy. The dispatch time required to serve a set of orders is the sum of the setup, service and routing time, where the routing time is given by a TSP from the depot to each of the orders' locations using actual driving times. The dispatcher allows unserved orders to accumulate as long as their dispatch time is less than the remaining time in the service day; when the two times are equal, a vehicle is dispatched with the unserved orders. When a new order arrives, the dispatcher recalculates the total dispatch time including the new order; if the new dispatch time exceeds the remaining time in the service day, the vehicle is immediately dispatched *without this order*, to ensure the vehicle returns before the end of the service day. If this is the first dispatch, the new order is assigned to the second vehicle and the process repeats; otherwise, this order is not accepted for SDD. More generally, the dispatcher stops accepting orders once the second vehicle is dispatched, which may occur before or after the nominal cutoff time of 3:40 PM; this represents a dynamic modification of the cutoff time at the operational level and is in line with how some SDD services operate in practice.

We simulated the operational benchmark 300 times. For each realization, we record the the number of orders served and the dispatch time for each vehicle, respectively. Table 2.2 reports results; for each quantity we include the prediction of our tactical model, and the sample mean and 95% confidence intervals of the operational benchmark. As the table shows, our tactical model predicts the expected number of orders served and the expected

total dispatch time to within less than 1%.

Table 2.2: Computational study results, many-vehicle policy.

	Tactical	Operational	A Posteriori
First Dispatch Quantity	48.40 units	48.20 units (± 0.51)	43.90 units (± 0.63)
First Dispatch Time	249.58 min.	249.69 min. (± 1.81)	228.07 min. (± 2.62)
Second Dispatch Quantity	18.26 units	18.45 units (± 0.35)	22.75 units (± 0.45)
Second Dispatch Time	139.95 min.	139.16 min. (± 1.47)	144.88 min. (± 1.48)
Total Quantity	66.66 units	66.65 units (± 0.71)	66.65 units (± 0.71)
Total Time	389.53 min.	388.85 min. (± 2.85)	372.95 min. (± 3.29)

Because our main goal is tactical design and describing average behavior rather than operational management, we do not consider many potential dispatching improvements or modifications that could be used at the operational stage; the logistics literature has several works dedicated to this question, e.g. [11, 15, 16]. Nevertheless, it is also important to assess the quality of our prescribed solution when compared to what an operational decision support tool could accomplish.

Motivated by this question, we compute an *a posteriori* or “hindsight-optimal” solution for each simulated realization, which provides a lower bound on the dispatch time any operational policy can achieve. For each realization, we assume that the dispatcher knows in advance the exact time and location of each order served by our operational benchmark and then optimizes the two vehicles’ routes with this knowledge. For example, in one of the realizations the operational benchmark could have both vehicles visiting the same neighborhood to deliver two different orders; the *a posteriori* solution could use its advance knowledge to shift the first order to be served by the second vehicle (with virtually no increase in its dispatch time), while deleting this order from the first vehicle’s dispatch would reduce its dispatch time. This example also illustrates that despite having advance

knowledge of order times and locations, the *a posteriori* solution must still satisfy operational constraints; in particular, an order can only be served by a dispatch that departs the depot after the order ready time, and the vehicles must return to the depot by the end of the service day. In the appendix, we include the formulation we use to compute these solutions.

In our experiments, for each of the 300 simulated realizations we optimize the *a posteriori* solution in Gurobi with a two-hour time limit. As Table 2.2 details, the sample mean of the total dispatch time in the *a posteriori* solution is within approximately 4% of the operational benchmark. For comparison, operational SDD models are notoriously difficult to benchmark; many works in the SDD literature do not include lower bounds at all, and those that do often report larger gaps against *a posteriori solutions* even for complex heuristic policies, e.g. [11]. We therefore conclude that our model prescribes reasonable operational behavior, in line with what a dispatcher could accomplish with sophisticated decision-support tools.

2.5.2 Single-Vehicle Policy

We now suppose the service region is served with a single delivery vehicle. Since only a single vehicle is available, we also move the cutoff time back to 2:00 PM ($N = 50$), which results in a more reasonable workload. Using Algorithm 1 and the results in Section 2.3.2, we obtain a two-dispatch solution: The first dispatch serves 35.01 orders with a time of about 205 minutes, and the second takes the remaining 14.99 orders with a duration of approximately 125 minutes.

The operational benchmark for the single-vehicle policy here is similar in spirit to the many-vehicle one. As our discussion in Section 2.3.2 suggests, each dispatch should take all currently unserved orders. To determine the time of first dispatch α , we operationally mimic the equation $\alpha + f(\alpha) + f(N - \alpha) = T$, which determines the dispatch time in the tactical model with two dispatches. Before the first dispatch, orders accumulate and the dispatcher iteratively recalculates their total dispatch time. Define τ to be the elapsed time

in the service day, O_τ to be the set of orders that have arrived by τ , and $\text{dispatch}(O_\tau)$ to be their dispatch time. While $\tau + \text{dispatch}(O_\tau) + f(N - \tau) < T$, the dispatcher allows orders to accumulate; the first dispatch occurs when equality holds. As in the previous case, if a new order causes the left-hand value to exceed T , the vehicle is immediately dispatched without this order. While the vehicle is en route, we know its return time and thus the maximum possible duration of the next dispatch such that it returns by T . As new orders arrive, the dispatcher again calculates their dispatch time, and accept orders only until the next dispatch's total duration matches the remaining time in the service day. As in the many-vehicle case, this corresponds to an operational dynamic adjustment of the order cutoff time that, in expectation, should match N if our model is accurate. Note also that we can extend the benchmark in an analogous fashion to single-vehicle problems with more than two dispatches.

We again simulated 300 realizations of the operational benchmark, and report results in Table 2.3, in a similar fashion to Table 2.2. As in the previous experiment, our tactical model predicts the expected number of orders served and the expected total dispatch time of the operational benchmark to within 1% or less. In the second dispatch, we see a slightly higher quantity served on average in the operational benchmark compared to the tactical prediction. This is due to the routing time approximation's slight conservatism for smaller values, which allows the operational policy to serve a slightly higher number of orders than expected on average. Furthermore, we again implemented an *a posteriori* benchmark, which allows the dispatcher to optimize the two delivery routes with full advance knowledge of the time and location of each order, using the same experimental setup as in the many-vehicle case. In this case, the operational policy is within approximately 13% of the *a posteriori* benchmark, suggesting again that our system is modeling reasonable behavior when compared to what a complex operational decision support tool can hope for. Interestingly, in the single-vehicle case, we observe that the *a posteriori* solution moves more orders to the second dispatch; with advance knowledge of future order locations, the dis-

patcher is able to anticipate areas where more orders will take place late in the day, and wait until the second dispatch to serve these locations more efficiently. Nevertheless, the gaps we observe between the operational benchmark and a posteriori solution in this experiment are still in line with other results in the operational SDD literature [11].

Table 2.3: Computational study results, single-vehicle policy.

	Tactical	Operational	A Posteriori
First Dispatch Quantity	35.01 units	34.96 units (± 0.30)	12.99 units (± 1.32)
First Dispatch Time	204.52 min.	203.45 min. (± 0.74)	87.28 min. (± 6.48)
Second Dispatch Quantity	14.99 units	15.57 units (± 0.63)	37.54 units (± 1.45)
Second Dispatch Time	125.41 min.	123.73 min. (± 3.27)	196.91 min. (± 6.61)
Total Quantity	50.00 units	50.53 units (± 0.73)	50.53 units (± 0.73)
Total Time	329.93 min.	327.18 min. (± 3.79)	284.19 min. (± 3.26)

2.5.3 Many Capacitated Vehicles

Having established the model’s prediction accuracy for our two base cases in the previous experiments, we next examine vehicle capacities and their impact on model accuracy. Specifically, we consider the same experimental setup from the many-vehicle policy in Section 2.5.1, with order cutoff at 3:40 PM ($N = 66.66$), but we additionally suppose that delivery vehicles have a capacity of $Q = 20$ orders. For this instance, the capacitated MVP (Section 2.4.3) has four dispatches; the first three are at capacity, each serving 20 orders with dispatch duration of about 147 minutes; the fourth dispatch serves the remaining 6.66 orders with a predicted dispatch duration of about 82 minutes.

The operational benchmark here is almost identical to the one used in Section 2.5.1, with the additional constraint that when 20 orders accumulate, the current vehicle is dispatched immediately. We examine the performance of the operational benchmark on 300 simulated service days (the same simulations used in the uncapacitated experiment), with

results in Table 2.4. As before, the tactical model predicts orders served within 1% of its operational benchmark. In this case, the total predicted dispatch time is within 3.5% of the benchmark. We conclude that the tactical model still makes accurate predictions in the presence of a larger number of vehicles with a maximum capacity. Moreover, as the table indicates, the slightly larger discrepancy between predicted and observed dispatch time is mostly due to two factors: First, the third dispatch is not always at capacity, serving about 19 orders on average; second, and most importantly, the fourth dispatch's observed duration is lower than its prediction. Both factors can be explained by the routing approximation's slight conservatism and decrease in accuracy for a small number of locations, particularly the very small number of orders served by the last dispatch. (Since continuous approximations assume a large number of locations, they can be slightly inaccurate when the number of locations is very small.) In practice, our discussion in Section 2.4.4 suggests that the manager of this SDD system would probably prefer to either decrease the order cutoff to 60 and fully utilize only three vehicles, or perhaps to increase the cutoff so the fourth vehicle can be better utilized.

Table 2.4: Computational study results, four capacitated vehicles at $Q = 20$.

	Tactical	Operational
First Dispatch Quantity	20.00 units	20.00 units (± 0.00)
First Dispatch Time	147.33 min.	145.80 min. (± 0.83)
Second Dispatch Quantity	20.00 units	20.00 units (± 0.00)
Second Dispatch Time	147.33 min.	145.89 min. (± 0.91)
Third Dispatch Quantity	20.00 units	18.87 units (± 0.27)
Third Dispatch Time	147.33 min.	141.18 min. (± 1.52)
Fourth Dispatch Quantity	6.66 units	7.28 units (± 0.64)
Fourth Dispatch Time	81.93 min.	72.54 min. (± 3.62)
Total Quantity	66.66 units	66.15 units (± 0.89)
Total Time	523.92 min.	505.42 min. (± 6.51)

2.6 Conclusions

We have proposed a tactical analysis model for same-day delivery that captures operations at the level of a single depot and its service region. By approximating the order arrival process and the dispatch time, we are able to derive simple and transparent optimal solutions for the model that describe the average performance of a reasonable SDD system; our empirical validation shows that the model can indeed predict system behavior very accurately at an operational level.

Using our model, a system manager can easily perform what-if analysis on various potential system configurations, and compare the cost and operating conditions of these configurations to decide various tactical questions, such as the size of the delivery fleet, the order cutoff time, or whether to have vehicles deliver to the entire service region versus partitioning the region by vehicle. We similarly hope the community derives other applications

of the model in SDD tactical design.

Our results motivate several interesting avenues for research. One possibility is to further investigate the interplay of service region partitioning with our model. For example, it would be useful for SDD managers to know precisely when partitioning is preferable to serving the whole region, or to determine if the system can operate more efficiently by serving different parts of the service region differently. A manager may wish to offer SDD with different cutoff times in different areas, based on how efficiently customers in the different areas can be served; perhaps more densely populated urban centers can be profitably served until later in the day while outlying suburban areas need an earlier cutoff. More generally, it may be useful to address partitioning and fleet sizing in tandem, where some sub-regions are served by more vehicles because of higher order density, while others get a smaller delivery fleet because of relative order paucity.

CHAPTER 3

TIME-VARYING SERVICE REGIONS IN SAME-DAY DELIVERY

3.1 Introduction

Driven by increased internet access, the e-commerce retail sector has been expanding steadily in recent years. Changes in travel behavior due to the COVID-19 pandemic [2, 3] have accelerated this trend: total e-retail volume in the U.S. between April 2020 and March 2021 surpassed \$817 billion, representing an increase of over 30% from the prior year [4, 5]. In an effort to capture a larger share of this market, e-retailers have improved their delivery time guarantees. Same-day delivery (SDD), which was once leveraged as a service offering differentiator, has now become expected by consumers at large. Amazon, which has been offering SDD to select premium subscribers for over a decade [6], continually bolsters their same-day supply chain network in order to serve customers faster and provide SDD options in more cities [44]. Large American retailers such as Walmart, Target, and Costco have recently turned to partnering with third party managers of their SDD systems [45]. Some smaller niche retailers, including Sephora (beauty products) and Michaels (arts and crafts), have done the same in order to provide SDD to their customers [46, 47]. Clearly, there is a need for designing efficient SDD systems in order to satisfy consumer demand while maintaining their profitability for retailers.

As a service offering, SDD allows e-commerce firms to directly compete with brick-and-mortar retail by providing the customer with near-instant gratification; however, this pressure to deliver under a same-day deadline requires careful planning. The last-mile component of traditional parcel delivery often exceeds 50% of total costs [48, 49], and this issue is exacerbated in SDD systems, which provide reduced opportunities for consolidation due to a high degree of dynamism. While SDD service offerings are simplified on the

front-end for customers, they entail difficult optimization problems for system designers and operators.

One such problem is the question of who to offer SDD fulfillment to and how late in a service day to offer the SDD promise. A service region that is too small or an early deadline may lose SDD customers and market share, while a large region or a late SDD order cutoff time may result in very costly operations or failed deliveries and loss of customer goodwill. Considerations of equity and access are also of paramount importance: Amazon has faced criticism in the past for perceived racial bias in their selection of SDD service regions [50], which they later addressed [51, 52]. If designed well, however, e-commerce systems — including SDD — have potential to help customers who may be unable to access traditional brick-and-mortar retail stores. With this motivation in mind, our goal in this study is the selection of SDD service regions and order cutoff deadlines from the perspective of an e-retailer operating from a single fulfillment center (i.e., depot) with a fixed delivery fleet.

Our objective is to determine a service region and order deadline that maximizes the expected order volume the retailer can feasibly serve each day. In particular, we study the question of whether the system gains by allowing the service region to vary over the course of the service day; i.e., by offering different deadlines to different parts of the overall region. Our results indicate that the system may indeed increase order served substantially by such variations. The intuition behind this result is straightforward. If the e-retailer is operating a small fleet, customers that are farther away from the depot (e.g., in suburban areas) may need to place orders earlier in the day to obtain SDD. We can increase our SDD order volume by allowing nearby customers, who can be served by more efficient vehicle tours, to place SDD orders until later in the day.

While there has been significant research attention devoted to SDD in recent years, the focus has been on the operational (i.e., regarding decisions which are made over the course of a service day) management of SDD systems rather than tactical level system design (i.e., regarding decisions which are made and implemented every few weeks or

months). These works seek to optimize day-to-day operations, including vehicle routing and order acceptance mechanisms, of SDD systems [11, 12, 13, 14, 16]. While these studies are paramount to understanding how to efficiently manage a pre-defined system, they do not focus on optimally designing aspects of the system topology itself. For instance, operationally-focused SDD literature often assumes known vehicle fleet sizes, or a fixed service region from which SDD demand realizes. It may be possible to solve multiple operational models over a variety of design considerations; however, such a process is highly unlikely to be tractable and transparent.

In order to capture average case system behavior for tactical decision-making, we leverage continuous approximation methods. Such approximations allow one to remove fine-grained operational complexities from the tactical design process. It was first proposed in [53, 54] to use continuous approximations of SDD systems to study the tactical problems of expected cost minimization and fleet sizing. These works, however, assume a fixed service region determined *a priori*. Our work here is the first to use continuous approximation techniques to design SDD systems in which a system manager decides not only when to dispatch delivery vehicles, but to what service region with the possibility of changing this service region over the course of the day. The methods and results presented in this chapter are intended to support SDD e-retailers in conducting effective system design and assessing potential design options.

Section 3.1 concludes with a review of the relevant literature. A formal definition of our general model is given in Section 3.2. In Section 3.3, we analyze a one vehicle, one dispatch variant of the model. In Section 3.4, we study the setting in which multiple vehicles each dispatch once per day. In Section 3.5, we study the setting in which one vehicle dispatches multiple times per day. In Section 3.6 we perform various computational exercises, applying our modeled results in a realistic setting. Section 3.7 contains concluding remarks. An appendix contains material omitted from the main body.

3.1.1 Literature Review

The majority of the SDD literature has focused on operational problems, in which system features are fixed and a system manager must determine an optimal policy to guide decision-making over a short horizon (typically a single service day). Such works typically focus on vehicle dispatching and routing as customer information is dynamically revealed. Proposed solutions are often compared to offline heuristics or current best practices. Specific problems considered in the literature include the same-day delivery problem for online purchases [55, 16] and the dynamic dispatch waves problem [11, 12, 56]. Other works integrate autonomous vehicles [57], drones [58, 35], and additional extensions [15, 59]. Operational SDD problems are closely related to the broad problem classes of stochastic VRPs [60, 61] and dynamic VRPs [62, 63].

Operational SDD problems are often modeled as mixed-integer linear programs (MILPs), Markov decision processes (MDPs), or a combination thereof. Because of their underlying stochasticity and extremely large decision spaces, these problems are generally solved without optimality guarantees; solution techniques include approximate dynamic programming [11, 15], neighborhood search [55], and tailored heuristics [58]. Such models may be sufficient for day-to-day operational usage. However, it is difficult to perform tactical SDD system design with these operational models as they often require heavy computational power for solving even moderately-sized problems to suboptimality over a single set of design parameters. While simulation is an option for gaining tactical insights [64, 33], the lack of transparency and interpretability in simulation-based methods reveals a need for analytical approaches to SDD tactical design problems.

While we are not aware of any literature directly studying optimal service region selection for SDD systems, a few papers examining operational problems have considered how service regions influence their modeling and results. Notably, [65] formulate an operational SDD model where the dispatcher of the system can choose whether or not to accept orders for SDD servicing into the system, but is constrained to accept orders across dif-

ferent customer partitions at the same rate. The authors note that the benefits of enforcing such fairness constraints come at the cost of lowering the total quantity of served orders. Another work which also allows a dispatcher to accept or reject orders for SDD servicing articulates that as time progresses during the service day, the operator is less likely to accept orders for customers living further away from the depot if the dispatcher is to maximize the number of orders served [66].

Seminal works in the area of continuous approximations for vehicle routing show that the length of vehicle tours can be functionally approximated by the number of stops in the tour as well as the specific service region area from which demand points originate. The foundational Beardwood-Halton-Hammersley (BHH) Theorem [24] states that the length of an optimal traveling salesperson problem (TSP) tour over n points in a region of area A approaches $\beta\sqrt{An}$ as n grows, where β is a region- and metric-dependent constant.

Ensuing studies analyze BHH-type approximations of vehicle tour lengths in various settings [67, 25, 26, 68]. Various works have focused on empirical estimation of BHH routing constants on stylized regions [36, 27] and real-world road networks [69]. Comprehensive surveys of the continuous approximation literature, from fundamental works to recent results and applications, are given by [70, 23].

This chapter is most closely related to the works of [54, 53], which we believe to be the first SDD studies to leverage continuous approximations for tactical system design. Similar to these works, we also approximate demand for SDD continuously over time and approximate the routing time to serve orders as a function in the number of orders on the delivery tour. In [53], the authors assumed that orders for SDD could only arrive into the system from a predefined fixed service region until some defined cutoff time. As such, their objective was to minimize the total routing time to serve all of the SDD orders. In a similar setting, again with a predefined fixed service region and cutoff time, [54] seek to minimize the total number of vehicles needed to serve SDD orders assuming the region is to be partitioned into single-vehicle zones. In contrast, here we maximize the number

of orders served while incorporating an additional tactical design dimension: we allow the service region and order cutoff time to be chosen as decision variables.

3.2 Model Formulation

We consider systems in which a single depot serves as a stocking and dispatching location, with the objective of serving as many SDD orders as possible within a service day. Orders are fulfilled via the depot's vehicle fleet, but only after each order is realized into the system. Without loss of generality, once an order arrives into the system, it becomes immediately eligible for delivery. A dispatcher chooses the service region in which orders accumulate from, knowing that any realized order must be served by the end of the service day. Additionally, every vehicle in the fleet must arrive back at the depot by the end of the service day. The modeling of such systems, including their relevant objectives, constraints, alongside several additional assumptions, is outlined in the proceeding paragraphs.

Service Day: The first time any vehicle can leave the depot is denoted by time $t = 0$, and all vehicles must be back at the depot by the end of the service day, time $t = T$. For convenience, we assume without loss of generality that $T = 1$ and all other parameters are appropriately scaled. Therefore, we often refer to t as a proportion of the service day.

Service Region: At the start of the service day, and after each dispatch, the dispatcher must determine a region for which SDD orders will accrue from until the next vehicle dispatch. In general, the service regions act as unconstrained variables for the dispatcher to leverage, but may be further constrained as desired.

Customer Orders: Demand for SDD accumulates at a rate of λ orders per time unit per area unit starting at time $t = 0$. For tactical planning purposes, demand is modeled continuously over time and homogeneously over area. Furthermore, the demand rate

is modeled as a constant over the entire service day. Any accumulated demand must be serviced by the end of the service day.

Vehicle Restrictions: Vehicles are not explicitly constrained by capacity nor are they restricted to carry an integer number of orders. It is assumed that at each time of dispatch, a vehicle leaves the depot with all of the accumulated orders since the previous dispatch for the chosen service region. The fleet is comprised of m homogeneous vehicles.

Routing Time Function: The time it takes for a vehicle to serve $n \in \mathbb{R}_{\geq 0}$ orders in a region of area, A , and return to the depot is given by the function $f(A, n) = c_0 \sqrt{An}$, where c_0 is some known positive constant. Equivalently, we can define the routing time function as $f(A, \tau) = cA\sqrt{\tau}$ where τ is accumulation time since last dispatch and $c = c_0\sqrt{\lambda}$.

Here we take a quick aside to discuss some of the continuously approximated features of our model. Often when modeling customer demand, a fluid rate is used in lieu of more complicated discrete event modeling in order to study aggregate level behavior. The results of [53] show that for SDD modeling, serving all of the accumulated demand at a depot at the time of each dispatch to be structural property of optimal dispatching solutions in order to gain greatest routing efficiencies. Operationally it may be better to leave some orders for future dispatches, but for modeling system design, this is a reasonable assumption [53]. Finally, the modeled routing time function resembles the asymptotic BHH result [24] as referenced in the literature review in Section 3.1. It has been empirically shown that this functional approximation works well even for a small number of dispatches [36].

The model's objective is to choose a set of feasible accumulation times and service regions in order to serve a maximal number of orders. We formally define the d -th ordered dispatch as a tuple (τ_d, A_d, i_d) , where τ_d defines the order accumulation time for vehicle i_d serving all of the accumulated orders in a region of area A_d . A set of dispatches

$\{(\tau_d, A_d, i_d)\}_{d=1}^D$ is feasible for our model if the following conditions are satisfied:

$$\sum_{\delta=1}^d \tau_{\delta} + f(A_d, \tau_d) \leq 1 \quad \forall d, \quad (3.1a)$$

$$\sum_{\delta=1}^d \tau_{\delta} + f(A_d, \tau_d) \leq \sum_{\delta=1}^{d'} \tau_{\delta} \quad \forall d, d' \text{ s.t. } i_d = i_{d'}, d < d', \quad (3.1b)$$

$$i_d \in \{1, 2, \dots, m\} \quad \forall d, \quad (3.1c)$$

$$A_d \geq 0 \quad \forall d, \quad (3.1d)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.1e)$$

Our problem is to maximize $\sum_{d=1}^D \lambda A_d \tau_d$ over all $D \geq 1$, subject to (3.1a)-(3.1e). Constraint (3.1a) ensures that all vehicles will return to the depot by the end of the service day, while constraint (3.1b) requires that if a vehicle is to be used again, it will return before its next dispatch. Constraint (3.1c) assigns each dispatch to a vehicle in the fleet. Lastly, (3.1d) and (3.1e) are the nonnegativity constraints for the service area and accumulation time variables, respectively.

3.3 One vehicle, one dispatch systems

As a first step in analyzing and utilizing our model we study the family of problem instances where we constrain ourselves to use a single vehicle, and only dispatch it once during the service day. In terms of our model introduced in Section 3.2, we are constraining ourselves to problem instances where $m = 1$ and $D = 1$. Such systems are of interest for many retailers within many network topologies. For instance, leveraging a single vehicle may be of interest for when the broader service region has been pre-partitioned and leveraging a single dispatch may be of interest for certain pricing models and operational considerations. Additionally, as we will see in Sections 3.4 and 3.5, the analysis of one vehicle, one dispatch (1v1d) systems will provide insights on how to tackle larger, more complicated families of problem instances.

Next, we present the model as introduced in Section 3.2 for this specific case. We are interested in solving this problem to optimality by calculating the optimal service area and accompanying optimal accumulation time. The optimization problem for the 1v1d model is given by:

$$\max \quad \lambda A_1 \tau_1 \quad (3.2a)$$

$$\text{s.t.} \quad \tau_1 + cA_1\sqrt{\tau_1} \leq 1, \quad (3.2b)$$

$$A_1 \geq 0, \quad (3.2c)$$

$$\tau_1 \geq 0. \quad (3.2d)$$

Although it is near-trivial to solve this model over the two decision variables outright, we are interested in uncovering structural properties of optimal dispatching policies. One such property that will prove useful for later analyses is that the dimensionality of the decision space can be reduced to only the accumulation time variable. Property 10 forms the basis of this reduction.

Property 10. Given a fixed accumulation time $\tau_1 \in (0, 1]$ for the 1v1d model, the service area which maximizes the number of orders fulfilled is given by: $A_1 = \frac{1-\tau_1}{c\sqrt{\tau_1}}$.

Proof. See Appendix B.1. □

We can now reformulate the problem solely over the variable τ_1 :

$$\max \quad \frac{\lambda}{c} (1 - \tau_1) \sqrt{\tau_1} \quad (3.3a)$$

$$\text{s.t.} \quad \tau_1 \in [0, 1]. \quad (3.3b)$$

This problem can be solved analytically. The optimal solution is found at $\tau_1^* = \frac{1}{3}$, with objective value: $z^* = \frac{\lambda}{c} \frac{2}{3\sqrt{3}}$. Leveraging Property 10, we can determine that $A_1^* = \frac{2}{c\sqrt{3}}$. Figure 3.1 depicts the plot of the scaled objective value, $(1 - \tau_1)\sqrt{\tau_1}$, versus the accumulation time of τ_1 when $\lambda = c = 1$.

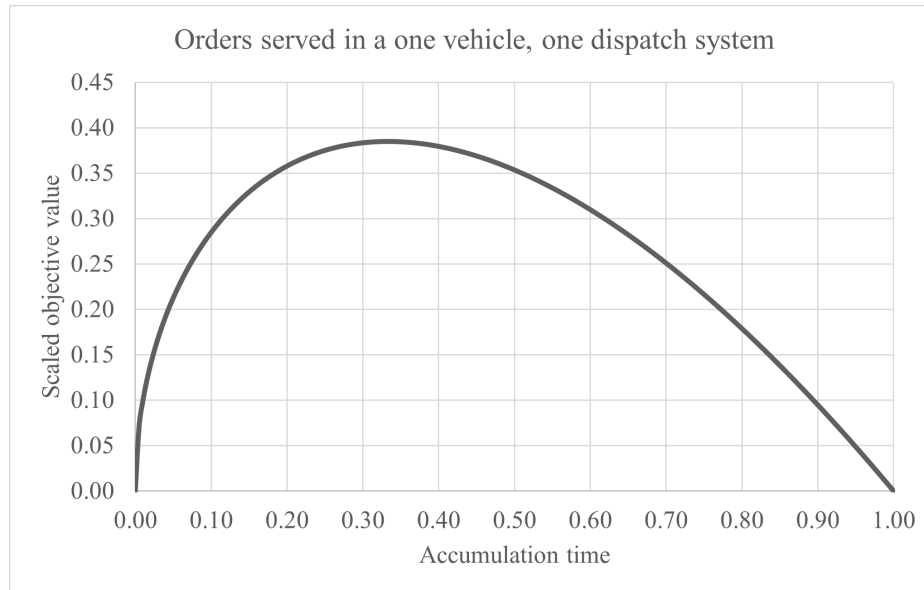


Figure 3.1: Quantity served versus accumulation time for the 1v1d system, $\lambda = c = 1$

This seemingly simple model serves to show how one may approach solving more complicated variants, while also showcasing an interesting design implication for certain SDD systems. Simply put, for a retailer operating a SDD system in a region wishing to only dispatch once, to maximize the number of orders served they should dispatch a vehicle after one-third of the service day is over, servicing an area large enough such that the vehicle arrives back to the depot exactly at the end of the service day. From the plot in Figure 3.1 we see that such a solution is rather insensitive with respect to accumulation time, which gives a dispatcher of an actual operational system based off of these design implications quite a bit of flexibility when dealing with possibly stochastic order arrivals.

Before moving on from our analysis of 1v1d systems, it is important to understand the implications of working with a bounded service region. Some network topology designs may rely on the addition of such a constraint. For example, 1v1d design may be a byproduct of a larger pre-partitioned region which could imply area bounds, the modeling of customer orders and a routing time function may rely on a specific customer density which is bounded geographically, or the SDD retailer may only have authorization to operate in a particular bounded jurisdiction. Mathematically, we can introduce the constraint

$A_1 \leq B$ into the model. Property 11 extends the optimization results of the 1v1d model under this constraint.

Property 11. The optimal dispatching policy for the 1v1d model where the service area is bounded by $A_1 \leq B$ is to serve an area of $A_1^* = \min\{\frac{2}{c\sqrt{3}}, B\}$ after accumulating orders for τ_1^* time where τ_1^* uniquely solves $\tau_1^* + cA_1^*\sqrt{\tau_1^*} = 1$.

Proof. See Appendix B.2. □

3.4 Multiple vehicles, one dispatch each

3.4.1 Model formulation

A natural extension of the 1v1d model is to consider using a fleet of $m \geq 1$ delivery vehicles, where each is dispatched at most once throughout the service day (we denote this as the *mv1d* setting). Figure 3.2 illustrates an example of a three vehicle, one dispatch each policy. In this example, each successive vehicle serves a smaller service region than that of the previous vehicle, while each successive accumulation time increases. Accumulation times are depicted on the horizontal time axis, while the arcs correspond to the routing of each vehicle. Note that each vehicle arrives back to the depot before the end of the service day. Line styles are alternated for visual clarity.

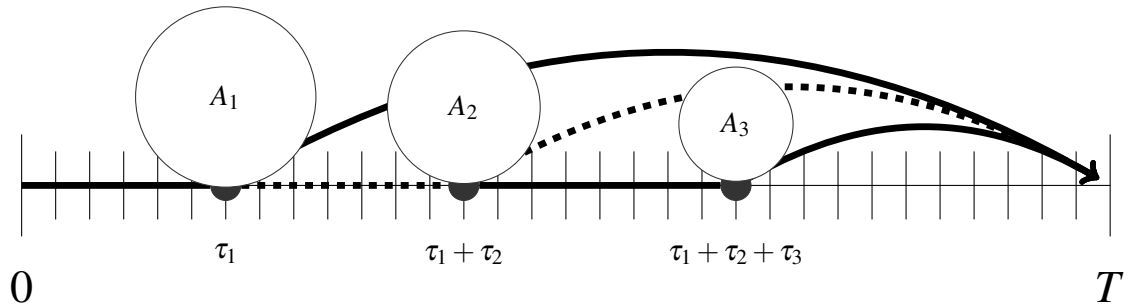


Figure 3.2: An illustration of a 3v1d policy where successive service areas decrease while successive accumulation times increase.

The optimization problem, as first defined in Section 3.2, for the *mv1d* model is given

by:

$$\max \sum_{d=1}^m \lambda A_d \tau_d \quad (3.4a)$$

$$\text{s.t.} \quad \sum_{\delta=1}^d \tau_{\delta} + c A_d \sqrt{\tau_d} \leq 1 \quad \forall d, \quad (3.4b)$$

$$A_d \geq 0 \quad \forall d, \quad (3.4c)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.4d)$$

From this formulation, we can deduce a structural property analogous to that of the 1v1d model. Specifically, we can reduce the dimensionality of this problem by observing that, in order to maximize the number of orders served, given a set of accumulation times, it is a strictly dominant strategy to set each service area such that each vehicle arrives back to the depot exactly at the end of the service day. Property 12 formalizes this observation.

Property 12. Given a set of fixed, positive, accumulation times, $\{\tau_1, \tau_2, \dots, \tau_m\}$, for the m v1d model, the set of service areas which maximize the total number of orders fulfilled are given by $A_d = \frac{1 - \sum_{\delta=1}^d \tau_{\delta}}{c\sqrt{\tau_d}}$ for all d .

Proof. See Appendix B.3. □

Taking the perspective of an individual vehicle, the claims and subsequent proofs of Properties 10 and 12 are very similar. Given a set of accumulation times (which imply a set of departure times), each vehicle, in a sense, is agnostic to the chosen service regions of the other $m - 1$ vehicles. That vehicle is only constrained by arriving back to the depot by the end of the service day, which bounds the size of its service area. An equivalent perspective is that, given a set of accumulation times, each vehicle operates within its own “universe” with a truncated service day. It is important to understand that this does not imply that vehicles can be dispatched in a greedy fashion throughout the service day. The dispatcher must still determine the set of optimal accumulation times, each of which does influence

the departure times of later dispatches. For obtaining $\{\tau_1^*, \tau_2^*, \dots, \tau_m^*\}$ we can solve the following optimization problem:

$$\max \quad \frac{\lambda}{c} \sum_{d=1}^m (1 - \sum_{\delta=1}^d \tau_\delta) \sqrt{\tau_d} \quad (3.5a)$$

$$\text{s.t.} \quad \sum_{d=1}^m \tau_d \leq 1, \quad (3.5b)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.5c)$$

However, this problem is still non-linear and non-convex. As such, we next develop an efficient solution method by analyzing structural properties of the model.

3.4.2 Model analysis and tactical design properties

To begin our analysis of the $mvld$ model, we first explain an important structural feature of optimal solutions which allows a dispatcher to efficiently solve for optimal dispatching policies. We note that each time a vehicle leaves the depot, the problem becomes “memoryless” with respect to that vehicle. That is, after the first dispatch, the remaining problem with $m - 1$ vehicles behaves as if one started the dispatch day with $m - 1$ vehicles, but with a reduced (i.e., truncated) service day. Therefore, it is plausible that this family of problems would have a recursive structure to finding optimal dispatch policies. Theorem 13 formalizes this observation and the implied solution method.

Theorem 13. *Given the optimal dispatch policy $\{(\tau_{m,d}^*, A_{m,d}^*)\}_{d=1}^m$ for the $mvld$ model with an objective value of $z_m = \frac{\lambda}{c} \sum_{d=1}^m (1 - \sum_{\delta=1}^d \tau_{m,\delta}^*) \sqrt{\tau_{m,d}^*}$, we can formulate the $(m + 1)vld$ optimization problem as:*

$$\max_{0 \leq \tau_{m+1,1} \leq 1} \quad \frac{\lambda}{c} (1 - \tau_{m+1,1}) \sqrt{\tau_{m+1,1}} + (1 - \tau_{m+1,1})^{1.5} z_m.$$

Furthermore, we can perform the following update equations to the $mvld$ optimal policy

to obtain the $(m + 1)$ th optimal policy:

$$\tau_{m+1,d}^* \leftarrow (1 - \tau_{m+1,1}^*) \tau_{m,d-1}^* \quad \forall d \geq 2,$$

$$A_{m+1,d}^* \leftarrow (1 - \tau_{m+1,1}^*)^{0.5} A_{m,d-1}^* \quad \forall d \geq 2,$$

$$A_{m+1,1}^* = \frac{1}{c} (1 - \tau_{m+1,1}^*) (\tau_{m+1,1}^*)^{-0.5}.$$

Proof. See Appendix B.4. □

We present the optimal dispatching solutions for $m = 1$ to $m = 4$ vehicles in Table 3.1. For simplicity, we have chosen to display the results for $\lambda = c = 1$. Each row of the table consists of four columns: accumulation time (τ_d^*), area served (A_d^*), orders served ($\lambda A_d^* \tau_d^*$), and time of day ($\sum_{\delta=1}^d \tau_\delta^*$), each rounded to four decimal places. It was determined that the maximum number of orders served is equal to 0.3849, 0.6337, 0.8281, and 0.9919 respectively for these cases.

Table 3.1: Optimal dispatching policies for the $mv1d$ model for up to $m = 4$ vehicles; total quantity served is tabulated and bolded for each model

	Vehicle	Accum. Time	Area Served	Orders Served	Time of Day
$m = 1$	1	0.3333	1.1547	0.3849	0.3333
	Total			0.3849	
$m = 2$	1	0.1841	1.9012	0.3501	0.1841
	2	0.2720	1.0430	0.2836	0.4561
	Total			0.6337	
$m = 3$	1	0.1243	2.4842	0.3087	0.1243
	2	0.1613	1.7791	0.2869	0.2855
	3	0.2382	0.9760	0.2324	0.5237
	Total			0.8281	
$m = 4$	1	0.0928	2.9772	0.2764	0.0928
	2	0.1127	2.3661	0.2667	0.2056
	3	0.1463	1.6945	0.2479	0.3519
	4	0.2160	0.9296	0.2008	0.5679
	Total			0.9919	

Note that we began with a formulation over $2m$ decision variables at the beginning of Section 3.4, reduced that decision space in half via Property 12, and used Theorem 13 to reduce the problem to a single decision variable. There are additional properties about the $mv1d$ model worth noting. The following properties are both interesting from mathematical point of view and of interest for tactically designing SDD systems.

Property 14. As the number of vehicles m in the model increases, the optimal accumulation time of the first vehicle $\tau_{m,1}^*$ strictly decreases, and the maximal number of orders z_m served strictly increases. Furthermore, as $m \rightarrow \infty$, $\tau_{m,1}^* \rightarrow 0$ and $z_m \rightarrow \infty$. Specifically, z_m increases as $\Theta(\sqrt{m})$.

Proof. See Appendix B.5. □

Property 15. There is a strictly decreasing marginal gain in additional orders served in the *mv1d* model when adding an additional vehicle. That is, $(z_{m+2} - z_{m+1}) < (z_{m+1} - z_m)$ for all $m \geq 1$. Furthermore, as $m \rightarrow \infty$, $(z_{m+1} - z_m) \rightarrow 0$.

Proof. See Appendix B.6. □

Property 16. In the optimal dispatch policy for the *mv1d* model, accumulation times are strictly increasing while service areas are strictly decreasing; that is, $\tau_{m,1}^* < \tau_{m,2}^* < \dots < \tau_{m,m}^*$ and $A_{m,1}^* > A_{m,2}^* > \dots > A_{m,m}^*$.

Proof. See Appendix B.7. □

These results confirm intuition about how a SDD system should operate from the perspectives of both system managers and consumers. The empirical results in Table 3.1 show that total number of orders increased by 65% when moving from a one-vehicle to a two-vehicle system, and by 31% when moving from a two-vehicle to a three-vehicle system. Property 15 states that this decrease in marginal benefit is to be expected, and if the fleet size is sufficiently large, adding a single vehicle will not entail any practical benefit. A dispatcher operating a *mv1d* could therefore leverage these models to perform a cost-benefit fleet sizing analysis by comparing SDD revenue versus the cost of maintaining the vehicle fleet. Another useful design implication, via Property 16, is that optimal dispatch areas are strictly decreasing. Therefore, from a customer's perspective, the offering of SDD will not fluctuate during the course of a service day. For example, a customer will not experience a loss of SDD availability during the middle of a service day, only to have that service become available again at a later time.

3.4.3 Value of varying service regions

Thus far, we have modeled various SDD systems where a dispatcher was allowed to vary the service region between dispatches without constraint. We saw from the optimal dis-

patching solutions in Section 3.4.2 that it is advantageous to systematically reduce the service area throughout the service day. Although such solutions maximize the number of orders fulfilled via SDD, the system operators may consider time-varying service regions to be inequitable. In this subsection, we quantify the value (in terms of orders served) of allowing the dispatcher to vary the service area throughout the service day, rather than choosing a fixed service region for all dispatches. The following mathematical program models the *mv1d* problem modified with the additional constraint $A = A_1 = A_2 = \dots = A_m$:

$$\max \sum_{d=1}^m \lambda A \tau_d \quad (3.6a)$$

$$\text{s.t.} \quad \sum_{\delta=1}^d \tau_{\delta} + cA\sqrt{\tau_d} \leq 1 \quad \forall d, \quad (3.6b)$$

$$A \geq 0, \quad (3.6c)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.6d)$$

In two of our previous optimization models, (3.2a)-(3.2d) and (3.4a)-(3.4b), we were able to reduce the problem dimensionality by observing that, for a given set of accumulation times, the service areas should be chosen such that they were as large as possible in order to serve a maximal number of orders. For this model variant, we no longer have that flexibility; however, it is important to observe that it is still a dominant dispatching policy to have all of the vehicles return to the depot exactly at the end of the service day. That is, for any given fixed service area, all of the constraints given by (3.6b) should hold at equality. Property 17 formalizes this observation.

Property 17. Consider a variant of the *mv1d* model where each service region serves a fixed area of size $A > 0$. The set of accumulation times which then maximize the total number of orders served are such that $\sum_{\delta=1}^d \tau_{\delta} + cA\sqrt{\tau_d} = 1$ for all dispatches d . Equivalently, $\tau_d = R_d + \frac{cA}{2} \left(cA - \sqrt{(cA)^2 + 4R_d} \right)$ for all dispatches d , where $R_1 = 1$ and

$$R_d = 1 - \sum_{\delta=1}^{d-1} \tau_\delta \text{ for all } d \geq 2.$$

Proof. See Appendix B.8. □

We are able to leverage Property 17 to reduce the search space for optimal dispatching policy solutions to this model variant. What remains is to determine an optimal service area size, A^* , then use Property 17 to calculate the associated accumulation times. Unfortunately, there is no known, scalable method for solving this optimization problem. We present the numerically computed solutions for one to four vehicles in Table 3.2. For purposes of comparison with Table 3.1, we have display results, rounded to four decimal places, for $\lambda = c = 1$. Additionally, Figure 3.3 depicts a plot of the maximum amount of orders that can be served versus fixed service area for each fleet size. It can be shown that even for the one-vehicle variant, the maximal number of orders served is not concave (even though it may appear so) with respect to the fixed service area, which leads to the difficulty in solving such problems at scale.

Table 3.2: Optimal dispatching policies for the fixed-area $mv1d$ model for up to $m = 4$ vehicles

	Vehicle	Accum. Time	Area Served	Orders Served	Time of Day
$m = 1$	1	0.3333	1.1547	0.3849	0.3333
	Total			0.3849	
$m = 2$	1	0.2454	1.5232	0.3738	0.2454
	2	0.1550	1.5232	0.2360	0.4004
	Total			0.6099	
$m = 3$	1	0.1953	1.8207	0.3556	0.1953
	2	0.1352	1.8207	0.2462	0.3305
	3	0.0984	1.8207	0.1791	0.4289
	Total			0.7809	
$m = 4$	1	0.1626	2.0766	0.3377	0.1626
	2	0.1195	2.0766	0.2482	0.2821
	3	0.0911	2.0766	0.1892	0.3732
	4	0.0715	2.0766	0.1485	0.4447
	Total			0.9235	

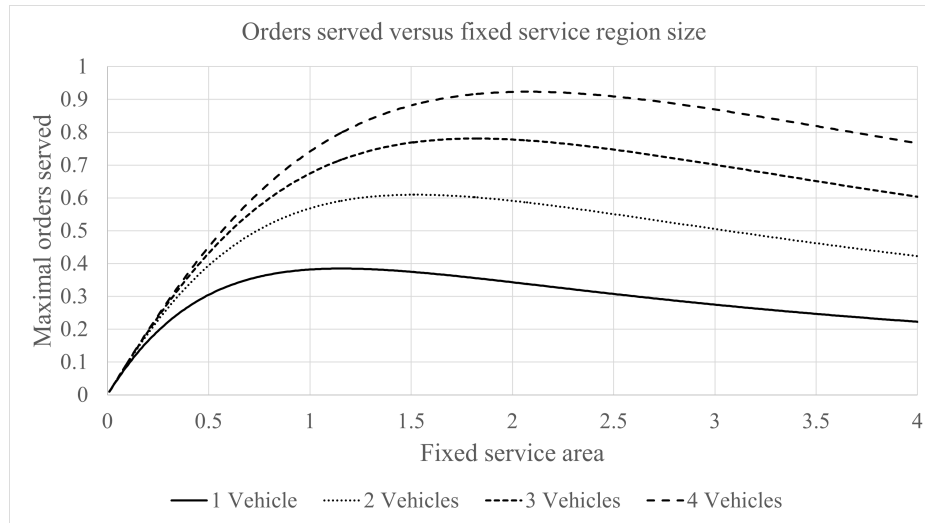


Figure 3.3: Graphical depiction of the maximum quantity served versus fixed service area size for up to $m = 4$ for the fixed-area $mv1d$ variant

Comparing the results of Table 3.1 and Table 3.2 allow us to quantify the cost of enforcing equal areas. As expected due to the additional constraint, the total number of orders fulfilled decrease in the fixed-area model. For two-vehicle systems, there is a 3.9% increase in orders served to be had when service regions can change between dispatches. This gap increases to 6.0% and then to 7.4% for three- and four-vehicle systems, respectively. There are also observable differences in the areas served and accumulation times used in the optimal dispatching policies. What may be important for some SDD dispatchers is understanding how their service affects customers who demand SDD servicing. It is important to understand that there are effects experience by customers are not universal. Some customers will experience shorter time windows in which they are offered SDD while others will experience longer time windows. Some customers may even lose SDD servicing all together. Such considerations and their relative importance will vary depending on the service provider, and continuously approximated models such as ours offer a methodology for which such implications can be better understood.

3.4.4 Impacts of constrained service regions

By the previously discussed structural properties of the $mv1d$ problem, it is clear that $A_{m,1}^*$ tends to infinity as the number of vehicles m grows. From a system design perspective, such behavior limits the usefulness of the $mv1d$ model in practice. For this reason (and for the previously discussed reasons that motivated Property 11), we now consider the $mv1d$ model where all service areas are bounded above by a given $B > 0$.

As before, the resulting optimization problem is non-linear and non-convex; therefore, we are interested in an efficient solution method. One approach is to solve the unconstrained model for a given m , then post-process the optimal service areas such that they satisfy $B \geq A_d$ for all dispatches d . Such a heuristic provides a feasible dispatch policy; however, the resulting policy is not optimal in general. A natural improvement to this heuristic is to first compare A_1^* in the unconstrained solution to B . By Property 16, as long as the first service region has an area smaller than than the bound B , the unconstrained dispatching solution is also feasible, and therefore optimal, for the constrained problem. If it is instead the case that $A_1^* > B$, we only post-process the first dispatch. We then re-optimize, with respect to the remaining $m - 1$ vehicles, over the remaining service day and repeat the process as required. Theorem 18 states that this intuitive procedure, as formalized in Algorithm 2, indeed produces an optimal dispatching policy.

Theorem 18. *For the $mv1d$ problem with an upper bound $B > 0$ on the service areas, Algorithm 2 returns an optimal policy. Additionally, the optimal areas satisfy $A_{m,1}^* \geq A_{m,2}^* \geq \dots \geq A_{m,m}^*$.*

Proof. See Appendix B.9. □

Algorithm 2 Iterative solution procedure for the constrained $mv1d$ model

- 1: **given** vehicles m , area upper bound B , parameters c, λ
 - 2: **initialize** remaining service day time $T_{var} \leftarrow 1$, remaining vehicles $w \leftarrow m$
 - 3: **while** $w > 0$ **do**
 - 4: calculate the optimal policy $\{(\tau_{w,d}^*, A_{w,d}^*)\}_{d=1}^w$ to the unconstrained $wv1d$ model as given by (3.4a)-(3.4d)
 - 5: Let $\tau_{w,d}^* \leftarrow \tau_{w,d}^* T_{var} \quad \forall d \geq 1$
 - 6: Let $A_{w,d}^* \leftarrow A_{w,d}^* \sqrt{T_{var}} \quad \forall d \geq 1$
 - 7: **if** $A_{w,1}^* \leq B$ **then**
 - 8: $A_{m,m-w+d}^* \leftarrow A_{w,d}^* \quad \forall d \geq 1$
 - 9: $\tau_{m,m-w+d}^* \leftarrow \tau_{w,d}^* \quad \forall d \geq 1$
 - 10: $w \leftarrow 0$
 - 11: **else**
 - 12: $A_{m,m-w+1}^* \leftarrow B$
 - 13: $\tau_{m,m-w+1}^* \leftarrow T_{var} + \frac{cB}{2} \left(cB - \sqrt{(cB)^2 + 4T_{var}} \right)$
 - 14: $w \leftarrow w - 1$
 - 15: $T_{var} \leftarrow T_{var} - \tau_{m,m-w+1}^*$
 - 16: **end if**
 - 17: **end while**
 - 18: **return** optimal dispatching policy $\{(\tau_{m,d}^*, A_{m,d}^*)\}_{d=1}^m$
-

3.5 One vehicle, multiple dispatch case

3.5.1 Model formulation

In Section 3.3 we were able to determine a simple, yet insightful tactical system design for SDD networks where a single vehicle dispatches a single time from the depot. From there, the maximal number of orders served only stands to increase as the vehicle is allowed to

make multiple dispatches within the same service day. For an operationalized system, it can be very difficult to gauge the drawbacks versus benefits of such system designs; our continuous approximation models offer a methodology for comparison.

In this section we analyze the $1vDd$ family of problem instances in which a single vehicle makes D dispatches per day. It is of interest to study such tactical designs so that a dispatcher may better understand what gains in order volume arise with an increase in system complexity. Figure 3.4 illustrates an example of a one vehicle, two dispatch policy. Akin to Figure 3.2, the two circles suggest the relative size of the chosen service area while the curved arcs corresponds to the time it takes for the vehicle to serve accumulated orders. In this example, the service area reduces in size during the service day while the accumulation time increases from the first to the second dispatch. Note that the vehicle arrives back to the depot before the end of the service day.

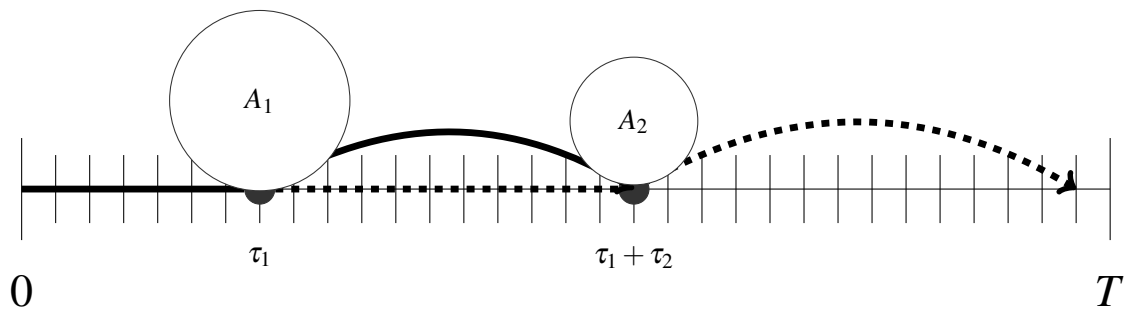


Figure 3.4: A $1v2d$ policy where the vehicle serves a larger service region at the beginning of the day; the second dispatch, however, allows orders to accumulate at the depot for a longer period of time

Formally, we wish to find an optimal dispatch policy, $\{(\tau_d^*, A_d^*)\}_{d=1}^D$, for the problem:

$$\max \sum_{d=1}^D \lambda A_d \tau_d \quad (3.7a)$$

$$\text{s.t. } \sum_{\delta=1}^D \tau_\delta + cA_D \sqrt{\tau_D} \leq 1, \quad (3.7b)$$

$$cA_d \sqrt{\tau_d} \leq \tau_{d+1} \quad \forall d < D, \quad (3.7c)$$

$$A_d \geq 0 \quad \forall d, \quad (3.7d)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.7e)$$

For the previous models discussed in Sections 3.3 and 3.4 where vehicles were only dispatched once, we were able to reduce the dimensionality of the decision space by the observation that area variables could be set such that each vehicle returned to the depot exactly at the end of the service day. For the 1vDd problem, however, in order for a dispatching policy to be feasible, the vehicle must return back to the depot in time for its next dispatch. Property 19 claims that is a dominant strategy to never have the vehicle idly wait at the depot after the first dispatch.

Property 19. Given a set of fixed, positive, accumulation times, $\{\tau_1, \tau_2, \dots, \tau_D\}$, for the 1vDd model, the set of service areas which maximize the total number of orders fulfilled are given by: $A_d = \frac{\tau_{d+1}}{c\sqrt{\tau_d}}$ for all $d < D$, and $A_D = \frac{1 - \sum_{\delta=1}^D \tau_\delta}{c\sqrt{\tau_D}}$.

Proof. See Appendix B.10. □

The results of Property 19 state that given a set of accumulation times, the operator should always choose large enough service areas such that the vehicle is always busy delivering orders between consecutive dispatches. Having no idle vehicle time during the course of the day after the first dispatch is a property found in other SDD models [54, 12, 53]. Knowing that the operator can choose service areas to maximize orders served given a set of accumulation times, we turn our attention to choosing the best set of accumulation

times for the system. For obtaining $\{\tau_1^*, \tau_2^*, \dots, \tau_D^*\}$ we can solve the optimization problem:

$$\max \frac{\lambda}{c} \left(\sum_{d=1}^{D-1} \tau_{d+1} \sqrt{\tau_d} + \left(1 - \sum_{d=1}^D \tau_d\right) \sqrt{\tau_D} \right) \quad (3.8a)$$

$$\text{s.t.} \quad \sum_{d=1}^D \tau_d \leq 1, \quad (3.8b)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.8c)$$

We numerically calculate and present the optimal dispatching solutions to this model for one to four dispatches in Table 3.3 below. Similar to Tables 3.1 and 3.2, we display the results for $\lambda = c = 1$. The maximum number of orders served based on the number of potential dispatches is equal to 0.3849, 0.4444, 0.4513, and 0.4514, respectively.

Table 3.3: Optimal dispatching policies for the 1vDd model for up to $D = 4$ dispatches.

	Dispatch	Accum. Time	Area Served	Orders Served	Time of Day
$D = 1$	1	0.3333	1.1547	0.3849	0.3333
	Total			0.3849	
$D = 2$	1	0.1111	1.3333	0.1481	0.1111
	2	0.4444	0.6667	0.2963	0.5556
	Total			0.4444	
$D = 3$	1	0.0167	1.3538	0.0226	0.0167
	2	0.1750	1.0953	0.1917	0.1917
	3	0.4582	0.5171	0.2370	0.6500
	Total			0.4513	
$D = 4$	1	0.0002	1.3543	0.0002	0.0002
	2	0.0181	1.3270	0.0241	0.0183
	3	0.1788	1.0844	0.1939	0.1971
	4	0.4585	0.5086	0.2332	0.6556
	Total			0.4514	

From Table 3.3, we observe an increase of 15.5% orders served when considering 1v2d system over a 1v1d system. This is in stark contrast compared to the 1.6% gain in orders served when moving from a 1v2d to a 1v3d system. Additionally, it is important to note, empirically, that the the accumulation time of the first dispatch shrinks as the number of dispatches increases. Once we start using three or four dispatches, we see that the first accumulation time is negligible with respect to the length of the service day. For example, a retailer operating with a 1v3d system would only accumulate orders for the first dispatch for less than ten minutes during a nine-hour service day using the optimal dispatching policy. Additionally, the number of orders served in that first dispatch would only account for 5% of the total orders served between all three dispatches. The near-insignificance of

this first dispatch is even further exacerbated in the four-dispatch solution. As a design implication, it is therefore advisable for an operator to never consider using more than two dispatches in one-vehicle systems.

3.5.2 Model analysis and tactical design properties

Even though one can reasonably assess the relative utility of a one- or two-dispatch system over systems with greater number of dispatches, it may still be a worthwhile exercise to provide a bound as to how well a two dispatch system performs with respect to an infinite dispatch system. As highlighted in Property 14 for the analogous $mv1d$ model, the total number of orders served in an optimal dispatching solution grows infinitely as $m \rightarrow \infty$; one may ask whether similar behavior occurs in $1vDd$ systems. Since the dispatcher can always feasibly set any dispatch to serve an area of size zero for a length of zero accumulation time, the maximal number of orders served is non-decreasing in D . From Table 3.3 we have empirically seen that these marginal benefits seem to diminish greatly for even a small number of dispatches. We next introduce Property 20, which leads to Theorem 21, which states that a single vehicle, with an arbitrarily large number dispatches, has a bounded number of orders it can feasibly serve. Furthermore, the one dispatch solution is a 2-factor approximation for such a bound. At this time, it remains to be proven whether this factor is tight.

Property 20. In the optimal dispatch policy for a $1vDd$ model, optimal service areas are bounded with respect to a function of the D -th optimal accumulation time. Specifically, $A_d^* \leq \frac{2}{c} \sqrt{\tau_D^*}$ for all $d < D$.

Proof. See Appendix, B.11. □

Theorem 21. *The maximum number of orders served by any $1vDd$ model is upper-bounded by twice the number of orders served by the optimal $1v1d$ solution ($\frac{\lambda}{c} \frac{4}{3\sqrt{3}}$ orders).*

Proof. See Appendix, B.12. □

3.5.3 Value of varying service regions

We now study the benefits of allowing a dispatcher to vary the service area between dispatches in a 1vDd system. Adding the fixed area constraint $A = A_1 = A_2 = \dots = A_D$ to the original 1vDd model, we arrive at the following optimization problem:

$$\max \sum_{d=1}^D \lambda A \tau_d \quad (3.9a)$$

$$\text{s.t.} \quad \sum_{\delta=1}^D \tau_{\delta} + cA\sqrt{\tau_D} \leq 1, \quad (3.9b)$$

$$cA\sqrt{\tau_d} \leq \tau_{d+1} \quad \forall d < D, \quad (3.9c)$$

$$A \geq 0, \quad (3.9d)$$

$$\tau_d \geq 0 \quad \forall d. \quad (3.9e)$$

Qualitatively, let us consider the implications of keeping the vehicle idle at the depot between successive dispatches. Constraint (3.9c) in our model enforces that the vehicle arrives back to the depot before the next dispatch. Now, consider a dispatch policy where the vehicle arrives back to the depot with some positive time left before its next dispatch (or at the end of the service day after a final dispatch), then during that dispatch it would have been feasible (and therefore preferable) to accumulate orders for a longer amount of time. Property 22 formalizes this observation.

Property 22. Consider a variant of the 1vDd model where each service region serves a fixed area. Given a fixed total accumulation time $N \in (0, 1)$, the area A and set of accumulation times τ_1, \dots, τ_D which maximize the total number of orders served are such that, after the first dispatch, the vehicle will never wait idly at the depot again. Furthermore, the vehicle will return to the depot exactly at the end of the service day. Mathematically, $cA\sqrt{\tau_d} = \tau_{d+1}$ for all $d < D$, and $N + cA\sqrt{\tau_D} = \sum_{\delta=1}^D \tau_{\delta} + cA\sqrt{\tau_D} = 1$.

Proof. See [54]. □

Property 22 allows us to reduce the search space for optimal dispatching policy solutions to those for which (3.9b) and (3.9c) hold at equality. In other words, given some candidate total accumulation time N , we do not need to observe all possible sets of areas and accumulation times, only the ones which satisfy the conditions given in Property 22. With this knowledge and the algorithm given in [54], we are able to calculate the results for the fixed-area 1vDd model for up to four dispatches in Table 3.4. Again, results will be displayed for $\lambda = c = 1$ for the sake of comparison. The maximum number of orders served based on the number of potential dispatches is equal to 0.3849, 0.4305, 0.4356, and 0.4358 respectively.

Table 3.4: Optimal dispatching policies for the fixed-area 1vDd model for up to $D = 4$ dispatches

	Dispatch	Accum. Time	Area Served	Orders Served	Time of Day
$D = 1$	1	0.3333	1.1547	0.3849	0.3333
	Total			0.3849	
$D = 2$	1	0.1183	0.9599	0.1136	0.1183
	2	0.3302	0.9599	0.3169	0.4485
	Total			0.4305	
$D = 3$	1	0.0219	0.9006	0.0197	0.0219
	2	0.1332	0.9006	0.1199	0.1551
	3	0.3287	0.9006	0.2960	0.4837
	Total			0.4356	
$D = 4$	1	0.0006	0.8957	0.0006	0.0006
	2	0.0226	0.8957	0.0202	0.0232
	3	0.1346	0.8957	0.1206	0.1579
	4	0.3287	0.8957	0.2944	0.4865
	Total			0.4358	

Comparing the results of Table 3.4 with the results of Table 3.3, we observe that allowing service regions to vary in size yields a 3.2% gain in the maximum number of orders served for two dispatch systems. Furthermore, the optimal fixed service area is slightly less than half of the average optimal service area sizes of the unconstrained model. One of the larger differences is that the maximal cutoff time for SDD ordering over all customers would increase from 44.9% of the service day to 55.6% of the service day when moving from a fixed area design to a variable area design. Another implication of these results is how we observe the marginal benefits of adding additional dispatches. Previously, in the variable-area model, we observed a 15.5% and a 17.3% marginal increase in orders served when adding one additional or two additional dispatches on top of a one dispatch solution. In the fixed area variant, these marginal gains decrease to 11.8% and 13.2%, respectively. For a provider of SDD fulfillment, it is important to understand these differences in system design and their effects on the business, both operationally and on the customer-facing side. These tactical design models allow these providers to quantify design implications at an aggregate level with a high level of fidelity.

3.6 Computational Experiments

In this section, we provide numerous examples of our models implemented on more realistic sets of input parameters. Although the results depicted in Tables 3.1-3.4 can be scaled to varying inputs of λ and c , it is useful to walk through how a SDD provider may arrive at such values and then leverage our models and results. For selected examples, we simulate service days where customers arrive via a Poisson point process and true dispatch durations are calculated exactly with Traveling Salesman tours. We now describe a computational experiment that empirically demonstrates the practical application of our theoretical results and the relative accuracy of our continuously approximated model.

Consider a hypothetical provider of SDD fulfillment wishing to study the impact of service region design and fleet sizing choice for their network. Orders are fulfilled out of a

single depot, serving a circular service region of size to be determined. What is known is that the service day starts at 9:30 am, and ends at 6:30 pm. Additionally, in order to gain an understanding of a realistic input for λ , we consider the metropolitan area of Minneapolis, Minnesota. The metropolitan statistical area surrounding Minneapolis has a population density of 3,400 people per square mile [71], while 76.9% of the population in the state of Minnesota are 18 years of age or older. If we assume that on average 1.0% of the adult population orders from this fictitious SDD provider 6 times a year, then we can derive a value of $\lambda = 0.0478$ orders per area unit per hour for SDD demand.

As discussed in Sections 3.1 and 3.2, there has been various studies on determining an appropriate routing time function constant c_0 , for various network topologies [24, 25, 26, 27]. In the spirit of [27], here we design a calibration experiment to determine an appropriately sized c_0 , and subsequently c , for our fictitious SDD provider. To do this, we randomly sample locations within a unit circle and then calculate an optimal TSP tour serving these points stemming from the center of the circle (representing the depot) according to a euclidean distance metric. We complete this calibration for sample sizes ranging from 10 to 50 randomly generated points, which represents a reasonable number of packages to be dispatched on each vehicle, for 30 repetitions each (1230 samples in total). Using a simple linear regression model of TSP tour length versus the square-root of the number of stops, it was determined that $c_0 = 0.8216$ with a 95% confidence range of ± 0.0031 . Finally, assuming a vehicle speed of 15 miles per hour and our modeled choice for λ , it is calculated that $c = .0120$ for this experiment. Figure 3.5 illustrates the sampled value for the routing coefficient, c_0 , for each of the 1230 simulated tours.

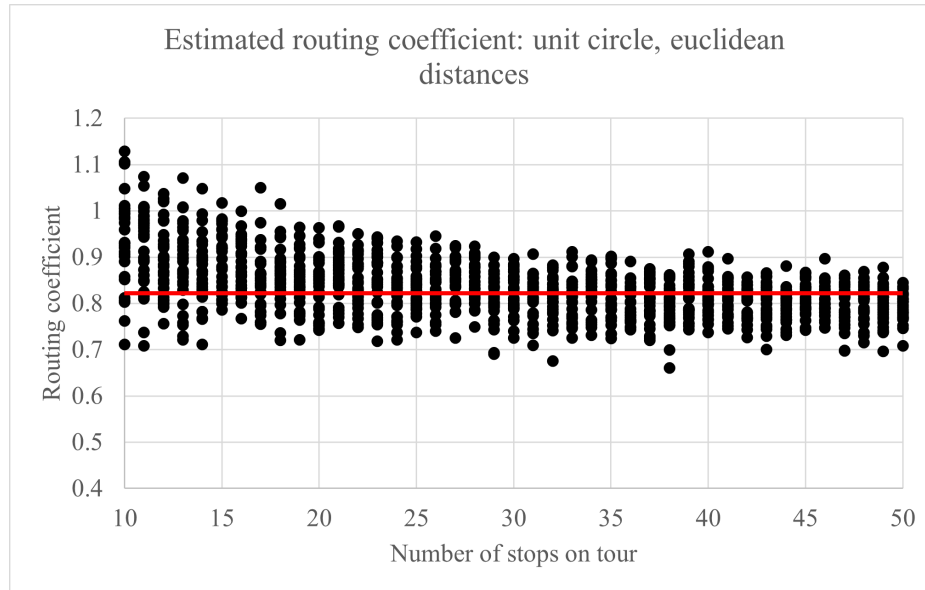


Figure 3.5: Graphical results of the routing function constant calibration for a unit circle geometry with euclidean distances. A straight line depicts the fitted routing constant.

To study the efficacy of our continuously approximated modeling, we will define more realistic simulation assumptions going forward. For order arrivals, we use a Poisson arrival process in lieu of a fluid arrival rate. The Poisson process will use the rate of $\lambda = 0.0478$ orders per square mile per hour. Additionally, once these points are randomly generated in the circular service region in order to serve them the total routing time is calculating by solving a TSP using a standard integer programming formulation. All experiments leverage Gurobi 9.1.1 as an optimization solver, while being coded in Python 3.6.3.

3.6.1 Two vehicles, one dispatch each simulation

This SDD provider first starts their topology design study by considering the usage of two vehicles, where each vehicle is dispatched once (2v1d). Following the results of Table 3.1, in order to maximize the number of orders served, this SDD provider should set service areas of 476 square miles, and 261 square miles respectively. For a circular service region, these values would correspond to having a 12.3 mile and a 9.1 mile radius design, respectively. The first dispatch would be expected to leave at 11:10 am serving 37.73 orders

while the second dispatch would be expected to leave at approximately 1:36 pm serving 30.57 orders.

As for our operationally simulated benchmark, we first describe the process of translating this optimal tactical dispatching policy into an operationalized design. First we fix the service region sizes as prescribed by the optimal dispatch policy of the tactical model. This is a reasonable assumption as service regions need to be fixed before the service day begins so the SDD provider can know beforehand the particular population to offer such services. For each dispatch, orders begin to accumulate via the Poisson arrival process throughout the service day. As orders arrive into the system, the dispatcher re-calculates an optimal TSP tour in order to serve all built-up demand. In spirit of Property 12, the dispatch allows orders to accrue until a time until the vehicle must be dispatched immediately to serve all accumulated orders and still make it back to the depot before the end of the day. If an order arrives into the system that would cause the vehicle to arrive back to the depot after 6:30 pm, that order is not offered same-day servicing, however, if this order is able to be served within the service region of the second vehicle, it will be added to that vehicles workload. We simulated this process for 300 independent service days and summarise the results in Table 3.5. Each operational benchmark statistic reports a sampled mean value with a 95% confidence interval. These results highlight, at least empirically, that our tactically designed system has an operational behavior closely resembled to what was modeled. Even with a more realistic order arrival process and actual TSP tour calculations, these operational benchmarks serve within 1.0% the approximated number of orders and within 1.5% of the total approximated routing time of the tactical design policy.

Table 3.5: Computational experiment results for a realistic 2v1d design.

	Tactical	Operational
First Dispatch Quantity	37.73 units	38.15 units (± 0.41)
First Dispatch Time	440.56 min.	433.51 min. (± 1.53)
Second Dispatch Quantity	30.57 units	30.24 units (± 0.46)
Second Dispatch Time	293.71 min.	290.85 min. (± 2.03)
Total Quantity	68.30 units	68.39 units (± 0.62)
Total Routing Time	734.27 min.	724.36 min. (± 2.94)

For a second experiment with two vehicles, we study the effects of a fixed area design. Applying the results of Section 3.4.3, we are able to calculate that an optimal service region design would serve a circular area of 382 square miles, having a radius of 11.0 miles. The continuously approximated optimal dispatch policy can be found in Table 3.6. Also in Table 3.6 are the results of another 300 simulated service day instance with the same operational benchmark as before, but with updated service areas.

Table 3.6: Computational experiment results for a realistic fixed-area 2v1d design.

	Tactical	Operational
First Dispatch Quantity	40.29 units	40.88 units (± 0.42)
First Dispatch Time	407.48 min.	401.18 min. (± 1.64)
Second Dispatch Quantity	25.44 units	24.40 units (± 0.36)
Second Dispatch Time	323.79 min.	322.74 min. (± 1.78)
Total Quantity	65.73 units	65.28 units (± 0.58)
Total Routing Time	731.27 min.	723.92 min. (± 2.95)

As in the previous Table, here we note that the average performance of the operationalized system closely resembles that of the tactical design. This supports the usage of such

continuously approximated models for use of modeling the average behavior in SDD systems, especially as it pertains to comparing service region design. From these results our fictitious provider of SDD fulfillment should be able to determine the more desirable system for their business, a fixed-area design or a variable-area design. Once this topology design is chosen, the vehicle dispatcher can leverage operationally focused SDD modeling, such as in [14, 16, 12, 11, 13], in order to manage the day to day dispatching of the vehicles in an optimal fashion.

3.6.2 One vehicle, two dispatches simulation

Now suppose our fictitious retailer is interested in studying the outcomes of single vehicle systems. We can scale the results of Table 3.3 to this, $\lambda = 0.0478$, $c = 0.0120$ setting and determine an optimal topology design of 289 square miles for a 1v1d design and 334 square miles followed by 167 square miles for a 1v2d design. The optimal dispatch policy for the 1v1d system should expect to serve 41.48 orders in a day while the 1v2d system should expect to serve a total of 47.90 orders.

Next we describe the operationally simulated benchmark for these one vehicle systems. For both a single dispatch and a two dispatch design, we fix the service region to that calculated in the tactical design policy. If the vehicle is to serve only one dispatch, following the results of Property 10, the vehicle is to wait at the depot, accruing orders subject to the Poisson order arrival process, until a point in time at which the vehicle must dispatch in order to make it back to the depot by the end of the service day. This possibly includes the necessity to reject an incoming order for same-day fulfillment if it would not be feasibly achievable to serve that order along with the existing orders in the system. In that case, the dispatcher dispatches the vehicle immediately without the last order arrival. In the two dispatch setting the dispatcher is mindful that after the vehicle dispatches the first time, it must continue to accrue orders for the smaller (167 square mile) service region. In spirit of Property 19, the vehicle will dispatch immediately upon returning from the first

dispatch, serving all accumulated orders for the second service area that built up while the first service area was being served. As orders accrue for the first dispatch, the dispatcher continuously solves updated TSP routing times in order to determine the amount of time before the vehicle is back at the depot and ready to serve the second service region. They are also able to approximate the number of orders that will have built up for this second dispatch in that time, and the expected routing time of those number of orders for the second service area. Thus when the total routing time of the first set of actual accumulated orders plus the expected time to route the second set of orders equals the remainder of the service day, the vehicle will be sent out to serve its first dispatch. As for orders that then start accruing for the second dispatch, the dispatcher will again continually update a TSP routing time and will stop accepting orders in order to have the vehicle return to the depot by the end of the service day.

As with the two vehicle design experiments, 300 independent service days are simulated and the results are summarised in Tables 3.7 and 3.8 below. Both operational benchmarks report orders served to within 1.5% of their continuously approximated counterparts, while the total routing time is within 2.0% for both experiments. These results highlight the efficacy of the tactical design modeling and ability to approximate more realistic settings.

Table 3.7: Computational experiment results for a realistic 1v1d design.

	Tactical	Operational
First Dispatch Quantity	41.48 units	41.95 units (± 0.44)
First Dispatch Time	360.00 min.	353.22 min. (± 1.59)

Table 3.8: Computational experiment results for a realistic 1v2d design.

	Tactical	Operational
First Dispatch Quantity	15.97 units	14.97 units (± 0.31)
First Dispatch Time	240.00 min.	251.51 min. (± 1.28)
Second Dispatch Quantity	31.93 units	32.40 units (± 0.53)
Second Dispatch Time	240.00 min.	237.98 min. (± 1.91)
Total Quantity	47.90 units	47.37 units (± 0.60)
Total Routing Time	480.00 min.	489.49 min. (± 2.43)

With the results of Tables 3.7 and 3.8 it is possible to weigh the pros and cons of a second dispatch with a single vehicle. The 15.5% gain in orders served with a second dispatch was first calculated in Section 3.5, and is supported by the tactical design policy for this fictitious example. Furthermore, the operational benchmark policies show a 12.9% gain in orders served. While these two statistics are not equivalent, they are certainly correlated and reflect the strength of using a tactical design model such as ours to narrow down topology designs for SDD providers. The results of Tables 3.7 and 3.8 can also be compared to those in Tables 3.5 and 3.6 in order to choose a favorable SDD service design. As stated previously, once a design is to be chosen, more operationally focused models are able to help a dispatcher with day to day dispatching decision making.

3.7 Conclusions

In this chapter, we studied the tactical design of SDD systems in which the service region is allowed to vary over the course of the service day. We leverage continuous approximations on order arrivals and vehicle routing times to capture average-case system behavior. We perform structural analyses for three different fleet settings: one vehicle dispatching once daily, multiple vehicles each dispatching once daily, and one vehicle dispatching multiple

times daily. For each of these settings, we derive theoretical properties and calculate tactical dispatching policies, characterized by service region areas and order cutoff times, which maximize the number of orders served per day. In order to quantify the tangible value of allowing service areas to vary dynamically, we also calculate solutions to the problem of maximizing orders served with fixed service areas for each setting.

Our structural analysis of optimal dispatching policies for these models exhibit some profound design implications for SDD systems. First of all, we were able to show that for many models that there is often a calculable optimal service area based on a set of known accumulation times. Next, while there is a benefit to using multiple vehicles in the same region for delivery, we explicitly quantified that this benefit is marginally decreasing. Additionally, while re-using a single vehicle for a second dispatch may be desirable (15.5% increase in orders served), using more than two dispatches seems to needlessly complicate the dispatching solution. Lastly, we have shown what gains can be expected by allowing a variable service region between dispatches over a fixed one. For example, when service regions are allowed to vary in a two vehicle, one dispatch each system, orders served increases by 3.9%.

Future works may derive optimal order-maximizing vehicle dispatching policies for more general fleet settings. Additionally, just as we use service area variation as a tactical lever to increase profitability and manage demand, future studies may consider using other methods, such as tactical pricing, in conjunction with varying service areas.

CHAPTER 4

CYCLE TIME CONSIDERATIONS FOR CAPACITATED VEHICLE ROUTING PROBLEMS

4.1 Introduction

Logistical systems are typically designed to minimize cost or maximize profitability while meeting objectives for, or constraints on, customer service. Vehicle routing optimization problems are used to model the design of distribution (or collection) routes to move shipments to customers (or to collect them) for some operating period, where each customer and its known demand is served once during the period. Time window constraints may be included when customers can only be served within specific time intervals during the period, but objective functions typically focus only on vehicle dispatch and operating costs and do not prioritize which customers receive service earlier or later within the period. This may not be reasonable.

Routing models also typically consider the scenario where it has already been decided which customers will receive service during the operating period; the duration of the operating period (*e.g.*, a day) is also decided in advance. For example, when used in last-mile logistics the classical Traveling Salesman Problem (TSP) models the sequencing a pre-determined set of customers that are to be served together by a single vehicle for some operating dispatch. Similarly, the well-studied Capacitated Vehicle Routing Problem (CVRP) focuses on how to partition a pre-determined set of customers among multiple vehicles and to determine a sequence for each; again the decision that each of these customers is to receive a single visit (or shipment) in some operating period is pre-determined. Building a design this way is useful in many applications and is often appropriate. However, it does separate some key customer service decisions from transportation operations deci-

sions. Modern last-mile logistics systems may need to visit different delivery (or pickup) locations with varying frequencies or may seek to balance the value of increasing the frequency of visits to locations (and the related customer service impacts) with the costs of doing so.

Balancing of service-based objectives with cost-based objectives is growing in importance, and improving customer service can be used to hedge a service provider's short-term gains with respect to long-term benefits. A common expression in service-oriented supply chain systems is that customer retention is less costly than customer acquisition. Specifically, one recent survey suggests that 82% of companies believe that customer retention is cheaper than customer acquisition [8]. A study from the Harvard Business School shows that increasing customer retention rates by 5% can increase profits by over 25% [9]. A related study shows that loyal customers are 23% more likely to spend more than an average customer [72]. These references highlight the need for intelligently-designed supply chain systems, and in particular last-mile logistics systems, that wisely balance customer service with operating costs.

In this chapter, we will consider modifications to standard models for vehicle routing to include decisions about service frequency in a new way. This work will build from some related approaches that exist in the literature. Some routing models do consider objective functions that help prioritize when certain customers receive service within a given operating period. In traditional scheduling problems, objective functions such as makespan minimization or cycle time minimization are used in models that determine when tasks should be sequenced and completed. In a makespan minimization problem, the service provider seeks to minimize the time gap between the start and end of the processing of a number of tasks. While the minimax makespan objective seeks to balance when tasks are completed, a cycle time minimization problem focuses on reducing the (weighted) average time a unit spends in the system before its processing is completed. The traveling deliveryman and its multiple vehicle extension use objectives or constraints that are similar to these

from scheduling. Again, decisions about how frequently a customer might receive service are ignored. On the other hand, inventory routing problems and some specialized period vehicle routing problems do typically explicitly address some decisions about when to serve customers and how frequently. Most inventory routing problems focus on replenishing customer stocks over a longer horizon to keep them within safe operating bounds. Period vehicle routing problems again plan for multiple operating periods and typically have some customers that require more frequent service than others. While some inventory routing research has attempted to move away from *a priori* time discretization, this work is still emerging and has not explicitly addressed last-mile delivery problems.

In this chapter, we offer a first attempt at studying capacitated vehicle routing problems (CVRPs) which consider the total cycle time of demand units on the service routes in their objectives and constraints. The frequencies with which vehicles visit each particular customer are modeled explicitly, and we do not rely on time-expanded network representations. We consider topologies with a single dispatching node, leveraging a fleet of capacitated vehicles, serving a finite set of demand nodes. We denote these problems as cycle time considering capacitated vehicle routing problems, or CTC-CVRPs. To avoid time discretization, the CTC-CVRP model assumes that customer demand can be modeled as a constant rate of consumption (or production) of units over time. A feasible dispatching solution then must visit each customer node with enough frequency and allocated capacity to satisfy the demand over time. Routing solutions will then also be specified as rates of dispatches over time; thus, service routes will be assumed to be repeated continuously.

Many transportation logistics systems may be modeled in this way. For a non-package movement example, albeit with multiple origin locations, consider the design of repeatable bus routes within a city. The demand at each potential bus stop can be expressed as demand rates over possible destinations, and each route must be designed with capacity in mind. A traditional CVRP-like objective might be to order the stops to minimize the total routing time of the bus on each cycle. In contrast, our objective would be to order

the stops such that the time that an average person spends on the bus would be minimized. Additionally, it is common to have multiple (often identical) service vehicles cycle through the same route in order to minimize waiting time at each stop as well as to meet capacity requirements; the dispatch *headway*, or inverse of the service frequency, refers to the time between consecutive arrivals of vehicles and waiting at each stop is proportional to the headway. We also consider such headway-based waiting times in our routing models. Going forward, we choose to limit the scope of this chapter routing problems for package deliveries, rather than pickup routes or hybrid pickup-and-delivery routes. As an aside, it is often the case that with reasonable modeling assumptions such as symmetric edge costs, calculated pickup or delivery routes can be used interchangeably [73].

After formally defining all of the key inputs to our model, we discuss the mathematical challenges associated with solving for optimal dispatching and routing policies depending on how one may choose to formulate a CTC-CVRP. We then describe a set-partitioning integer programming (IP) formulation, which has been shown to simplify the necessary constraint modeling and optimal policy finding for similar problems. Common solution techniques such as intelligently restricting the routing decision space or implementing a route creation algorithm using column generation techniques are subsequently discussed [74, 75, 76].

The remainder of the chapter is organized as follows. Section 4.1 provides a review of the relevant literature. Then in Section 4.2, we formalize the key input variables and modeling of VRPs with cycle time considerations. In Section 4.3 we show some key modeling properties and discuss their broader design implications. In Section 4.4 we examine a column generating approach for solving linear programming (LP) relaxations of our models, and how such procedures can be used to improve their IP counterparts. Section 4.5 contains various computational exercises which walk through some of the key solution strategies, topology design implications, and key output sensitivities. Lastly, we conclude our work in Section 4.6 with a brief discussion, including a summary of findings and future research

opportunities. An appendix contains material omitted from the main body.

4.1.1 Literature Review

CTC-CVRPs can be broadly classified within the rich family of VRPs. Classical models such as the TSP or CVRP form the basis of many of the models studied by the literature. The distinguishing features of our model in contrast to other models include capacitated vehicles, continuous demand rates, repeatable service routes, and service-based objectives or constraints. For the remainder of the chapter we primarily reference the notion of package movement as a primary problem motivator, but there may be applications of our cycle time focused, rate-based model, to other disciplines such as queuing theory or project scheduling.

The classical objective of a TSP is to minimize total tour length. Such an objective primarily focuses on the goals of the service provider. One interesting adaptation of the TSP in the literature, however, is the Traveling Deliveryman Problem (TDP). As an aside, we note that this model is sometimes referred to as the Traveling Repairmen Problem or even the Minimum Latency Problem. As specified in [77], the objective of the TDP is to find a Hamiltonian circuit which minimizes the sum of all distances along the circuit from a depot node to all vertices along the tour. The motivation behind this problem objective is that it is desirable to minimize the total waiting time it takes to service each demand node. In [78] the authors are among the first to describe the problem and go on to formulate the problem with a Dynamic Program (DP) in the case where all vertices exist on a 2-dimensional line. In [77], the authors formulate the problem as a Integer Program (IP) over a complete directed graph and observe empirically that their formulation could only be solved to optimality for graphs having up to 60 vertices. More recently, in [79], the authors express the TDP as a Mixed Integer Program (MIP) with a strengthened LP relaxation. Although the solution quality of this model improved compared to [77], it still failed to solve larger problem instances due to excessive computational run-times. With the literature suggesting

how difficult this problem is to solve compared to the TSP, which is already itself NP-Hard, authors such as [80] have moved towards specifying effective solution metaheuristics. Before comparing and contrasting the TDP to our CTC-CVRPs, we will examine a natural extension of the TDP.

Much like how the CVRP extends the TSP to capacitated vehicle fleets, the Multiple Traveling Deliveryman Problem extends the TDP. Referred to as the (kTDP) in [81], this model considers larger vehicle fleets for feasibility like a CVRP would, but it also introduces additional vehicles as a means to reduce the total objective time. There is limited published research on the kTDP; most of the work being done in this area focuses on introducing novel formulations which either attempt to improve upon past works with respect to solution speed (CPU times) or solution quality. In [81], the authors propose a MIP formulation which is able to solve simulated problem instances of up to 49 nodes within a time limit of two hours. There is a recent survey paper in [82] which compares and contrasts five different formulations via various computational experiments and theoretical properties. One of the most application-driven pieces of literature came from [83] where the authors formulated the kTDP in the context of minimizing the average waiting time for hospitals to receive blood products from a depot node. The authors of that work chose to implement a two-phase solution to solving the kTDP for a scenario with a two vehicle fleet. In the first phase the authors partitioned the service region into two disjoint sets of hospitals, each to be serviced by a single vehicle. Then in a second phase implemented a TDP solution on each partitioned set separately. While the objectives of the TDP and kTDP take a step in the direction of considering the total cycle time of their routes, there are some notable differences of these models and our CTC-CVRPs. First of all, while the total time it takes to arrive at each demand node appears in the objective, it is not demand-weighted. Furthermore, the TDP and kTDP do not consider rate-based demands which need to be serviced by repeatable cycles. As such, the addition of multiple vehicles to a single route to reduce waiting time at the nodes is inconsequential to these models, unlike in our CTC-CVRPs.

Another closely-related existing problem in research literature to our CTC-CVRPs is the Energy Minimizing Vehicle Routing Problem (EMVRP). Similar to the TDP or kTDP, the objective is dependent on the sum of incremental path lengths it takes to serve each demand node, however in the EMVRP this distance is weighted by the demand at each node. One motivation for this objective is to minimize total fuel consumption (or equivalently energy expenditure) for the vehicles performing the routes, which is proportional to vehicle weight while traveling. In [73] the authors formulate the EMVRP as a MIP for a given fixed fleet size. To illustrate their novel problem, the authors solve the model over two small, simulated problem instances with 3-4 vehicles and 12-16 demand nodes. The authors also solve CVRPs with the same inputs and find (empirically) that even though CVRP solutions return routes which travel 12% less distance, they have to use 4% more energy doing so. The authors in [75] were able to formulate the EMVRP with an IP, which they proved had a stronger LP bound than the model in [73]. The authors then introduce a set partitioning formulation over a restricted set of feasible routes. This set was chosen to be the set of all capacity-feasible routes which forbid any cycles of length at most s , for some parameter $s \geq 1$. Another work in the literature for the EMVRP, [84], restricts the problem to the single vehicle variation. The authors focus on bounding the objective function by leveraging properties of minimum spanning trees and later present an approximation algorithm based on the well-known Christofides's heuristic for symmetric TSPs [85]. A notable flexibility of the EMVRP is that its objective also applies to areas outside of energy consumption. This objective can be re-motivated as a service-oriented objective that measures costs proportional to the average time packages spend in a system. Depending on if a delivery model prioritizes the nodes being visited or the volume of demand being fulfilled at each node, the kTDP or EMVRP may be a better suited model. In comparison to our CTC-CVRPs, the EMVRP does not consider visit frequency decisions for demand locations.

Now, we briefly discuss classical column generation techniques for solving the LP re-

laxation of a set partitioning formulation for the CVRP and the relation to our problem. For a seminal reference on column generation techniques, we refer a reader to [86]. A classical set partitioning formulation for any multiple vehicle routing problem may take the following form:

$$\min \sum_{r \in \mathcal{R}} c_r z_r \quad (4.1a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} \alpha_{i,r} z_r = 1 \quad \forall i \in N, \quad (4.1b)$$

$$z_r \geq 0 \quad \forall r \in \mathcal{R}, \quad (4.1c)$$

$$z_r \in \mathbb{Z} \quad \forall r \in \mathcal{R}, \quad (4.1d)$$

where z_r denotes a binary decision on whether or not to include a route in the final dispatching policy. A route generally corresponds to a unique tour which starts and ends at a depot node, serving a subset of the demand node set, N . The variable, \mathcal{R} , denotes the set of all feasible routes (the components of this set may vary depending on the particular definition of a “route”). In the objective, (4.1a), the variable c_r denotes the cost of route. In constraint (4.1b), which describes the set partitioning of the demand nodes, the variable $\alpha_{i,r}$ denotes an indicator variable which equals 1 if node- i is serviced by route r and 0 otherwise. The set partitioning model is a powerful one for routing since all constraints on the feasibility of an *individual* route are captured in the variable definition phase; only feasible routes are included in \mathcal{R} .

The challenge, of course, is that identifying feasible routes requires enumerating (and possibly sequencing) all subsets of customer locations, which has exponential complexity and may lead to an IP with an exponential number of variables. Column generation is used to address this issue for the LP relaxation of (4.1). The dual problem for this LP is given

by:

$$\max \sum_{i \in N} \rho_i \quad (4.2a)$$

$$\text{s.t.} \quad \sum_{i \in N} \alpha_{i,r} \rho_i \leq c_r \quad \forall r \in \mathcal{R}, \quad (4.2b)$$

where ρ_i are decision variables which correspond to dual variables to the set partitioning constraints indicated by (4.1b). A notable method based on the work of [74] exists for solving the LP relaxation of (4.1) and its associated dual (4.2). In short, one can solve the LP relaxation of (4.1) over a subset of routes, say $R \subseteq \mathcal{R}$, and observe the corresponding dual solution $\{\rho_i^{R^*}\}_{i=1}^n$. Then one must then solve the separation problem:

$$\max_{r \in \mathcal{R}} \sum_{i \in N} \alpha_{i,r} \rho_i^{R^*} - c_r$$

typically modeled using a DP formulation. Note, to do this, one may choose to redefine (and relax) the notion of a “route”, which may include allowing repeat visits to a node. After solving this DP for an optimal route r^* with value $z_{r^*} = \sum_{i \in N} \alpha_{i,r^*} \rho_i^{R^*} - c_{r^*}$, if $z_{r^*} > 0$ then one must add the route r^* to R and resolve the LP relaxation of (4.1) and repeat this process. If it is the case that $z_{r^*} \leq 0$, then one is guaranteed to have solved the LP relaxation of (4.1) over all $r \in \mathcal{R}$ and not just $r \in R$.

The authors in [75] formulated the EMVRP using a set partitioning framework and attempted to find a process for column generation when solving the LP relaxation. They note specifically that the method highlighted above cannot be as easily applied to generating columns for the restricted set partitioning formulation of the EMVRP. This is because in the separation problem the cost of a route depends on knowing exactly how much demand will ultimately traverse each arc, which is not possible to do in a DP where the route is being built node by node. That being said, the authors in [75] were able to formulate a DP with an expanded state space using a multigraph to solve the separation problem.

Finally, we reference a few routing models in the literature which consider dispatching frequency decisions. A first family of models to note are Inventory Routing Problems (IRPs) which study the relationship between inventory management, vehicle routing, and delivery scheduling. There are many modeling variants for IRPs and for a recent survey please refer to [87]. Common formulations optimize over routing and inventory holding costs subject to service constraints on each demand node’s inventory level. Service providers often manage the routing and scheduling as well as the inventory levels of their customers in order to realize greater overall transportation efficiencies [88]. Recently, some authors have suggested that IRPs must evolve to meet the needs of the booming e-commerce era [89]; one example being that objective functions and customer routing frequencies should be studied in tandem, rather than just routing for feasibility. We believe our CTC-CVRPs help address this stated need. A second routing model which studies dispatch frequencies is the Period Vehicle Routing Problem (PVRP). As described in a recent survey paper [90], in these models vehicle routes are constructed over a planning period of many days/weeks/time-units. The goal being to find a set of dispatching schedule across this planning period which visits each demand node with a desired frequency at minimum cost. One variation most pertinent to our CTC-CVRPs is the Period Vehicle Routing Problem with Service Choice (PVRP-SC). In this model the objective function reflects an added benefit of serving customers more frequently than their minimum service level [91]. Although our planning horizon differs and we also consider the total accumulated routing times of the demand units, our models share this idea that routing frequency should be considered in the objective function.

4.2 Problem Description and Formulation

In this section we formally define the modeling inputs, constraints, and objectives required to formulate CTC-CVRPs. Consider a static VRP defined over a complete directed graph, $G = (N \cup \{0\}, A)$, where 0 denotes the singleton depot node and $N = \{1, 2, \dots, n\}$ defines

a set of n demand nodes. Additionally, $A = \{(i, j) : i, j \in N \cup \{0\}\}$ defines the set of arcs in the graph.

First we must define a cost metric over the arc set. Let $c_{(i,j)}$ denote the travel time associated with traversing arc $(i, j) \in A$, measured in hours. It is assumed that these travel times satisfy triangle inequality and non-negativity constraints. That is:

$$c_{(i,j)} \leq c_{(i,k)} + c_{(k,j)} \quad \forall i, j, k \in N \cup \{0\}, \quad c_a \in \mathbb{R}_{\geq 0} \quad \forall a \in A.$$

A route, r , is then defined as an ordered subset of N , which signifies the order of demand nodes to visit on a tour leaving from (and ending at) the depot node. For example, the route $\{r_1, r_2, r_3\}$ corresponds to the delivery tour “ $0 \rightarrow r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow 0$ ”, where r_i is defined as the i -th demand node visited by the route. For convenience, we define $|r|$ as the total number of demand nodes visited on the delivery route, and let $r_0 = r_{|r|+1} = 0$. Thus, the total time associated with traversing a route can be calculated as:

$$\ell_r = s_0 + |r|s_1 + \sum_{i=0}^{|r|} c_{(r_i, r_{i+1})}$$

where $s_0 \in \mathbb{R}_{\geq 0}$ defines a fixed setup time required at the depot, and $s_1 \in \mathbb{R}_{\geq 0}$ defines the constant service times of visiting each demand node.

Each of the demand nodes in the network generates demand for delivery at a constant rate of $q_i \in \mathbb{R}_{\geq 0}$, measured in volume per time; in the application that motivates our work, this metric is packages per hour. Furthermore, the appropriate packages for delivery are generated at the depot location at the same rates; note then that the total packages then arriving to the depot is $\sum_{i \in N} q_i$. Thus, while this model treats geographic locations as discrete, it considers demand as a continuous fluid over time albeit with a constant rate. This time demand model is a continuous approximation, and as an aside, a reference on continuous approximation models in logistics is provided by the recent survey [23].

Demand is served via a delivery fleet of size $m \in \mathbb{Z}_{\geq 1}$. Each vehicle has a maximum carrying capacity of $Q \in \mathbb{R}_{\geq 0}$ packages at any one time. A defining feature of our model is the notion that planned routes will be operated continuously over time, loading arriving packages at the depot and serving customers. Multiple delivery vehicles can be assigned to the same route, in order to create enough capacity to serve demand. For a given route, r , a single vehicle generates a package movement capacity of Q/ℓ_r packages per hour. Additionally, the demand nodes on the route generate a total demand of $q_r = \sum_{i=1}^{|r|} q_{r_i}$ packages per hour. Letting $m_r \in \mathbb{Z}_{\geq 1}$ variably represent the number of vehicles assigned to route r , for the sake of feasibility it must be the case that $m_r \geq m_r^{\min} = \lceil \frac{q_r \ell_r}{Q} \rceil$.

In order to specify a cycle-time-related objective function, let us construct one piecewise from the perspective from an individual package at a demand node. Consider some demand node, $n' \in N$, which is served uniquely by some route, r . Thus, for some $i \in \{1, 2, \dots, |r|\}$, we can say that $r_i = n'$. Once the package leaves the depot it spends $\sum_{j=0}^{i-1} (c_{(r_j, r_{j+1})} + s_1)$ hours on the route before reaching its final destination. Irrespective of the number of vehicles assigned to this route, m_r , all of the $q_{n'}$ packages per hour incur this traveling time from the depot to node n' . What will change with respect to m_r , however, is the time each package spends waiting at the depot before leaving on the route. The expected time spent waiting at the depot, per package, is given by $\frac{\ell_r}{2m_r}$ hours.

We can now price a route by the total demand weighted time of its associated waiting and traveling times. The cost of a given route, c_r , which uses $m_r \geq m_r^{\min}$ vehicles is given by:

$$c_r = \sum_{i=1}^{|r|} q_{r_i} \left(\frac{\ell_r}{2m_r} + \sum_{j=0}^{i-1} (c_{(r_j, r_{j+1})} + s_1) \right).$$

We are now ready to formulate our CTC-CVRP models. One of the most natural approaches is to define a set partitioning formulation over some predefined set of routes, \mathcal{R} . For now, suppose that this set is known and given. Let us first specify a nonlinear IP formulation. Let z_r be a binary decision variable for whether or not route r is chosen in the final dispatching policy, and let m_r denote the integral number of vehicles assigned to service

the route. Consider then the following optimization problem:

$$\min \sum_{r \in \mathcal{R}} z_r \left(\sum_{i=1}^{|r|} q_{r_i} \left(\frac{\ell_r}{2m_r} + \sum_{j=0}^{i-1} (c_{(r_j, r_{j+1})} + s_1) \right) \right) \quad (4.3a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} \alpha_{i,r} z_r = 1 \quad \forall i \in N, \quad (4.3b)$$

$$\sum_{r \in \mathcal{R}} m_r z_r = m, \quad (4.3c)$$

$$m_r \geq z_r m_r^{\min} \quad \forall r \in \mathcal{R}, \quad (4.3d)$$

$$m_r \in \mathbb{Z} \quad \forall r \in \mathcal{R}, \quad (4.3e)$$

$$z_r \geq 0 \quad \forall r \in \mathcal{R}, \quad (4.3f)$$

$$z_r \in \mathbb{Z} \quad \forall r \in \mathcal{R}. \quad (4.3g)$$

The constraint (4.3b) describes the set partitioning restriction to visit each demand node with exactly one route, using the variables $\alpha_{i,r}$, where $\alpha_{i,r} = 1$ if $i \in r$ and zero otherwise. The constraint (4.3c) enforces the vehicle fleet restriction, while (4.3d) enforces the constraint that each route must generate enough capacity to serve demand. The difficulty in working with this formulation as is, is that the objective function, (4.3a), includes the nonlinear term z_r/m_r , and constraint (4.3c) includes the nonlinear term $m_r z_r$.

A simple way to linearize this model is to redefine the definition of a route and the feasible set of available routes. Suppose now that a route as a tuple, (r, m_r) , where the tour “ $0 \rightarrow r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_{|r|} \rightarrow 0$ ” is served by exactly m_r vehicles. Furthermore, this route is only feasible for $m_r \geq m_r^{\min}$. Letting \mathcal{R}^+ denote some predefined set of feasible routes of this type, according to this new definition of a route, we can now consider the linearized optimization problem:

$$\min \sum_{r \in \mathcal{R}^+} c_r z_r \quad (4.4a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}^+} \alpha_{i,r} z_r = 1 \quad \forall i \in N, \quad (4.4b)$$

$$\sum_{r \in \mathcal{R}^+} m_r z_r = m, \quad (4.4c)$$

$$z_r \geq 0 \quad \forall r \in \mathcal{R}^+, \quad (4.4d)$$

$$z_r \in \mathbb{Z} \quad \forall r \in \mathcal{R}^+. \quad (4.4e)$$

Since c_r and m_r are given for any route, this formulation with a cycle time minimizing objective has been successfully linearized. We know for any individual tour that by adding more vehicles we can reduce the overall cycle time by reducing the average package waiting time at the depot. We also know that by increasing the overall fleet size we stand to reduce the overall average cycle per package in the system. Therefore, it is of interest to understand the sensitivities of our model with respect to the variable m . In order to calculate the minimal fleet size for which a given problem instance is feasible, we can formulate a second CTC-CVRP formulation, this time with a fleet size minimizing objective as follows:

$$\min \sum_{r \in \mathcal{R}^+} m_r z_r \quad (4.5a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}^+} \alpha_{i,r} z_r = 1 \quad \forall i \in N, \quad (4.5b)$$

$$z_r \geq 0 \quad \forall r \in \mathcal{R}^+, \quad (4.5c)$$

$$z_r \in \mathbb{Z} \quad \forall r \in \mathcal{R}^+. \quad (4.5d)$$

Going forward, we refer back to the fleet size minimization formulation given in (4.5) as our “Phase 1” problem, and refer to the cycle time minimization formulation given in

(4.4) as our “Phase 2” problem.

4.3 Model Properties and Observations

In this section, we observe various analytical properties of our models and their impact on questions of tactical system design. To start the conversation, let us consider the impact of fleet size relative to our modeling outcomes. Given a set of input parameters, solving the Phase 1 problem allows us to solve for the minimal fleet size necessary to be able to repeatably serve all of the demand nodes in the system; feasibility here means providing enough vehicle capacity to meet all customer demand. As an aside, note that this minimal fleet size of course depends on the given set \mathcal{R}^+ . Unlike standard vehicle routing problems, however, this model has an interesting feature: any ordered subset of customers generating a route r can feasibly serve the demand of those customers by simply increasing the number of vehicles m_r assigned to the cycle. Thus, we might expect these models to have many more feasible routes.

Furthermore, in order to determine the set of routes which are cycle time minimizing, this minimal fleet size must be used in place of the variable m in constraint (4.4c) within the Phase 2 optimization problem. What would be interesting from a design viewpoint, is how fleet sizing beyond this minimal value impacts the cycle time objective in (4.4a). As an extreme case, we first observe a lower bound on the cycle time objective per Theorem 23.

Theorem 23. *For any given predefined set of feasible routes \mathcal{R}^+ and any given fleet size m the optimal objective value returned by solving the Phase 2 problem, $z(\mathcal{R}^+, m)$, is lower bounded by the total package weighted time it would take to serve each demand node via a direct route without considering any waiting time. That is, $z(\mathcal{R}^+, m) \geq \sum_{i=1}^n q_i (c_{(0,i)} + s_1)$ for all \mathcal{R}^+, m .*

Proof. See appendix, C.1. □

Intuitively, as the available fleet size grows, the average time packages spend waiting in the system becomes insignificant with respect to the time packages spend en route to their destination. Informally, Theorem 23 illustrates that at some point, the fleet size becomes large enough that it is desirable to serve each demand node via a direct route. What is interesting about this Theorem is that this type of asymptotic behavior is not exhibited in standard CVRP-like models. Before elaborating on this point, recall from our discussion in Section 4.1.1 that the CTC-CVRP fleet serves two purposes. Adding more vehicles on a route not only reduces the waiting time packages spend at the depot, but it also adds more generated capacity on a route. In a CVRP-like model, increasing the fleet size beyond some calculable point may not decrease total objective time, however increasing the capacity of each vehicle will eventually cause the optimal set of delivery tours to converge to a single tour solution. Thus we see that the two models exhibit differing asymptotic behaviors.

From a practical design standpoint, it is of interest to understand the relationship between our cycle time objective and the fleet size constraint (4.4c) in our Phase 2 problem. Specifically, are we able to succinctly model the behavior of the objective function as the fleet size marginally increases? Naturally, the cycle time in the system is non-increasing as fleet size increases, but it would be of greater interest if we could definitively state that cycle time reductions are marginally decreasing as fleet size increases. For a general set of modeling inputs, this is not true as will be shown in section 4.5. Nevertheless, it may be the case that certain families of modeling instances exhibit solution structures that allow for more predictable cycle time outcomes in optimal solutions given changes in overall fleet size. What is certain, however, is that the cycle time cost of any individual route does exhibit this marginally decreasing behavior as more vehicles are added onto the route. This observation is formalized in Theorem 24.

Theorem 24. *Consider an ordered subset of N , namely r , and the minimal number of vehicles required to construct a feasible route tuple m_r^{min} . For any given $k \geq m_r^{min}$ define $r(k)$ as the routing tuple (r, k) which has a cost of $c_{r(k)}$. Then for any $k \geq m_r^{min}$ we have*

that: $c_{r(k)} - c_{r(k+1)} > c_{r(k+1)} - c_{r(k+2)}$.

Proof. See appendix, C.2. □

Although we will not explore these ideas further in this chapter, we should also make a note about unnecessary space capacity. It is clear that the Phase 1 optimization model determines the minimum fleet size necessary to serve all node demand over time given the available route set \mathcal{R}^+ . Given this route set, then, these solution exhibits high vehicle utilization. One way to measure utilization is some time-weighted measure of vehicle load, but we cannot claim that any optimal solution to the Phase 1 optimization maximizes this utilization metric. For example, it is easy to see that with symmetric travel times two routes serving the same customers but in reversed order will have different utilizations. More relevant in our model is the average load carried when dispatched from the depot. To use the fewest vehicles, vehicles should leave the depot frequently nearly full. The dual idea is that we should dispatch empty vehicle space frequently from the depot. Once more vehicles are provided to a system (to reduce average demand cycle time), the empty vehicle space dispatched per time will increase thus measuring a decrease in vehicle utilization. If this empty space is significant, then reducing vehicle size Q might be an appropriate strategy.

Thus far, the set partitioning models described by our Phase 1 and 2 optimizations rely on the use of some predefined set of routes \mathcal{R}^+ . A classical approach to defining this set may be to enumerate all possible routes satisfying certain conditions. We see from our definition of route cost c_r in Section 4.2 we as well as from our feasibility conditions on m_r , our problem will naturally avoid lengthy routes, especially ones with a large number of demand nodes visited. Define Ω_p^m to be the set of all feasible routing tuples which visit less than or equal to p demand nodes and use less than or equal to m vehicles to do so. Thus our CTC-CVRPs, it is natural to choose a $p \geq 1$ and let $\mathcal{R}^+ = \Omega_p^m$.

Note that if we choose $\mathcal{R}^+ = \Omega_p^m$, as p increases the decision space of routing options increases along with model complexity. It is of interest to determine if we are able to preprocess and remove any of these routes from the model before calling upon an optimization

solver. For example, due to the complexity of the cost function and feasibility criteria of our route variables, it is not inherently obvious if there is a single best permutation of demand nodes to visit given a subset of N , as there may be in a more traditional CVRP-like model. The following definition formally introduces the concept of route domination, which is subsequently explored in Property 25 and the subsequent example.

Dominated routes: Consider two ordered subsets of N which contain the same demand nodes to be visited but in two distinct orderings. Call these subsets $N_a \subseteq N$ and $N_b \subseteq N$. Additionally consider two vehicle fleet sizes, namely m_a and m_b , which are used to define the feasible routes $r_a = (N_a, m_a)$ and $r_b = (N_b, m_b)$. If it is the case that $c_{r_a} < c_{r_b}$ while $m_a \leq m_b$, then route r_a is said to dominate route r_b .

Property 25. Given a predefined set of feasible routes \mathcal{R}^+ consider two possible routing options, $r_a \in \mathcal{R}^+$ and $r_b \in \mathcal{R}^+$. In the case that r_b is dominated by r_a , then we can remove r_b from \mathcal{R}^+ without loss of generality before solving either the Phase 1 or Phase 2 optimization problem.

Dominance relations are often found in operations research modeling to simplify analysis and solution algorithms. What is interesting about our CTC-CVRP modeling is that while the conditions to establish dominance are readily understood, complicated relationships between routing permutations may arise for even the simplest of topologies. Consider the situation where three demand nodes $\{A, B, C\}$ and a depot node 0 are described on a Cartesian plane and distances are euclidean, as illustrated in Figure 1. Under a set of modeling inputs, fully described in appendix C.3, it can be shown that permutation $A \rightarrow B \rightarrow C$ dominates all other permutations for 1 and 2 vehicle routes, while the permutation $B \rightarrow C \rightarrow A$ dominates all other permutations for all routes using 3 or more vehicles. Furthermore, the permutation $A \rightarrow C \rightarrow B$ is dominated for any fleet size.

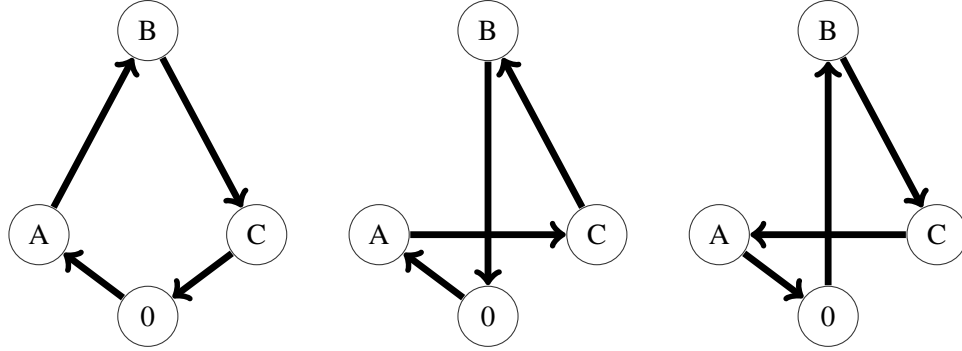


Figure 4.1: Illustration of three different routing permutations to serve the demand nodes A, B, and C from the depot node 0 on the same route. The remaining three permutations are each identical to one of the above permutations by symmetry and are removed for visual clarity.

4.4 Column Generation Techniques

As discussed in Section 4.2, nonlinearity difficulties in our cycle time objectives as well as in our fleet allocation constraints guided the modeling of our CTC-CVRPs towards set partitioning formulations. In the literature review we discussed how the related EMVRP model was also shown to have better modeling capabilities in a set partitioning framework [73, 75, 84]. Specifically in [75] an arc-variable MIP formulation and a route-variable set partitioning formulation are compared and it was found that the LP relaxation of the set partitioning formulation was at least as strong as that of the arc-variable MIP formulation. Due to its strengthened relaxation and also its simpler application within a branch-cut-and-price framework, the set partitioning model of the EMVRP was highlighted as the preferred modeling choice [75]. Although in this chapter it is out of scope to describe a full branch-cut-and-price algorithm, we review some findings relevant to solving the LP relaxation of our formulations and subsequent use within IP improvement heuristics. In Section 4.5 we will describe various computational exercises which illustrate such techniques.

Recall from our earlier discussions concerning \mathcal{R}^+ that for our models with cycle time considerations, shorter tours are qualitatively more desirable than lengthier tours. Thus, we defined a set of routing tuples which limited the number of delivery nodes visited and

proposed using this set as our \mathcal{R}^+ . Another advantage of working through a column generation methodology here, besides for future use in branch-cut-and-price algorithms, is that we may be able to dynamically generate more desirable lengthier tours without having to enumerate all of them.

For the remainder of the section, we choose to focus on Phase 2 problem as the complex cost structure in the objective makes for a more interesting analysis. Now, consider the LP relaxation of (4.4) and its associated dual problem:

$$\min \sum_{r \in \mathcal{R}^+} c_r z_r \quad (4.6a)$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}^+} \alpha_{i,r} z_r = 1 \quad \forall i \in N, \quad (4.6b)$$

$$\sum_{r \in \mathcal{R}^+} m_r z_r = m, \quad (4.6c)$$

$$z_r \geq 0 \quad \forall r \in \mathcal{R}^+, \quad (4.6d)$$

$$z_r \in \mathbb{R} \quad \forall r \in \mathcal{R}^+. \quad (4.6e)$$

$$\max \sum_{i \in N} \rho_i + m\mu \quad (4.7a)$$

$$\text{s.t.} \quad \sum_{i \in N} \alpha_{i,r} \rho_i + m_r \mu \leq c_r \quad \forall r \in \mathcal{R}^+, \quad (4.7b)$$

$$\rho_i \in \mathbb{R} \quad \forall i \in N, \quad (4.7c)$$

$$\mu \in \mathbb{R}. \quad (4.7d)$$

When initially choosing the set of feasible routes for this problem, we have discussed the merits of setting \mathcal{R}^+ to equal Ω_p^m for some $p \geq 1$. Note that when $p = n$, Ω_p^m will equal the set of all possible routing tuples for the given problem instance. Generally speaking,

however, any $\mathcal{R}^+ \subseteq \Omega_n^m$ can be chosen to optimize over. Depending on the problem size, enumerating all routes of a predetermined length may be costly in terms of solver time or computational memory. It may be better to start with a smaller subset of routes, then systematically grow the routing set via a column generation procedure. For example if we are interested in optimizing over $\mathcal{R}^+ = \Omega_5^3$, we could start by optimizing over $\mathcal{R}^+ = \Omega_3^3$ then solve a sequence of pricing problems to efficiently determine if any routes of length four or five should be added to the decision space. Such a procedure can achieve the same performance as if one had started with the larger set of routes Ω_5^3 to begin with.

Consider the problem of solving problem (4.6) over the set \mathcal{R}^+ where it may be intractable or undesirable to completely enumerate all of \mathcal{R}^+ . We can instead initially choose a smaller decision space $R^+ \subseteq \mathcal{R}^+$ to optimize over. After solving (4.6) over R^+ , consider the associated optimal dual variables $\{\rho_i^{R^+*}\}_{i \in N}$ and μ^{R^+*} . From constraint (4.7b), if it were possible to show that $\sum_{i \in N} \alpha_{i,r} \rho_i^{R^+*} + m_r \mu^{R^+*} \leq c_r$ for all $r \in \mathcal{R}^+$ then it would be correct to say that $\{\rho_i^{R^+*}\}_{i \in N}$ and μ^{R^+*} optimally solve (4.7) over all $r \in \mathcal{R}^+$ and not just for $r \in R^+$. With that in mind, we now formally introduce the CTC-CVRP pricing problem for column generation.

CTC-CVRP Pricing Problem Consider the optimal dual variables $\{\rho_i^{R^+*}\}_{i \in N}$ and μ^{R^+*} obtained via solving (4.7), or equivalently (4.6), over a given route set R^+ . The CTC-CVRP pricing problem is then defined as: $\max_{r \in \mathcal{R}^+} \sum_{i \in N} \alpha_{i,r} \rho_i^{R^+*} + m_r \mu^{R^+*} - c_r$.

The optimal objective of this pricing problem allows us to determine whether the current solution is optimal; if the maximum price (reduced cost) is not greater than zero, then the solution is optimal and if not we have identified a dual row (primal column) that should be added to R^+ to improve the primal objective. Column generation relies on being able to solve this underlying pricing problem efficiently. As described in the literature review in Section 4.1.1, a traditional CVRP modeled within a set partitioning formulation can solve its pricing problem via a DP which constructs an optimal reduced-cost route stop by stop in a prize-collecting fashion. The difficulty in directly applying that approach here is that

the route cost cannot be explicitly determined without knowing the entire route length and stop-by-stop demand to be served upfront. Equivalently stated, there is simply not enough information to properly construct the route cost when the state space consists solely of a current location and remaining vehicle capacity. The authors in [75] were able to define a a EMVRP column generation procedure by adding cumulative supply variables to be picked up before each arc is traversed. For our problem, not only do we need to keep track of cumulative packages, but also cumulative path lengths. This is necessary since the waiting time component of route cost depends on the overall tour length.

We will describe a DP methodology with a state space that is sufficient to solve our pricing problem. In addition to the more complicated state space, in our problem we also consider adding multiple vehicles to the same routing tour. Thus, for both implementing feasibility conditions as well procedurally building the cost of a route, it is advantageous to perform the pricing iteratively over a fixed fleet size variable. Now we will modify the pricing problem before formalizing the final DP.

Modified CTC-CVRP Pricing Problem Consider the optimal dual variables $\{\rho_i^{R^{+*}}\}_{i \in N}$ and $\mu^{R^{+*}}$ obtained via solving (4.7), or equivalently (4.6), over a given route set R^+ .

The modified CTC-CVRP pricing problem is then defined as:

$$\max_{m' \leq m} \max_{r \in \mathcal{R}^+ | m_r = m'} \sum_{i \in N} \alpha_{i,r} \rho_i^{R^{+*}} + m_r \mu^{R^{+*}} - c_r.$$

In order to be able to define an efficient DP which solves the modified CTC-CVRP pricing problem, we will relax the definition of what constitutes a routing tour. Being able to solve the DP more efficiently with a simpler state space, comes at the cost of having a weaker LP bound. To do this, we allow a route to repeat visits to the same demand node, just not consecutively. Thus the variable $\alpha_{i,r}$ now indicates the number of times that the route visits node $i \in N$. Keeping in mind our tendency to limit the number of stops on each considered delivery tour, we also introduce a state space variable which keeps track of the number of demand nodes visited. Now to model and subsequently solve the modified CTC-CVRP pricing problem, consider the following DP defined for a given value of m' :

State Space: Intermediate states are of the form: $(i, \sum q, \sum \ell, p)$, where the current location on the path is given by $i \in N$, the total volume of packages delivered up to this point is given by $\sum q \geq 0$, the current path length is given by $\sum \ell \geq 0$, and the total number of demand nodes visited is given by $p \geq 0$.

Initial State: The initial state is given by $S^{(0)} := (i = 0, \sum q = 0, \sum \ell = 0, p = 0)$.

Terminal State: There is a single terminal state, simply named $S^{(T)}$.

Actions and Rewards to Non-Terminal States: Any non-terminal state $(i, \sum q, \sum \ell, p)$ can transition to another non-terminal state $(j, \sum q + q_j, \sum \ell + c_{(i,j)} + s_1, p + 1)$ only if: $(\sum q + q_j)(\sum \ell + c_{(i,j)} + c_{(j,0)} + s_1 + s_0) \leq Qm'$ and $i \neq j$. This transition obtains a reward of: $\rho_j^{R^{+*}} - q_j(\sum \ell + c_{(i,j)} + s_1) - \frac{1}{2m'}(\sum q + q_j)(\sum \ell + c_{(i,j)} + s_1 + c_{(j,0)} + s_0) + \frac{1}{2m'}(\sum q)(\sum \ell + c_{(i,0)} + s_0)$.

Actions and Rewards to the Terminal State: Any non-terminal state $(i, \sum q, \sum \ell, p)$ can transition to the terminal state of $S^{(T)}$. This transition obtains no reward.

The objective of the DP is to move from the initial state to the terminal state maximizing the total accumulated reward. The network of states and possible transitions is guaranteed to be acyclic since transitions only exist between a state and one with reduced capacity, or to a terminal state. Given the structure of this particular DP, we note that a forward-reaching methodology may be the preferred solution approach. Thus, we define $z^*(S^{(0)}) = m'\mu^{R^{+*}}$ to equal the DP valuation of the initial state with the goal of using the above DP definition to calculate the value of the terminal state $z^*(S^{(T)})$. Once we have obtained this value for all $m' \leq m$ we are able to solve the modified CTC-CVRP pricing problem.

Suppose that the pricing problem were solved and returned the route $r^* \in \mathcal{R}^+$ with an objective value strictly greater than 0; then we can add r^* to the set R^+ . We can then repeat the process and determine if any more routes should be added to the decision space. In contrast, if the pricing problem returned an objective value less than or equal to 0, then we

can terminate the column generation procedure. Note that column generation and solving the associated pricing problem can be sometimes more of an art rather than a science. For example, in Section 4.5 we choose to continually cycle through the possible values of m' , each time adding the most cost-reduced route to the set R^+ , if appropriate. This is in contrast to finding the most cost-reduced route over all values of m' before adding anything to R^+ . Then, once no more desirable routes can be generated for any value of m' , the procedure terminates. This procedure is further detailed in Section 4.5.3. On the other hand, one could consider adding multiple routes to the set R^+ at each column generation step, not just the most cost-reduced.

As an final, yet highly important, note we discuss the concept of dynamic state generation and state domination when solving the DP model for the pricing problem. Similar to the idea of route domination discussed in Section 4.3, not all feasible DP states need be considered when attempting to find the most cost-reduced route. Formally a state $S^{(a)} := (i^{(a)}, \sum q^{(a)}, \sum \ell^{(a)}, p^{(a)})$ dominates the state $S^{(b)} := (i^{(b)}, \sum q^{(b)}, \sum \ell^{(b)}, p^{(b)})$ if the following criteria are met: $i^{(a)} = i^{(b)}$, $\sum q^{(a)} \leq \sum q^{(b)}$, $\sum \ell^{(a)} \leq \sum \ell^{(b)}$, $p^{(a)} \leq p^{(b)}$, and $z^*(S^{(a)}) > z^*(S^{(b)})$. Qualitatively, if two states in the DP are at the same demand node, but one of states has achieved strictly greater state valuation while consuming less than or equal to the capacity of the other state, then the dominated state can be removed from the DP without loss of generality. Such state space reductions can improve not only computational memory but also solution speed.

4.5 Computational Examples

In this section, our goal is to walk our models through a variety of computational experiments. Due to the novelty of our problem, our intent here is to simplify the network topologies we simulate and instead focus illustrating the modeling, solving, and utility of our CTC-CVRPs. After describing a set of simulation parameters, we will solve the Phase 1 optimization problem to determine the minimally-necessary fleet size for a generated

sample instance. We follow this with an extended analysis of the Phase 2 problem, specifically focusing on design implications varying the fleet size parameter. Finally, we study the impacts of a column generation implementation heuristic which dynamically creates desirable routes not in the initial decision space for varying generated instances. All experiments leverage Gurobi 9.1.1 as an optimization solver and all software for this work is coded in Python 3.6.3.

4.5.1 Phase 1 Optimization

We start our experimentation by considering an artificial 20 mile by 20 mile service region with a depot node at its center. Within this region are $n = 10$ demand nodes, each generated at a random lattice point with uniform probability. The cost metric between any two nodes on the graph is measured via their Manhattan distance normalized by a vehicle speed of 20 miles per hour, that is, $c_{(i,j)} = \frac{1}{20} (|x_i - x_j| + |y_i - y_j|)$. The fixed setup time at the depot and the service time at each demand node are ignored in this analysis and set equal to zero.. Each demand node generates demand via an independent and identically distributed discrete uniform distribution over the integer values from 1 to 5 packages per hour, inclusively. As a final note, each service vehicle has a maximum carrying capacity of $Q = 20$ packages. From here we generated a random instance of our demand locations and demand rates, which are described fully in the appendix, C.4.

Much like a realistic service provider designing delivery routes to serve these demand nodes, at this point it is unknown how many vehicles to assign to this region. We use our Phase 1 optimization problem first to determine a minimally necessary fleet size. The final modeling component yet to be defined, however, is how to approach the feasible set of routing options. To start our analysis, we choose to let $\mathcal{R}^+ = \Omega_4^3$, as first discussed in Section 4.3. Recall that Ω_p^m is the set of all feasible routing tuples which visit less than or equal to p demand nodes and use less than or equal to m vehicles to do so.

There are 5,860 different permutations of demand nodes which convert to tours which

visit four or fewer demand nodes for our given problem instance. From these permutations, 12,532 feasible route tuples were created which use three or less vehicles per tour. By solving the Phase 1 optimization problem, it was determined that at least three vehicles are necessary to service this region. In order to arrive at this minimal fleet size, the model returned the following routes: $(5 \rightarrow 2 \rightarrow 6)$, $(7 \rightarrow 3 \rightarrow 10)$, and $(4 \rightarrow 9 \rightarrow 8 \rightarrow 1)$ each using a single vehicle. Although these routes were not explicitly selected to minimize total cycle time costs, that value was determined to be 96.1 minutes per package (on average). Figure 4.2 illustrates the graph of the simulated region and the solution routes from the Phase 1 optimization.

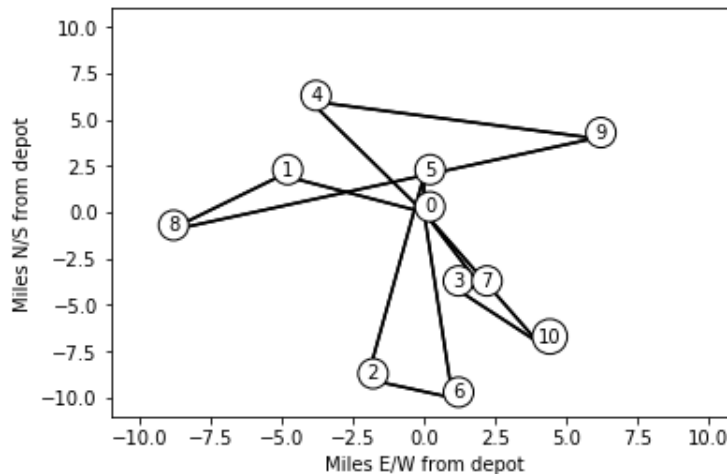


Figure 4.2: Our simulated instance requires at least three vehicles in order to serve demand, as determined by solving the Phase 1 problem. The depicted solution uses three delivery routes, each using one vehicle.

4.5.2 Phase 2 Optimization

Now that we know the minimal fleet size necessary to serve the demand in this region, subject to the selected parameters, it is of interest to determine which of these 12,532 route tuples to leverage in order to minimize package cycle time. Having initialized our Phase 2 problem with three vehicles, our Gurobi solver returns the following routes: $(3 \rightarrow 7 \rightarrow 10)$, $(6 \rightarrow 2 \rightarrow 8)$, and $(5 \rightarrow 1 \rightarrow 4 \rightarrow 9)$ each using a single vehicle. The average cycle time

per package was determined to be 76.2 minutes. These results are illustrated in Figure 4.3.

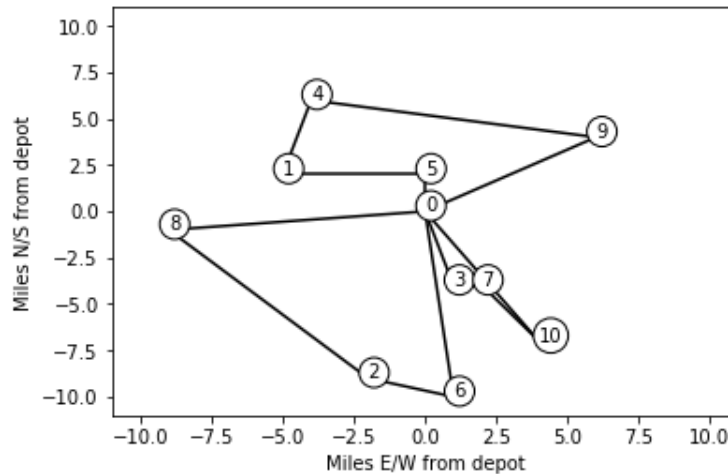


Figure 4.3: Give the minimal fleet size to serve demand, our Phase 2 optimization problem is able to determine which delivery routes to leverage in order to minimize package cycle time.

As previously discussed, it is of interest to quantify the achievable marginal reductions in package cycle time as additional vehicles are added to the service fleet. As we chose $\mathcal{R}^+ = \Omega_4^3$ we have that each feasible route can use no more than three vehicles. Thus, we know to serve these $n = 10$ demand nodes our models cannot use more than thirty vehicles in total. As our Phase 1 model chose to utilize three vehicles, at minimum, this upper bound on fleet size is seemingly more than sufficient and supports the initial choice of \mathcal{R}^+ . Therefore, we repeatedly solved the Phase 2 optimization problem for this problem instance for $m = 3$ to $m = 30$ and have plotted the resulting optimal objective value in Figure 4.4. Notably, we observe a 12.6 minute reduction in average cycle time (16.6% reduction) when implementing a four vehicle solution instead of the minimal three vehicle solution. An additional 8.8 minutes of average cycle time per package (13.9% reduction) can be reduced by adding a fifth vehicle on top of the four vehicle solution. All other marginal reductions in cycle time are less than 5 minutes per package which, depending on a service provider’s use-case, may or may not be desirable to pursue.

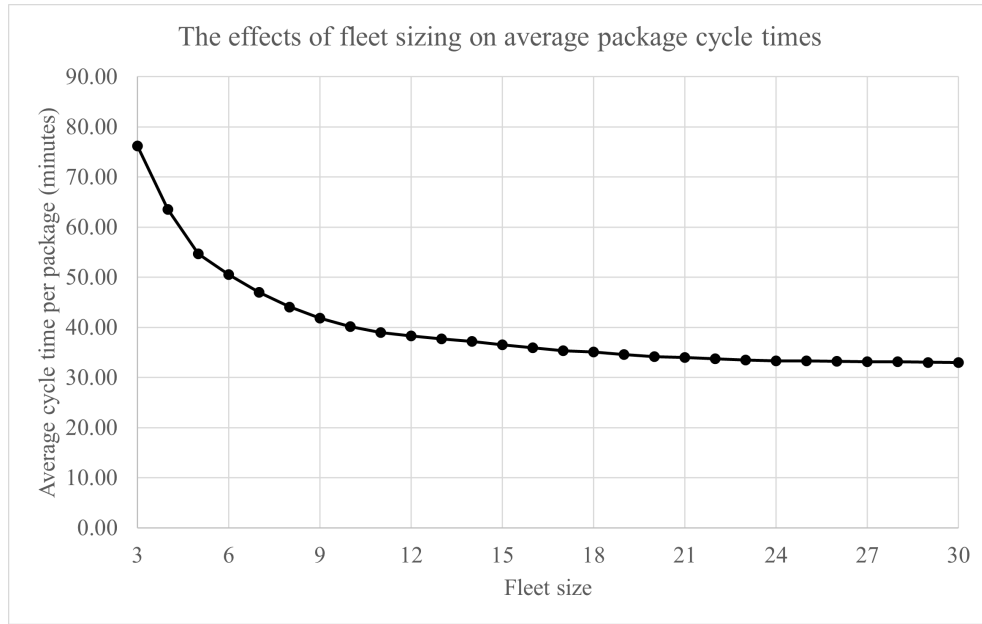


Figure 4.4: Depiction of how the Phase 2 objective value changes as the fleet size parameter increases from $m = 3$ to $m = 30$ for our simulated problem instance.

Another interesting observation illustrated in Figure 4.4 is that while the cycle time objective is strictly decreasing for all values from $m = 3$ to $m = 30$, we empirically observe that this plot is non-convex between integer points. Specifically, there are multiple instances where the adding an additional vehicle yields less than half the cycle time reduction than if two vehicles were to be added to the fleet size. This result implies, at least for some topologies and modeling parameters, that a service provider may have to explicitly evaluate a range of fleet sizes in order to determine what they consider to be an optimal fleet size compared to marginal cycle time benefits. This is in contrast to a simpler valuation that could greedily determine vehicle by vehicle if incrementing the fleet size would be advantageous. This result shows that while Theorem 24 proves incremental cycle time benefits for any individual route are marginally decreasing with respect to increasing fleet sizes, this result does not generally extend to the overall optimization problem.

Theorem 23 asserts that for any vehicle fleet size, the Phase 2 optimization problem can do no better than direct routes with no assumed waiting time. We plotted the optimal

Phase 2 delivery routes for the fleet sizes of 5, 10, 20, and 30 vehicles in Figure 4.5. A last notable observation from this computational experiment is that as fleet size progressively increases, the number of direct routes from the depot to an individual demand node and back increases. Specifically, for the $m = 5$ case no direct routes were realized in the optimal delivery solution, while in the $m = 10$ case we observe one direct route ($0 \rightarrow 8 \rightarrow 0$). The $m = 20$ and $m = 30$ cases observe four and ten direct routes, respectively.

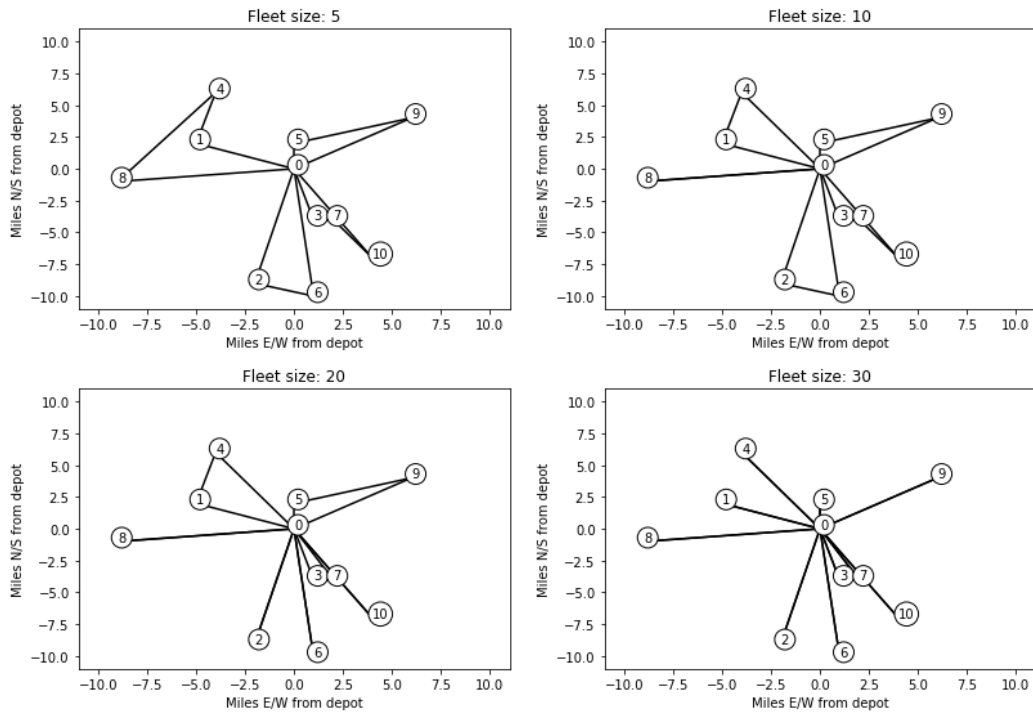


Figure 4.5: As fleet size increased, for the Phase 2 optimization problem, more and more direct routes are found in the optimal delivery solution.

4.5.3 Column Generation Applications

In Section 4.4 we discussed a column generation methodology for solving an LP relaxation of our Phase 2 optimization problem. There are a number of purposes for such a procedure. LP relaxations in general are important to both bounding the objective value of the corresponding IP, as well as in solving the IP itself. Column generation can be used within a full branch-cut-and-price framework, but for this chapter and this experiment specifically we solely focus on the use within an IP improvement heuristic. For some motivation, note

that although Ω_p^m may be an initially good choice for \mathcal{R}^+ given the structure of our CTC-CVRPs, it may be advantageous to consider routes with more than p stops.

For this experiment we create 18 new simulated problem instances with varying input parameters. For each instance we may vary the number of demand nodes in the region n or the size of the vehicle fleet m . Specifically we consider $(n, m) \in \{(12, 4), (12, 6), (24, 8), (24, 12), (36, 12), (36, 18)\}$. Repeating each of these combinations three times, we arrive at our 18 simulated instances, or scenarios. Note that all scenarios are independently generated. Each demand node still generates demand via an independent and identically distributed integer value from 1 to 5 packages per hour and each service vehicle has a maximum carrying capacity of $Q = 20$ packages. Vehicle speeds, arc cost metrics, as well as fixed setup times and service times all remain the same as the earlier instance. However, we sample demand node locations from a larger 30 mile by 30 mile region.

After randomly generating the node location and hourly package demand at each of the n demand nodes for each scenario, we chose to let $\mathcal{R}^+ = \Omega_3^3$ for an initial decision space. We subsequently solved the Phase 2 optimization problem to determine the resulting average cycle time per package in each optimal delivery policy. Since our values for n ranged from 12 to 36 stops and our values for m ranged from 4 to 18 vehicles, any optimal dispatching policy limited to routes of three stops or less seems quite restrictive. Thus, we are interested in knowing how much the average cycle time per package could decrease if we allowed routes with more stops, say up to six. That is, we wanted to determine what would happen to our Phase 2 objective values if we would have let $\mathcal{R}^+ = \Omega_6^3$.

Setting $R^+ = \Omega_3^3$ and $\mathcal{R}^+ = \Omega_6^3$ we use the results of Section 4.4 in order to accomplish this feat. Recall that now the definition of a route will allow for multiple, non-consecutive, visits to the same demand node. For each scenario we start by solving the Phase 2 LP relaxation for $R^+ = \Omega_3^3$. Then by restricting ourselves to a single value for $m' \in \{1, 2, 3\}$ we solved the modified CTC-CVRP pricing problem in order to determine if any routes of six or less stops should be added to the set R^+ . We cycled through this process three

times, for $m' = 1$, $m' = 2$, and finally for $m' = 3$ respectively. Once no more routes could be generated with negative reduced cost for any value of m' this process terminated.

One purpose of solving this LP relaxation is to lower bound the objective function value that could be achieved if the Phase 2 IP were solved for $\mathcal{R}^+ = \Omega_6^3$ to begin with. This value will be a lower bound since not only are the decision variables allowed to be continuous rather than binary, but also the definition of a route itself was relaxed. In Table 4.1 we can compare the results of the initial Phase 2 IP with a three stop limit and the corresponding LP relaxation with a six stop limit as determined via our column generation procedure. In Table 4.1 we record the objective values for each model expressed as average cycle time minutes per package, the number of routes generated to solve the 6-stop LP relaxation, and the associated objective gap between the two models.

From Table 4.1 we observe anywhere from a 1.7% to a 27.7% gap in the average cycle time per package between the 3-stop IP and the 6-stop LP relaxation. For a fixed number of demand nodes, this gap appears to be larger in the case of a smaller fleet size. This result suggests the importance of allowing lengthier routes into the decision space when the fleet is more restrictive. This result agrees with the asymptotic behavior discovered for our CTC-CVRP models. Furthermore, the gap was also larger when considering a greater number of demand nodes n . This result suggests that there may be more opportunities for desirable lengthier routes when there is a greater density of demand nodes, as the service region did not vary in size. Overall, we know that these gaps determine the maximum amount of cycle time reduction that could be seen from an optimally solved IP solution with three stops and an optimally solved IP solution with six stops.

As discussed, another motivation for building a column generation procedure is that we can take the routes that were generated for the 6-stop LP relaxation and add them to our 3-stop IP decision space. The hope is that after solving this new IP the average cycle time per package can be reduced, ultimately result in an improved dispatching policy. We call this a heuristical policy for the 6-stop IP solution. As some of the generated routes

Table 4.1: Computational study results, column generation for CTC-CVRPs.

Scenario ID	n	m	Minutes per package:		Routes: generated / added	%Gap to LP relaxation:	
			3-stop IP	6-stop LP relaxation		3-stop IP	6-stop IP heuristic
A	12	4	136.3	125.6	15 / 12	8.6%	1.0%
B	12	4	113.5	111.6	15 / 6	1.7%	0.8%
C	12	4	92.3	78.7	19 / 7	17.3%	4.9%
D	12	6	79.3	74.7	14 / 3	6.1%	3.7%
E	12	6	105.8	103.4	8 / 3	2.3%	0.0%
F	12	6	89.4	82.5	25 / 10	8.3%	0.7%
G	24	8	118.9	103.9	43 / 26	14.5%	5.4%
H	24	8	109.2	90.5	33 / 13	20.6%	6.4%
I	24	8	107.4	91.7	56 / 28	17.2%	12.0%
J	24	12	77.1	70.3	54 / 29	9.7%	0.9%
K	24	12	99.6	88.3	64 / 25	12.8%	3.7%
L	24	12	87.4	81.5	45 / 22	7.2%	2.6%
M	36	12	107.1	83.9	105 / 65	27.7%	9.8%
N	36	12	96.8	83.3	117 / 71	16.3%	6.0%
O	36	12	118.9	95.1	111 / 67	25.0%	15.0%
P	36	18	80.2	72.3	68 / 40	10.8%	5.3%
Q	36	18	92.2	81.6	112 / 64	13.0%	2.4%
R	36	18	86.5	74.7	82 / 50	15.7%	5.9%

may visit some nodes more than once, in order to arrive at a sensible heuristic, it may be necessary to shortcut some of the generated routes before adding them to the $\mathcal{R}^+ = \Omega_3^3$ set. Such short-cutting here relies on the usage of the triangle inequality assumed in Section 4.2.

In Table 4.1 we can also observe the performance of our 6-stop IP heuristic. We record how many of the generated columns were transformed into routes with four, five, or six unique stops. From these scenarios, the largest nominal reduction in average package cycle time occurred in scenario M. After adding 65 routes discovered during the 6-stop LP relaxation to the 3-stop IP decision space, we were able to reduce the average cycle time per package by 15.0 minutes (14.0% reduction). This closes the gap to the LP relaxation to 9.8%, from an initial value of 27.7%. Future works may focus on improving these heuristics, developing more sophisticated column generation procedures for topology-specific data, or applying such results within a branch-cut-and-price framework.

4.6 Conclusions

In this chapter we introduced the concept of cycle time considering capacitated vehicle routing problems, or CTC-CVRPs. The motivation for these problems stems from the desire to decrease the average time packages spent within a delivery network. We note that other applications outside of package movement may exist for these models, especially in the realms of queuing theory and resource scheduling. Traditional VRP models tend to focus on the resource usage of the system or how routing policies affect the service provider. In contrast, our CTC-CVRP models consider the impact of routing policies on the units being served in the system. Factors such as average waiting times, demand weighted accumulated routing times, vehicle capacity constraints, and repeat servicing were all considered in our modeling. It is important to study models such as these in order to gain a more complete picture on how both service providers and service receivers are impacted in supply chain system design.

We utilize set partitioning formulations to model our CTC-CVRPs as such formulations lend themselves nicely to closely related problems in the literature and also form a basis for column generation procedures that can be used for dynamic route creation. Two set partitioning models were highlighted in this chapter. The first one, our Phase 1 model, focuses on determining the minimally sufficient fleet size necessary to continually service demand in the system. Our second model, unimaginatively named the Phase 2 model, has a cycle time objective and uses fleet sizing as a constraint. For the scope of our analysis, set partitioning models rather than set covering models, were chosen in order to simplify the modeling and subsequent analysis. Future works may relax the restriction that each demand node can only be serviced by one route in hope to even further reduce cycle time objectives.

We discussed column generation procedures and its uses in both the solving of LP relaxations as well as in solving IPs in the case that the decision space is exceptionally large. Even though our modeling objectives tend to support the concept of limiting overall route length, it can be advantageous to dynamically generate desirable delivery routes with a larger number of stops. In one computational experiment, we saw a 14.0% reduction in average package cycle time after a column generation improvement heuristic was added onto an initial IP solution. Future work here may focus on improving the underlying pricing problems and DP formulations.

CHAPTER 5

CONCLUSION

5.1 Summary

The primary focus of this dissertation is to aid in the tactical level decision making that goes into designing and developing of various last mile logistical systems. We have emphasized simplified modeling and transparent policy making for such systems, as long as operational decisions maintain fidelity at the aggregate level. With such models we are able to approximate the impact of various design choices on day-to-day operations. While we motivate our work around same-day and service-orientated systems, especially as they pertain to package delivery supply chains, our methods may be able to be applied to other disciplines within Operations Research.

In Chapter 2, we demonstrated that our proposed tactical model for same-day delivery successfully simplifies operations at the level of a single depot and its service region. By approximating the order arrival process and the vehicle routing times, we were able to derive simple and transparent optimal solutions for various system topologies. Using our model, a system manager can easily perform what-if analysis on various potential system configurations, and compare the cost and operating conditions of these configurations to decide various tactical questions, such as the size of the delivery fleet, the order cutoff time, or whether to have vehicles deliver to the entire service region versus partitioning the region by vehicle.

Then in Chapter 3, we studied the tactical design of SDD systems in which the service region is allowed to vary over the course of the service day. We again leveraged continuous approximations on order arrivals and vehicle routing times, knowing our focus on service region and order cutoff times as decision variables. We performed various structural anal-

yses and derived theoretical properties in order to maximize the number of orders served per day. In order to quantify the tangible value of allowing service areas to vary dynamically, we also calculate solutions to the problem of maximizing orders served with fixed service areas for each setting. Our structural analysis of optimal dispatching policies for these models exhibit some profound design implications for SDD systems. First of all, we were able to show that for many models that there is often a calculable optimal service area based on a set of known accumulation times. Next, while there is a benefit to using multiple vehicles in the same region for delivery, we explicitly quantified that this benefit is marginally decreasing. Additionally, while re-using a single vehicle for a second dispatch may be desirable (15.5% increase in orders served), using more than two dispatches seems to needlessly complicate the dispatching solution. Lastly, we have shown what gains can be expected by allowing a variable service region between dispatches over a fixed one. For example, when service regions are allowed to vary in a two vehicle, one dispatch each system, orders served stand to increase by 3.9%.

Lastly, in Chapter 4 we introduced the our cycle time considering capacitated vehicle routing problems, or CTC-CVRPs. The motivation for these problems stems from the desire to decrease the average time packages spent within a delivery network. These CTC-CVRP models consider the impact of routing policies on the units being served in the system. Factors such as average waiting times, demand weighted accumulated routing times, vehicle capacity constraints, and repeat servicing were all considered in our modeling. We utilize set partitioning formulations to model our CTC-CVRPs as such formulations lend themselves nicely to closely related problems in the literature and also form a basis for column generation procedures that can be used for dynamic route creation. Two set partitioning models were highlighted in this chapter. The first one focuses on determining the minimally sufficient fleet size necessary to continually service demand in the system, while the second has a cycle time objective and uses fleet sizing as a constraint. We discussed column generation procedures and its uses in both the solving of LP relaxations as well as

in solving IPs in the case that the decision space is exceptionally large. In one computational experiment, we saw a 14.0% reduction in average package cycle time after a column generation improvement heuristic was added onto an initial IP solution.

5.2 Conclusion

As mentioned in the epigraph: all models are wrong, but some are useful. The quality of output of any model is restricted by its input data, assumptions, and design considerations. This work has shown the viability of models which can be used to design last mile logistical systems intelligently. We hope that future academic works and practicing service providers are able to leverage our models, ideas, and results in guiding them the next time they are tasked with designing a system. We believe we were successful in adding to the existing Operations Research and broader supply chain literature by researching such topics.

5.3 Future Works

As motivated in Chapter 1, companies and consumers will continue to evolve in how they utilize last mile logistical systems. Whether new fulfillment strategies are introduced, or existing ones are pushed to their extremes, there will continue to be a need to design such systems intelligently. Everybody wants to innovate and have the next big supply chain idea. Whether its for growth, reinvention, or to change an existing value proposition, companies know they need to innovate or be left behind. They can get lucky once in awhile, but to have systematic and sustainable success, strategic design must be considered when innovating.

Appendices

APPENDIX A
TECHNICAL SUPPLEMENTATION FOR CHAPTER 2

A.1 Proof of Theorem 1

We start our proof by stating the optimization problem presented in Section 2.2 for the many-vehicle case. Without loss of generality, we can assume that no vehicle is used more than once. Here, a given dispatch policy will have the form $\{(t_d, q_d)\}_{d=1}^D$, where the d -th vehicle serves order quantity q_d at time t_d . For any such dispatch policy to be feasible, the following constraints must be satisfied:

$$\sum_{d=1}^D q_d = N, \quad (\text{A.1a})$$

$$q_d \geq 0 \quad \forall d, \quad (\text{A.1b})$$

$$t_d + f(q_d) \leq T \quad \forall d, \quad (\text{A.1c})$$

$$t_d \geq 0 \quad \forall d, \quad (\text{A.1d})$$

$$\sum_{\delta=1}^d q_\delta \leq t_d \quad \forall d. \quad (\text{A.1e})$$

An optimal dispatch policy minimizes $\sum_{d=1}^D f(q_d)$, subject to (A.1a)-(A.1e), over all $D \geq 1$. The Many-Vehicle Policy is feasible, satisfies (A.1c) at equality for $d = 1, 2, \dots, D-1$, and (A.1e) at equality for all d . By construction, the policy is unique for a given problem instance. Theorem 1 claims that this policy is optimal.

First, we prove that there exists some optimal dispatch policy, $\{(t_d^*, q_d^*)\}_{d=1}^{D^*}$, which uses a finite number of vehicles, D^* . For any problem instance to be feasible, there must be a quantity q' with $0 < q' \leq N$ and $f(q') \leq T - N$. Now consider any feasible dispatch policy that uses more than $2\lceil N/q' \rceil$ dispatches. There are at least two dispatches with $q'/2$ or fewer orders. By Lemma 2, we can consolidate these two dispatches into a single dispatch that

leaves at time $t = N$ while preserving feasibility, and without increasing the total dispatch time of the policy. Therefore, there exists some optimal dispatch policy $\{(t_d^*, q_d^*)\}_{d=1}^{D^*}$ with $D^* \leq 2\lceil N/q' \rceil$. Fix such a policy.

We now show that this policy can be transformed into one where all constraints (A.1e) hold at equality. We first reduce t_1^* until (A.1e) is satisfied at equality for $d = 1$. This process can be repeated for $d = 2, 3, \dots$ until we set $t_{D^*} = N$. As we only possibly decreased each time of dispatch, constraints (A.1a)-(A.1d) remain satisfied, and the objective value remains the same. This adjusted policy that satisfies all constraints (A.1e) at equality is also an optimal dispatch policy.

We next show that it is possible to re-order, if necessary, the dispatch quantities of the optimal policy so that they are non-increasing. Suppose two consecutive dispatches (t_d^*, q_d^*) and (t_{d+1}^*, q_{d+1}^*) have $q_d^* < q_{d+1}^*$; we can replace these two dispatches with $(t_d^* + (q_{d+1}^* - q_d^*), q_{d+1}^*)$ and (t_{d+1}^*, q_d^*) . The d -th dispatch now leaves with q_{d+1}^* units at an earlier time than t_{d+1}^* and thus is feasible. The new $(d+1)$ -th dispatch leaves at the same time as before, but now with fewer units, so it is also feasible. As the total dispatch time remains the same, this feasible dispatch remains optimal. Furthermore, all of the constraints in (A.1e) are still satisfied at equality by construction. By repeating this operation as necessary, we may assume our optimal dispatch policy has non-increasing dispatch quantities.

If constraint (A.1c) holds at equality for $d = 1, 2, \dots, D^* - 1$, we are done. Assume this is not the case. Take the first dispatch d' such that $t_{d'}^* + f(q_{d'}^*) < T$. This dispatch and its successor are given by $(t_{d'}^*, q_{d'}^*)$ and $(t_{d'+1}^*, q_{d'+1}^*)$. Replace these dispatches with $(t_{d'}^* + \sigma, q_{d'}^* + \sigma)$ and $(t_{d'+1}^*, q_{d'+1}^* - \sigma)$, where $\sigma = \min\{\varepsilon, q_{d'+1}^*\}$, and $\varepsilon > 0$ is the unique value with $t_{d'}^* + \varepsilon + f(q_{d'}^* + \varepsilon) = T$. By the definition of σ , this adjusted dispatch policy is feasible. Furthermore, since $q_{d'}^* \geq q_{d'+1}^*$, Lemma 2 implies that moving σ units from dispatch $d' + 1$ to d' cannot increase the total dispatch time; thus this adjusted policy remains optimal. See Figure A.1 for an illustration.

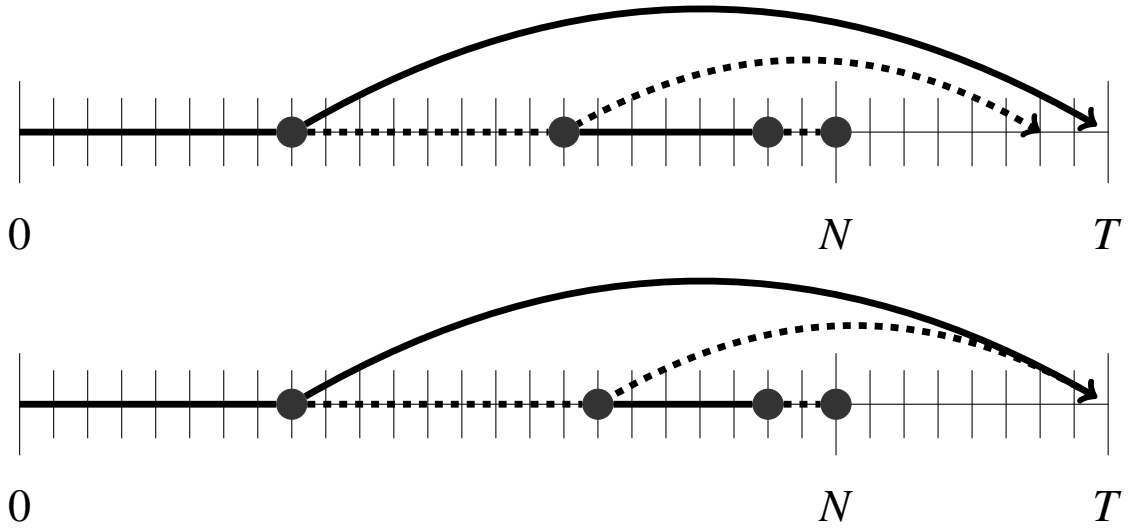


Figure A.1: If some dispatch, not the last one, returns to the depot before T , it would be better for this dispatch to stay at the depot longer before dispatching, “eating away” at the next dispatch. Note that the time of the next dispatch does not change.

If $\sigma = \varepsilon$, the resulting policy now has (A.1c) holding at equality for d' . If $\sigma = q_{d'+1}^*$, the resulting policy has one fewer dispatch. After either adjustment we can again reorder the remaining dispatches to be non-increasing if necessary. Thus, after this operation, we have either reduced the number of dispatches or the optimal policy satisfies the first d' constraints (A.1c) at equality. Furthermore, constraints (A.1e) still hold at equality by construction. Therefore, after a finite number of adjustments the resulting optimal policy satisfies (A.1c) at equality for all non-terminal dispatches.

We have constructed an optimal dispatch policy satisfying (A.1c) at equality for all non-terminal dispatches and (A.1e) at equality for all dispatches. This optimal dispatch policy is identical to the one prescribed by the Many-Vehicle Policy, so the first part of Theorem 1 is proved. The final claim of the theorem states that if the optimization problem presented in Section 2.2 is constrained to use $m < m^*$ vehicles, its objective value is bounded below by the Many-Vehicle Policy’s objective value. This follows because the Many-Vehicle Policy attains the minimum total dispatch time for any number of vehicles. \square

A.2 Proof of Lemma 3

Assume we are given an SDD problem instance with a single vehicle satisfying the sufficient processing speed and sufficient gap time conditions. We show that the instance has a feasible policy also satisfying the minimum dispatch size condition. First, let $D' = \lfloor N/q_{\min} \rfloor$.

Suppose $D' \leq 1$; then $N < 2q_{\min}$. Consider the dispatch policy $(t_1 = N, q_1 = N, 1)$. Because f is increasing we have $f(N) < f(2q_{\min})$. By the sufficient gap time condition, $N + f(N) < N + f(2q_{\min}) \leq T$, we see that the dispatch returns to the depot by the end of the service day and is therefore feasible. Additionally, note that the minimum dispatch size condition holds trivially.

Now assume $D' \geq 2$. Consider the dispatch policy

$$(t_1 = q_{\min}, q_1 = q_{\min}, 1), (t_2 = 2q_{\min}, q_2 = q_{\min}, 1), \dots, \\ (t_{D'-1} = (D'-1)q_{\min}, q_{D'-1} = q_{\min}, 1), (t_{D'} = N, q_{D'} = N - (D'-1)q_{\min}, 1).$$

By the sufficient processing speed condition, $f(q_{\min}) \leq q_{\min}$, which implies that the first $D' - 1$ dispatches return to the depot before q_{\min} more orders accumulate. Additionally, $q_{D'} \geq q_{\min}$, so the first $D' - 1$ dispatches return to the depot before the next dispatch must leave. The last dispatch takes all of the remaining orders. By the choice of D' we have that $q_{D'} < 2q_{\min}$, implying $f(q_{D'}) < f(2q_{\min})$ and subsequently $N + f(q_{D'}) < N + f(2q_{\min}) \leq T$ by the sufficient gap time condition. As the minimum dispatch size condition holds and the final dispatch returns to the depot by the end of the service day, the proposed policy is feasible and we are done. \square

A.3 Proof of Theorem 4

We start our proof by formulating the optimization problem presented in Section 2.2 for the single-vehicle case. A given dispatch policy in this case has the form $\{(t_d, q_d)\}_{d=1}^D$. For any such dispatch policy to be feasible, the following constraints must be satisfied:

$$\sum_{d=1}^D q_d = N, \quad (\text{A.2a})$$

$$q_d \geq 0 \quad \forall d, \quad (\text{A.2b})$$

$$t_D + f(q_D) \leq T, \quad (\text{A.2c})$$

$$t_d + f(q_d) \leq t_{d+1} \quad \forall d \leq D-1, \quad (\text{A.2d})$$

$$t_d \geq 0 \quad \forall d, \quad (\text{A.2e})$$

$$\sum_{\delta=1}^d q_\delta \leq t_d \quad \forall d. \quad (\text{A.2f})$$

An optimal policy minimizes $\sum_{d=1}^D f(q_d)$, subject to (A.2a)-(A.2f), over all $D \geq 1$. Additionally, Theorem 4 assumes the following conditions are met:

$$f(x) \leq x/\lambda = x \quad \forall x \geq q_{\min}, \quad (\text{A.3a})$$

$$q_d \geq q_{\min} \quad \forall d < D, \quad (\text{A.3b})$$

$$T - N \geq f(2q_{\min}). \quad (\text{A.3c})$$

Note that (A.3a), (A.3b) and (A.3c) correspond to the sufficient processing speed, minimum dispatch size and sufficient gap time conditions, respectively. We prove that there exists an optimal dispatch policy $\{(t_d^*, q_d^*)\}_{d=1}^{D^*}$ satisfying the following conditions:

- (C1) each dispatch takes all available unserved orders at the depot at the time of dispatch,
- (C2) after the first dispatch, the vehicle never waits at the depot again, and
- (C3) if the vehicle is dispatched more than once, the last dispatch arrives back at the depot

at exactly time $t = T$.

Mathematically, these conditions can be expressed as

$$(C1) \quad q_1^* = t_1^* \text{ and } \sum_{\delta=1}^d q_\delta^* = t_d^* \quad \forall d \leq D^* - 1,$$

$$(C2) \quad t_d^* + f(q_d^*) = t_{d+1}^* \quad \forall d \leq D^* - 1, \text{ and}$$

$$(C3) \quad \text{if } D^* \geq 2, \text{ then } t_{D^*}^* + f(q_{D^*}^*) = T.$$

We also prove that $q_1^* \geq q_2^* \geq \dots \geq q_{D^*}^*$ (C4), i.e. the policy's order quantities are non-increasing. We now assume that (A.3a) and (A.3c) hold for a given problem instance and that any feasible policy must satisfy (A.2a)-(A.2f) as well as (A.3b). By Lemma 3, such a feasible policy must exist.

If $N + f(N) \leq T$, a single dispatch is optimal and trivially satisfies (C1) through (C4). Now assume $N + f(N) > T$. Because of (A.2a) and (A.3b), a feasible dispatch policy cannot use more than $\lceil N/q_{\min} \rceil$ dispatches. From this we conclude that an optimal policy with finitely many dispatches exists and we can subsequently fix such a policy as $\{(t_d^*, q_d^*)\}_{d=1}^{D^*}$, where $D^* \geq 2$.

Suppose (A.2c) holds strictly; then we can increase $t_{D^*}^*$ until the constraint is binding without loss of optimality. Assuming (A.2c) is binding, if any of (A.2d) hold strictly, we can similarly increase dispatch times, beginning with $t_{D^*-1}^*$ and working backwards to t_1^* , until all constraints (A.2d) are also binding, without loss of optimality. This transformed policy satisfies (C2) and (C3).

Next, suppose the policy does not satisfy (C4). Fix the smallest index d such that $q_d^* < q_{d+1}^*$. Similarly to the proof of Theorem 1, we can replace dispatches (t_d^*, q_d^*) , (t_{d+1}^*, q_{d+1}^*) with (t_d^*, q_{d+1}^*) , $(t_d^* + f(q_{d+1}^*), q_d^*)$. This swap does not alter the total dispatch time, and (C2) and (C3) still hold after the change. The feasibility of the new solution follows by showing that the d -th dispatch can take q_{d+1}^* units:

$$t_d^* = t_{d+1}^* - f(q_d^*) \geq \sum_{\delta=1}^{d+1} q_\delta^* - f(q_d^*) \geq \sum_{\delta=1}^{d-1} q_\delta^* + q_{d+1}^*.$$

Note that $t_d^* = t_{d+1}^* - f(q_d^*)$ follows from (C2), $t_{d+1}^* \geq \sum_{\delta=1}^{d+1} q_\delta^*$ follows from (A.2f), and $q_d^* - f(q_d^*) \geq 0$ follows from (A.3a) as (A.3b) is satisfied. We can iteratively perform this operation as necessary, preserving feasibility and the total dispatch time, until the solution satisfies (C4).

We may now assume the policy satisfies (C2) through (C4), and suppose by contradiction that it does not satisfy (C1). Let $d < D^*$ be the first dispatch that violates (C1), so we have $\varepsilon = t_d^* - t_{d-1}^* - q_d^* > 0$ (with $t_0^* = 0$ if $d = 1$). As in the proof of Theorem 1, we consider shifting units from the $(d + 1)$ -th dispatch to the d -th. Let $\sigma = \min\{\varepsilon, q_{d+1}^* - q_{\min}\}$ if $d \leq D^* - 2$, or $\sigma = \varepsilon$ if $d = D^* - 1$. If $\sigma > 0$, we can redefine the two dispatches as $(t_d^*, q_d^* + \sigma)$, $(t_d^* + f(q_d^* + \sigma), q_{d+1}^* - \sigma)$. Alternatively, suppose $\sigma = 0$ and $d \leq D^* - 2$. Then (C4) implies $q_{d+1}^* = \dots = q_{D^*-1}^* = q_{\min} \geq q_{D^*}^*$. In this case, we can combine the quantities in the last two dispatches into a single dispatch: $q_{D^*-1}^* + q_{D^*}^* \leq 2q_{\min}$, while (A.3c), (C2) and (C3) imply $t_{D^*-1}^* > N$. So we can replace $(t_{D^*-1}^*, q_{D^*-1}^*)$, $(t_{D^*}^*, q_{D^*}^*)$ with a single dispatch $(t_{D^*-1}^*, q_{D^*-1}^* + q_{D^*}^*)$. In either case, the new policy will remain feasible and optimal while still satisfying (C2) and (C3); if it no longer satisfies (C4), we can proceed as before so it again satisfies this conditions. After this operation, the policy satisfies (C2) through (C4) and either also satisfies (C1), or we have increased the index of the first dispatch that does not satisfy (C1), or we have decreased the total number of dispatches used in the optimal policy. As with (C4), we can iteratively perform these policy transformations as necessary until the solution satisfies (C1), which completes the proof. \square

A.4 Proof of Corollary 5

Theorem 4 implies that the search for an optimal dispatch policy for a single-vehicle SDD problem instance satisfying the stated conditions can be restricted to policies satisfying (C1), (C2), and (C3). By (C2) and (C3), such an optimal policy has an objective value $T - \alpha^*$, where α^* is its time of first dispatch. Thus, over all feasible dispatch policies satisfying (C1), (C2), and (C3), the one maximizing its time of first dispatch must be optimal.

Feasible dispatch policies with D dispatches satisfying (C1) and (C2) have a time of first dispatch in $[\alpha_D, \alpha_{D-1})$, and the α_D values are decreasing in D , so we can further restrict our search by minimizing the number of dispatches. Finally, if two feasible dispatch policies satisfying (C1), (C2), and (C3) use the same number of dispatches, the one with the later time of first dispatch has a lower objective value. Thus, Algorithm 1 returns the optimal time of first dispatch. \square

A.5 Proof of Theorem 6

Assume the instance satisfies the sufficient gap time and sufficient processing speed conditions. Then Lemma 3 implies the instance is feasible for a single vehicle. In the hybrid policy, if the $(m-1)$ -th vehicle departs at time t_{m-1} , then the first $m-1$ vehicles serve $\lambda t_{m-1} = t_{m-1}$ orders, leaving the remaining $N' := N - t_{m-1}$ orders to be served by the m -th vehicle; since all previous conditions still hold in this reduced problem, the last vehicle can feasible serve these remaining orders.

The only potential concern is whether any of the first $m-1$ dispatches serve a quantity smaller than q_{\min} , which would violate the minimum dispatch size constraint. However, this is impossible, as all of these dispatches depart before time N and return at time T ; this implies a dispatch time for each of these dispatches greater than $T - N \geq f(2q_{\min})$, by the sufficient gap time condition, which means each dispatch serves at least $2q_{\min}$ orders.

For the second part of the Theorem, assume $f(n) = bn + c\sqrt{n}$. Given that the heuristic uses $m-1 + D_m$ dispatches in total, the Many-Vehicle Policy must also use no more than $m-1 + D_m$ vehicles. Let z^{many} represent the objective value of the Many-Vehicle Policy, z^{m-1+D_m} represent the objective value of the policy using $m-1 + D_m$ vehicles and z^m represent the objective value of the optimal policy where m vehicles are used. Finally, let $z^{\text{Hm}D_m}$ represent the objective value of the heuristic policy, using m vehicles where the last vehicle is dispatched D_m times. Thus, we know that $z^{\text{many}} = z^{m-1+D_m} \leq z^m \leq z^{\text{Hm}D_m}$ and our objective is to show that $z^{\text{Hm}D_m} \leq \frac{m-1+D_m\sqrt{D_m}}{m-1+D_m} z^{\text{many}} \leq \frac{m-1+D_m\sqrt{D_m}}{m-1+D_m} z^m$.

Let z_1^{many} represent the total dispatch time in the Many-Vehicle Policy of using the first $m - 1$ vehicles, while z_2^{many} represents the remaining dispatch time from the m -th dispatch onward. Similarly, let z_1^{HmDm} represent the total dispatch time in the heuristic of using the first $m - 1$ vehicles, while z_2^{HmDm} represents the remaining dispatch time from the last vehicle, which dispatches D_m times from the depot. We have that $z_1^{many} = z_1^{HmDm}$ by construction of the heuristic. If we assume that some N' orders remain to be dispatched after each policy identically dispatches the first $m - 1$ vehicles, then by concavity we maximize the total dispatch time of the (hybrid policy) last vehicle's D_m dispatches if they are all equal, each serving $\frac{N'}{D_m}$ orders; this has a total dispatch time of $D_m f(\frac{N'}{D_m})$. The most effective (though possibly infeasible) policy for the remaining N' orders would be to serve them together, incurring a dispatch time of $f(N')$. Therefore, $f(N') \leq z_2^{many} \leq z_2^{HmDm} \leq D_m f(\frac{N'}{D_m})$. As we have $f(n) = bn + c\sqrt{n}$, it follows that $D_m f(\frac{N'}{D_m}) \leq \sqrt{D_m} f(N')$. This implies that $z_2^{HmDm} \leq \sqrt{D_m} z_2^{many}$.

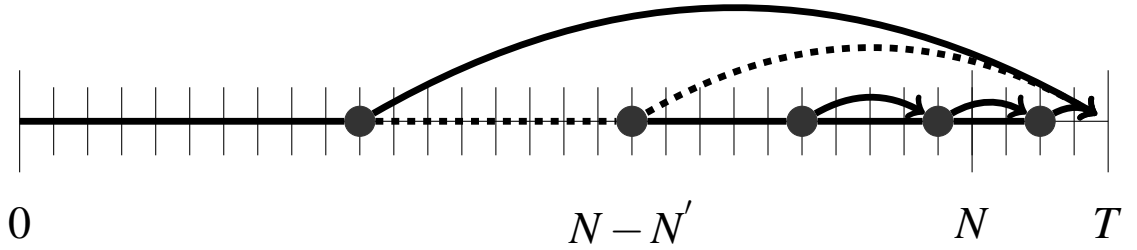


Figure A.2: Visual representation of the heuristic with $m = 3$ and $D_m = 3$. The first two dispatches account for the cost in z_1^{HmDm} while the last three dispatches account for the cost in z_2^{HmDm} .

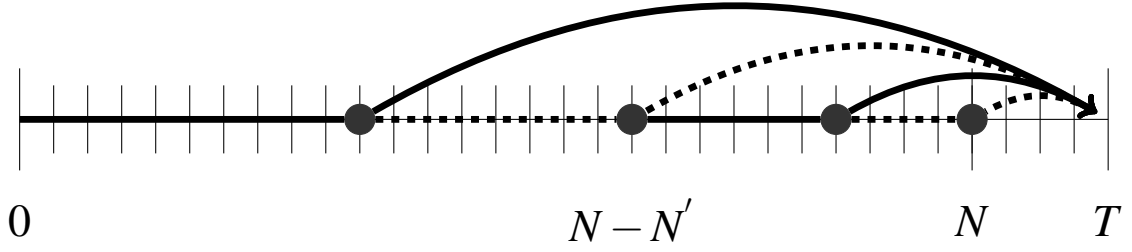


Figure A.3: Visual representation of many-vehicle policy over the same problem instance as in Figure A.2. Note that only 4 vehicles are used. The first two dispatches account for the cost in z_1^{many} while the last two dispatches account for the cost in z_2^{many}

Because the dispatch sizes are non-increasing in the MVP, the time of every one of the first $m - 1$ dispatches, which sum to z_1^{many} , is greater than or equal to that of any of the remaining dispatches, which sum to z_2^{many} . Thus, $z_1^{many} \geq \frac{m-1}{m-1+D_m} z^{many}$, which implies that $z_2^{many} \leq \frac{D_m}{m-1+D_m} z^{many}$. Combining the two, $z^{HmD_m} = z_1^{HmD_m} + z_2^{HmD_m} \leq z_1^{many} + \sqrt{D_m} z_2^{many} = z^{many} + (\sqrt{D_m} - 1) z_2^{many} \leq z^{many} (1 + \frac{D_m(\sqrt{D_m}-1)}{m-1+D_m}) = z^{many} (\frac{m-1+D_m\sqrt{D_m}}{m-1+D_m}) \leq z^m (\frac{m-1+D_m\sqrt{D_m}}{m-1+D_m})$, as desired. \square

A.6 Proof of optimality for the Generalized MVP

In the case that $N' \leq Q$, we can transform the problem instance of N', N, f, T to $\bar{N}' = 0, \bar{N} = N + N', f, \bar{T} = T + N'$. This transformed instance serves the same amount of total orders and is subject to the same gap time between the end of service day and the order cutoff time as in the original problem instance. We know that we can use the MVP to solve for the optimal policy in such an instance. Fix this optimal policy as $\{(\bar{t}_d^*, \bar{q}_d^*)\}_{d=1}^{D^*}$. It must be the case that $\bar{t}_1^* = \bar{q}_1^* \geq N'$ as $N' + f(N') \leq N' + f(Q) = N' + T = \bar{T}$. It follows that we can subtract N' time units from each \bar{t}_d^* and to obtain the policy $\{(\bar{t}_d^* - N', \bar{q}_d^*)\}_{d=1}^{D^*}$ which is feasible in the original problem instance. This policy must also be optimal in the original instance as it is optimal in the transformed instance, which is a relaxation of the original instance. Thus in the case that $N' \leq Q$ the generalized MVP is an optimal policy as it exactly prescribes the same dispatch sizes of the transformed policy.

In the case that $N' > Q$, it follows from Lemma 2 that any feasible dispatching policy that does not leave with as many as possible orders of size Q at time $t = 0$, as prescribed in the generalized MVP, can be transformed into one which does without increasing the total dispatch time of the policy. After these dispatches occur at time $t = 0$ some $0 \leq \bar{N}' < Q$ orders will remain at the depot at the start of the service day. We have already shown that the generalized MVP is optimal in the case of $N' \leq Q$ so the remaining orders are also served optimally, and thus we are done. \square

A.7 Proof of Theorem 8

For cutoff times in the interval of $(N_i, N_{i+1}]$ we know that exactly $i + 1$ vehicles will be used in the MVP, and that $g(N) = f(N_1) + f(N_2 - N_1) + \dots + f(N_i - N_{i-1}) + f(N - N_i)$ is concave in N (within the interval). It follows that $\pi(N)$ on $(N_i, N_{i+1}]$ is convex and is therefore maximized at one of its endpoints. Thus to maximize $\pi(N)$ over $N \in [0, U]$ we only need to consider the breakpoints of N_1, N_2, \dots, N_U as well as the endpoints of 0 and U . This completes the proof. \square

A.8 Proof of Proposition 9

We split this proof into cases. Assume we are given a U such that $0 \leq U \leq T - f(2q_{\min})$ and $i_{\max} \leq 2$. Suppose $U = 0$, then trivially we have $N^* = 0$. So, assume that $U > 0$, and therefore $i_{\max} \in \{1, 2\}$.

Assume that $i_{\max} = 1$. Then the range $0 \leq N \leq U$ can be partitioned as $N \in \{0\} \cup (0, U]$. We know from Theorem 4 that given $N \in (0, U]$, the optimal dispatch policy is given by a single dispatch of size N at cost $f(N)$. Therefore, $\pi(N)$, is a convex function over the interval $(0, U]$. Thus, either $N = 0$, or $N = U$ will maximize the profit function.

Now assume that $i_{\max} = 2$. Then the range $0 \leq N \leq U$ can be partitioned as $N \in \{0\} \cup (0, \bar{N}_1] \cup (\bar{N}_1, U]$. We know from Theorem 4 that given $N \in (\bar{N}_1, U]$, the optimal dispatch policy can be fully described by the time of first departure α_N . Additionally we

know $g(N) = T - \alpha_N$, and $\alpha_N + f(\alpha_N) + f(N - \alpha_N) = T$. Which means we can write $N = \alpha_N + f^{-1}(T - \alpha_N - f(\alpha_N))$. Thus N can be written as a convex function of α_N , and thus $g(N)$ is a concave function with respect to N . Thus $\pi(N)$ is a convex function over the interval $(\bar{N}_1, U]$. From before we also have that $\pi(N)$ is a convex function over the interval $(0, \bar{N}_1]$. Thus the solution to the profit maximization function can be found at $N = 0$, $N = \bar{N}_1$, or $N = U$. Thus, the claim is proven. \square

A.9 A Posteriori Formulations

We use the following integer programming formulations to compute *a posteriori* solutions in our computational study.

Parameters

- Node set: order locations $L = \{1, \dots, n\}$, depot 0
- Arc set: ordered pairs of nodes, $a = (i, j)$.
- Travel time: τ_{ij} , $i, j \in L \cup 0$, includes depot setup time and order service time as necessary
- Release time: $r_i \geq 0$, $i \in L$, the time when order i is ready
- Deadline: T
- Fleet size or number of routes: K ($K = 2$ in our experiments)

Decision Variables

- x_{ij}^k : indicates if vehicle/route k goes from i to j
- d_k : departure time of vehicle/route k

The many-vehicle formulation is then given by

$$\min_{d,x} \sum_{k=1}^K \sum_{i,j} \tau_{ij} x_{ij}^k \quad (\text{A.4a})$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} x_a^k = \sum_{a \in \delta^-(i)} x_a^k, \quad i \in L \cup 0, \quad k = 1, \dots, K \quad (\text{A.4b})$$

$$\sum_{k=1}^K \sum_{a \in \delta^+(i)} x_a^k = 1, \quad i \in L \quad (\text{A.4c})$$

$$\sum_{a \subseteq S} x_a^k \leq |S| - 1, \quad S \subseteq L, \quad k = 1, \dots, K \quad (\text{A.4d})$$

$$d_k + \sum_{i,j} \tau_{ij} x_{ij}^k \leq T, \quad k = 1, \dots, K \quad (\text{A.4e})$$

$$d_k - r_i \sum_{a \in \delta^+(i)} x_a^k \geq 0, \quad i \in L, \quad k = 1, \dots, K \quad (\text{A.4f})$$

$$d_k \geq 0, \quad x_{ij}^k \in \{0, 1\}. \quad (\text{A.4g})$$

In the formulation, (A.4b) ensures flow balance; (A.4c) requires each order to be served by a vehicle; (A.4d) eliminates subtours; (A.4e) establishes a route duration limit, so each vehicle returns by T ; (A.4f) prevents a vehicle serving i from departing the depot before the order is ready.

In the single-vehicle case, assume routes are indexed in order of departure. For all routes except K , we replace (A.4e) with

$$d_k - d_{k+1} + \sum_{i,j} \tau_{ij} x_{ij}^k \leq 0, \quad k = 1, \dots, K - 1. \quad (\text{A.5})$$

This ensures route k finishes before $k + 1$ begins, so one vehicle can perform all routes.

APPENDIX B
TECHNICAL SUPPLEMENTATION FOR CHAPTER 3

B.1 Proof of Property 10

For a fixed accumulation time of $\tau_1 \in (0, 1]$, inequality (3.2b) constrains the service area by, $A_1 \leq \frac{1-\tau_1}{c\sqrt{\tau_1}}$. As the objective value (3.2a) scales linearly with A_1 , we have that $A_1 = \frac{1-\tau_1}{c\sqrt{\tau_1}}$ must maximize the number of orders fulfilled for a fixed τ_1 . \square

B.2 Proof of Property 11

First consider the case where $B \geq \frac{2}{c\sqrt{3}}$. In this case, the optimal dispatch policy for the unconstrained-area problem of $\tau_1^* = \frac{1}{3}$, $A_1^* = \frac{2}{c\sqrt{3}}$ is feasible and therefore optimal for the constrained-area problem. Furthermore, it is also the dispatch policy prescribed by Property 11, which proves this case.

Now consider the case where $B < \frac{2}{c\sqrt{3}}$. Let us define time τ_B as the time which solves $\tau_B + cB\sqrt{\tau_B} = 1$ over $\tau_B \in (0, 1]$. As $\tau_B + cB\sqrt{\tau_B}$ is strictly increasing with respect to τ_B over this domain, it must be a unique solution to the equation. Furthermore, as $B < \frac{2}{c\sqrt{3}}$, $\tau_B \in (\frac{1}{3}, 1]$. For any given $\tau_1 \in [0, \tau_B]$ the optimal service area choice is to set $A_1 = B$, and for any given $\tau_1 \in [\tau_B, 1]$ the optimal service area choice is to set $A_1 = \frac{1-\tau_1}{c\sqrt{\tau_1}}$. It follows that the maximal number of orders served is equal to $\lambda B\tau_1$ when $\tau_1 \in [0, \tau_B]$, which is maximized when $\tau_1 = \tau_B$. Additionally, the maximal number of orders served is equal to $\frac{\lambda}{c}(1-\tau_1)\sqrt{\tau_1}$ when $\tau_1 \in [\tau_B, 1]$, which is also maximized when $\tau_1 = \tau_B$. Thus in the case that $B < \frac{2}{c\sqrt{3}}$, we have that $\tau_1^* = \tau_B$ and $A_1^* = B$. As this is the dispatch policy prescribed by Property 11, this case is proven as well. \square

B.3 Proof of Property 12

Given a set of fixed, positive, accumulation times, $\{\tau_1, \tau_2, \dots, \tau_m\}$, consider the d -th vehicle to dispatch. Inequality (3.4b) constrains the service area by: $A_d \leq \frac{1 - \sum_{\delta=1}^d \tau_\delta}{c\sqrt{\tau_d}}$. As the objective value (3.4a) increases linearly with each A_d , we have that $A_d = \frac{1 - \sum_{\delta=1}^d \tau_\delta}{c\sqrt{\tau_d}}$ maximizes the number of orders fulfilled for the d -th vehicle. As this is true for all vehicles d , we are done. \square

B.4 Proof of Theorem 13

Consider the optimization problem for the $(m+1)$ v1d model:

$$\frac{\lambda}{c} \max \sum_{d=1}^{m+1} \left(1 - \sum_{\delta=1}^d \tau_\delta\right) \sqrt{\tau_d} \quad (\text{B.1a})$$

$$\text{s.t.} \quad \sum_{d=1}^{m+1} \tau_d \leq 1, \quad (\text{B.1b})$$

$$\tau_d \geq 0 \quad \forall d. \quad (\text{B.1c})$$

We can re-formulate the problem as follows:

$$\frac{\lambda}{c} \max \left((1 - \tau_1) \sqrt{\tau_1} + \sum_{d=2}^{m+1} \left((1 - \tau_1) - \sum_{\delta=2}^d \tau_\delta \right) \sqrt{\tau_d} \right) \quad (\text{B.2a})$$

$$\text{s.t.} \quad \sum_{d=2}^{m+1} \tau_d \leq (1 - \tau_1), \quad (\text{B.2b})$$

$$\tau_d \geq 0 \quad \forall d. \quad (\text{B.2c})$$

Note that, given a value of $\tau_1 \in [0, 1)$, choosing the optimal values for $\tau_2, \dots, \tau_{m+1}$ equates to solving the $mv1d$ model over a reduced service day. With this in mind, define τ'_d such that $\tau'_d = \frac{\tau_d}{(1 - \tau_1)}$ for all $d \geq 2$ in order to equate the remaining accumulation times as proportions of the remaining service day. Thus, we can again re-formulate the problem as:

$$\frac{\lambda}{c} \max \left((1 - \tau_1) \sqrt{\tau_1} + (1 - \tau_1)^{1.5} \sum_{d=2}^{m+1} \left(1 - \sum_{\delta=2}^d \tau'_\delta \right) \sqrt{\tau'_d} \right) \quad (\text{B.3a})$$

$$\text{s.t.} \quad \sum_{d=2}^{m+1} \tau'_d \leq 1, \quad (\text{B.3b})$$

$$\tau_1 \leq 1, \quad (\text{B.3c})$$

$$\tau_d \geq 0 \quad \forall d. \quad (\text{B.3d})$$

There are no constraints in this optimization problem linking the τ'_d decision variables to the τ_1 decision variable. Thus we can independently optimize for the τ'_d decision variables; this equates to solving the m v1d model to optimality. By the presumptions of the Theorem, we have that $\tau_d^* = \tau_{m,d-1}^*$ for all $d \geq 2$. Additionally, by Property 12, we have that $A_d^* = A_{m,d-1}^*$ for all $d \geq 2$. What remains in the $(m+1)$ v1d model is to optimize:

$$\max_{0 \leq \tau_1 \leq 1} \frac{\lambda}{c} (1 - \tau_1) \sqrt{\tau_1} + (1 - \tau_1)^{1.5} z_m,$$

which proves the first claim given in Theorem 13. Once this function is optimized for τ_1^* , we can use Property 12 to determine that $A_1^* = \frac{1}{c} (1 - \tau_1^*) (\tau_1^*)^{-0.5}$. Furthermore, we can translate the optimal (τ'_d, A'_d) decision variables back to the (τ_d, A_d) decision space by performing the updates:

$$\tau_d^* \leftarrow (1 - \tau_1^*) \tau_d'^* = (1 - \tau_1^*) \tau_{m,d-1}^* \quad \forall d \geq 2$$

and

$$A_d^* \leftarrow \frac{1 - \sum_{\delta=1}^d \tau_\delta^*}{\sqrt{\tau_d^*}} = (1 - \tau_1^*)^{0.5} \frac{1 - \sum_{\delta=2}^d \tau_\delta'^*}{\sqrt{\tau_d'^*}} = (1 - \tau_1^*)^{0.5} A_d'^* = (1 - \tau_1^*)^{0.5} A_{m,d-1}^* \quad \forall d \geq 2,$$

which completes the proof. \square

B.5 Proof of Property 14

Fix a number of vehicles m . Consider the objective function of the one variable optimization problem for $m + 1$ vehicles as given in Theorem 13:

$$\frac{\lambda}{c}(1 - \tau_{m+1,1})\sqrt{\tau_{m+1,1}} + (1 - \tau_{m+1,1})^{1.5}z_m.$$

When solving for the optimal value of $\tau_{m+1,1} \in [0, 1]$, first order conditions imply that $\tau_{m+1,1}^*$ is the unique value of $\tau_{m+1,1} \in (0, \frac{1}{3}]$ which satisfies the equation

$$\frac{1}{3} = \tau_{m+1,1} + \frac{c}{\lambda}z_m\sqrt{(\tau_{m+1,1})(1 - \tau_{m+1,1})}.$$

By the uniqueness of $\tau_{m+1,1}^*$ and the fact that $\tau_{m+1,1}^* \neq 0$, we are able to claim that $z_{m+1} > z_m$. From this, it directly follows that $\tau_{m+2,1}^* < \tau_{m+1,1}^*$ as $\tau_{m+2,1}^*$ is the unique value of $\tau_{m+2,1} \in (0, \frac{1}{3}]$ which satisfies the equation

$$\frac{1}{3} = \tau_{m+2,1} + \frac{c}{\lambda}z_{m+1}\sqrt{(\tau_{m+2,1})(1 - \tau_{m+2,1})}.$$

Thus, the first part of Property 14 is proven.

Now consider an m vehicle, one dispatch each policy where each vehicle (feasibly) accumulates orders for $\frac{1}{m+1}$ units of time, that is, $\tau_d = \frac{1}{m+1}$ for all d . By Property 12, we would like these vehicles to each serve a maximal area of $A_d = \frac{1 - \sum_{\delta=1}^d \tau_\delta}{c\sqrt{\tau_d}}$. It follows that the total number of orders served by this policy is equal to

$$\frac{\lambda}{c} \sum_{d=1}^m \left(1 - \sum_{\delta=1}^d \tau_\delta\right) \sqrt{\tau_d} = \frac{\lambda}{c} \sum_{d=1}^m \left(1 - \frac{d}{m+1}\right) \sqrt{\frac{1}{m+1}} = \frac{\lambda m}{2c\sqrt{m+1}} > \frac{\lambda}{4c}\sqrt{m},$$

which tends to infinity as $m \rightarrow \infty$. As the optimal m vehicle policy serves at least as many orders as this policy, it follows that $z_m \rightarrow \infty$; specifically, $z_m = \Omega(\sqrt{m})$.

We next show that $z_m = \mathcal{O}(\sqrt{m})$ by constructing an upper bound. We relax the problem

by removing the vehicle return deadline, instead limiting the duration of each dispatch to not exceed 1. The relaxed problem is as follows:

$$\max \quad \lambda \sum_{d=1}^m A_d \tau_d \quad (\text{B.4a})$$

$$\text{s.t.} \quad cA_d \sqrt{\tau_d} \leq 1 \quad \forall d \in [m], \quad (\text{B.4b})$$

$$\sum_{d=1}^m \tau_d \leq 1, \quad (\text{B.4c})$$

$$A_d, \tau_d \geq 0 \quad \forall d \in [m]. \quad (\text{B.4d})$$

Without loss of optimality, we may assume the constraints (B.4b) hold at equality. This implies that, for all $d \in [m]$, $A_d = \frac{1}{c\sqrt{\tau_d}}$. As such, we may rewrite the problem without the A_d variables:

$$\max \quad \frac{\lambda}{c} \sum_{d=1}^m \sqrt{\tau_d} \quad (\text{B.5a})$$

$$\text{s.t.} \quad \sum_{d=1}^m \tau_d \leq 1, \quad (\text{B.5b})$$

$$\tau_d \geq 0 \quad \forall d \in [m]. \quad (\text{B.5c})$$

The optimal solution to this problem is $\tau_1 = \tau_2 = \dots = \tau_m = \frac{1}{m}$. The corresponding objective value is $\frac{\lambda}{c} m \sqrt{\frac{1}{m}} = \frac{\lambda}{c} \sqrt{m}$. Thus, for all m , $z_m \leq \frac{\lambda}{c} \sqrt{m}$, implying $z_m = \mathcal{O}(m)$. We conclude that $z_m = \Theta(m)$.

What remains to be seen is that as $m \rightarrow \infty$, $\tau_{m,1}^* \rightarrow 0$. By the construction of $\tau_{m+1,1}^*$ via

the first order conditions described above, we have that

$$\begin{aligned}
z_{m+1} &= \frac{\lambda}{c} (1 - \tau_{m+1,1}^*) \sqrt{\tau_{m+1,1}^*} + (1 - \tau_{m+1,1}^*)^{1.5} z_m \\
&= \frac{\lambda}{c} \frac{1 - \tau_{m+1,1}^*}{\sqrt{\tau_{m+1,1}^*}} \left(\tau_{m+1,1}^* + \frac{c}{\lambda} z_m \sqrt{(\tau_{m+1,1}^*)(1 - \tau_{m+1,1}^*)} \right) \\
&= \frac{\lambda}{c} \frac{1 - \tau_{m+1,1}^*}{3\sqrt{\tau_{m+1,1}^*}}.
\end{aligned}$$

As we know that $z_m \rightarrow \infty$ as $m \rightarrow \infty$, we can equivalently state that as $m \rightarrow \infty$, $\frac{1 - \tau_{m,1}^*}{3\sqrt{\tau_{m,1}^*}} \rightarrow \infty$.

This implies that $\tau_{m,1}^* \rightarrow 0$, which completes the proof. \square

B.6 Proof of Property 15

Fix a number of vehicles m . Consider the objective function of the one variable optimization problem given in Theorem 13. It can be shown via first order conditions that $\tau_{m+1,1}^*$ is the unique value of $\tau_{m+1,1} \in (0, \frac{1}{3}]$ which satisfies the equation $1 = 3\tau_{m+1,1} + \frac{3c}{\lambda} z_m \sqrt{(\tau_{m+1,1})(1 - \tau_{m+1,1})}$ and that $z_{m+1} = \frac{\lambda}{c} \frac{1 - \tau_{m+1,1}^*}{3\sqrt{\tau_{m+1,1}^*}}$ (see the proof of Property 14). This leads to the relation

$$z_{m+1} - z_m = \frac{\lambda}{c} \left(\frac{1 - \tau_{m+1,1}^*}{3\sqrt{\tau_{m+1,1}^*}} - \frac{1 - 3\tau_{m+1,1}^*}{3\sqrt{(\tau_{m+1,1}^*)(1 - \tau_{m+1,1}^*)}} \right),$$

which decreases as $\tau_{m+1,1}^*$ decreases and tends to 0 as $\tau_{m+1,1}^*$ tends to 0. Since $\tau_{m+1,1}^*$ is decreasing as m increases by Property 14, we have that $z_{m+1} - z_m$ also decreases as m increases. Therefore, we can conclude that $(z_{m+2} - z_{m+1}) < (z_{m+1} - z_m)$. Additionally, since $\tau_{m+1,1}^* \rightarrow 0$ as $m \rightarrow \infty$ by Property 14, we have that $(z_{m+1} - z_m) \rightarrow 0$ as $m \rightarrow \infty$. \square

B.7 Proof of Property 16

For the sake of induction, assume for a given m that $\tau_{m,1}^* < \tau_{m,2}^* < \dots < \tau_{m,m}^*$. We now show that it must be true that $\tau_{m+1,1}^* < \tau_{m+1,2}^* < \dots < \tau_{m+1,m+1}^*$. By Theorem 13, we have that $\tau_{m+1,d}^* = (1 - \tau_{m+1,1}^*)\tau_{m,d-1}^*$ for all $d \geq 2$. Thus, we can infer that $\tau_{m+1,2}^* < \tau_{m+1,3}^* < \dots < \tau_{m+1,m+1}^*$ by induction. What remains to be seen is if $\tau_{m+1,1}^* < \tau_{m+1,2}^*$.

Consider the objective value given in (3.5a), $\frac{\lambda}{c} \sum_{d=1}^m (1 - \sum_{\delta=1}^d \tau_{\delta}) \sqrt{\tau_d}$. The only term of this summation that depends on either $\tau_{m+1,1}$ or $\tau_{m+1,2}$, but not their sum, is given by

$$\frac{\lambda}{c} (1 - \tau_{m+1,1}) \sqrt{\tau_{m+1,1}} + \frac{\lambda}{c} (1 - \tau_{m+1,1} - \tau_{m+1,2}) \sqrt{\tau_{m+1,2}}. \quad (\text{B.6})$$

We claim that (B.6) can never be maximized when $\tau_{m+1,1} \geq \tau_{m+1,2}$. Consider a fixed $\theta = \tau_{m+1,1} + \tau_{m+1,2}$, and note that $0 < \theta \leq 1$. We can rewrite (B.6), without the multiplicative constant, as a function $h_{\theta} : [0, \theta] \rightarrow \mathbb{R}$ of $\tau_{m+1,1}$ that we wish to maximize in the interval $\tau_{m+1,1} \in [0, \theta]$:

$$h_{\theta}(\tau_{m+1,1}) = (1 - \tau_{m+1,1}) \sqrt{\tau_{m+1,1}} + (1 - \theta) \sqrt{\theta - \tau_{m+1,1}}.$$

Differentiating once and twice gives

$$h'_{\theta}(\tau_{m+1,1}) = \frac{1 - 3\tau_{m+1,1}}{2\sqrt{\tau_{m+1,1}}} - \frac{1 - \theta}{2\sqrt{\theta - \tau_{m+1,1}}},$$

$$h''_{\theta}(\tau_{m+1,1}) = -\frac{1}{4\tau_{m+1,1}^{3/2}} - \frac{3}{4\sqrt{\tau_{m+1,1}}} - \frac{1 - \theta}{4(\theta - \tau_{m+1,1})^{3/2}}.$$

Observe that $h'_{\theta}(\theta/4) > 0$, $h'_{\theta}(\theta/2) < 0$, and $h''_{\theta}(\tau_{m+1,1}) < 0$ for all $\tau_{m+1,1} \in (0, \theta)$. It follows that h_{θ} 's unique maximizer is located in the interval $(\theta/4, \theta/2)$, implying that (B.6) can never be maximized when $\tau_{m+1,1} \geq \tau_{m+1,2}$ (i.e., when $\tau_{m+1,1} \geq \theta/2$). Thus, we have that $\tau_{m+1,1}^* < \tau_{m+1,2}^*$.

To finish our proof by induction, what remains to be seen is if a base case value of m

yields $\tau_{m,1}^* < \tau_{m,2}^* < \dots < \tau_{m,m}^*$. From Table 3.1 we see that this is indeed true for $m = 2$. Thus, we have shown that accumulation times are strictly increasing throughout the service day. From this fact, and Property 12, it directly follows that the optimal service areas are strictly decreasing throughout the service day. \square

B.8 Proof of Property 17

Given a fixed value for $A > 0$, let us assume for the sake of contradiction that there exists such optimal dispatch policy $\{(\tau_d^*, A)\}_{d=1}^m$ such that at least one of the inequalities defined by constraint (3.6b) hold strictly. Fix such a policy and consider the first dispatch, d' , such that, $\sum_{\delta=1}^{d'} \tau_{\delta} + cA\sqrt{\tau_{d'}} < 1$. There must exist some $\varepsilon \in (0, 1)$ such that $\sum_{\delta=1}^{d'} \tau_{\delta} + \varepsilon + cA\sqrt{\tau_{d'} + \varepsilon} = 1$. We can then feasibly replace the d' -th dispatch with $(\tau_{d'}^* + \varepsilon, A)$ by removing the next ε orders from the subsequent dispatch(es). Any remaining dispatches will remain feasible since they are either: completely removed from the dispatch policy (i.e., their quantity is set to zero), set to depart at the same time of day with strictly less orders to serve, or set to depart at the same time of day with the exact same order amount to serve as before. This process of shifting orders to earlier dispatches can be repeated until (3.6b) holds at equality for each of the first $m - 1$ dispatches. Eventually, it will be the case that (3.6b) holds strictly for $d = m$. Then, this last dispatch could feasibly serve some $\delta > 0$ additional orders, which contradicts the assumed optimality of the given initial dispatch policy. Thus, constraints (3.6b) must hold at equality for all d in an optimal solution.

Lastly, knowing that $\sum_{\delta=1}^d \tau_{\delta} + cA\sqrt{\tau_d} = 1$ for all d , solving for τ_d we have

$$\tau_d = R_d + \frac{cA}{2} \left(cA - \sqrt{(cA)^2 + 4R_d} \right) \quad \forall d \geq 1,$$

where $R_1 = 1$ and $R_d = 1 - \sum_{\delta=1}^{d-1} \tau_{\delta}$ for all $d \geq 2$ via the quadratic formula. \square

B.9 Proof of Theorem 18

We prove a series of results which together imply the correctness of Algorithm 2.

Lemma 26. *Let z_m denote the optimal objective value of the unconstrained mv1d problem, and let $\frac{\lambda}{c}\hat{z}_m = z_m$. For notational convenience, let $\theta_m = \tau_{m,1}^*$ denote the corresponding optimal first dispatch time. Then,*

$$\hat{z}_m \sqrt{\frac{1-\theta_m}{1-2\theta_m}} - \sqrt{\frac{1-\theta_m}{\theta_m}} \leq 0 \quad (\text{B.7})$$

for all m .

Proof. By the proof of Property 14, we have that

$$\hat{z}_m = \frac{1-\theta_m}{3\sqrt{\theta_m}}. \quad (\text{B.8})$$

By the results of the 1v1d model and Property 14, we know that $\theta_m \leq \frac{1}{3}$. Additionally, observe that for all $\theta_m \in (0, \frac{1}{3}]$,

$$\frac{1-\theta_m}{3\sqrt{\theta_m}} \leq \sqrt{\frac{1-2\theta_m}{\theta_m}}. \quad (\text{B.9})$$

Hence,

$$\hat{z}_m \leq \sqrt{\frac{1-2\theta_m}{\theta_m}}, \quad (\text{B.10})$$

which implies

$$\hat{z}_m \sqrt{\frac{1}{1-2\theta_m}} \leq \sqrt{\frac{1}{\theta_m}}, \quad (\text{B.11})$$

which in turn implies

$$\hat{z}_m \sqrt{\frac{1-\theta_m}{1-2\theta_m}} \leq \sqrt{\frac{1-\theta_m}{\theta_m}}, \quad (\text{B.12})$$

as desired. □

Henceforth, let A_1^* denote the optimal first dispatch area in the unconstrained problem, and define τ_1^* such that $\tau_1^* + cA_1^*\sqrt{\tau_1^*} = 1$. The next lemma proves the correctness of the algorithm when $m = 2$ and also serves as the base case for the inductive proof of the general result.

Lemma 27. *Let $m = 2$ and $B < A_1^*$. In an optimal solution to the B -constrained problem, it must hold that $B = A_1 \geq A_2$.*

Proof. Suppose we are given an optimal solution $((\tau_1, A_1), (\tau_2, A_2))$ such that $B > A_1$. We know that

$$\tau_1 + cA_1\sqrt{\tau_1} = 1 \quad (\text{B.13})$$

and

$$\tau_2 + cA_2\sqrt{\tau_2} = 1 - \tau_1. \quad (\text{B.14})$$

As a preliminary note, if we are to solve the unconstrained 1v1d problem on a truncated service day of length $1 - \tau_1$ by rescaling the service day to have unit length, then we must use the following parameters instead of λ , c_0 , and c :

$$\hat{\lambda} = \lambda(1 - \tau_1),$$

$$\hat{c}_0 = \frac{c_0}{1 - \tau_1},$$

$$\begin{aligned} \hat{c} &= \hat{c}_0\sqrt{\hat{\lambda}} \\ &= \frac{c}{\sqrt{1 - \tau_1}}. \end{aligned}$$

First, suppose that $\tau_1 > \frac{1}{3}$. Recall from the analysis of the unconstrained 1v1d problem that the total quantity is maximized when the accumulation time is $\tau = \frac{1}{3}$. Additionally, the derivative of the total quantity as a function of the accumulation time is negative for all $\tau \in (\frac{1}{3}, 1)$ in that problem. Therefore, we can decrease τ_1 by a sufficiently small amount (and increase A_1 by a corresponding amount such that (B.13) still holds) to increase the

quantity served by the first dispatch without decreasing the quantity served by the second dispatch. By contradiction, the solution $((\tau_1, A_1), (\tau_2, A_2))$ cannot be optimal, so it must hold that $\tau_1 \leq \frac{1}{3}$ in the optimal solution to the constrained problem.

Now, let us consider the case when $A_1 \leq A_2$. This implies

$$\frac{1 - \tau_1}{c\sqrt{\tau_1}} \leq \frac{2}{\hat{c}\sqrt{3}} \quad (\text{B.15})$$

or, equivalently,

$$\frac{1 - \tau_1}{c\sqrt{\tau_1}} \leq \frac{2\sqrt{1 - \tau_1}}{c\sqrt{3}}. \quad (\text{B.16})$$

Rearranging gives $\tau_1 \geq \frac{3}{7} > \frac{1}{3}$, a contradiction to our previous result. Thus, it must hold that $A_1 > A_2$.

Since $B > A_1 > A_2$, we can express the quantity served by each of the first and second dispatches in terms of τ_1 :

$$q_1(\tau_1) = \frac{\lambda}{c}(1 - \tau_1)\sqrt{\tau_1}, \quad (\text{B.17})$$

$$q_2(\tau_1) = \frac{\hat{\lambda}}{\hat{c}} \cdot \frac{2}{3\sqrt{3}} = \frac{2\lambda}{3c\sqrt{3}}(1 - \tau_1)^{3/2}. \quad (\text{B.18})$$

We then take the derivative of the total quantity with respect to τ_1 :

$$q'(\tau_1) = \frac{\lambda}{c} \left(\frac{1}{2}\tau_1^{-1/2} - \frac{3}{2}\tau_1^{1/2} - \frac{1}{\sqrt{3}}(1 - \tau_1)^{1/2} \right). \quad (\text{B.19})$$

It can be verified that this expression is negative for all τ_1 for which the corresponding $A_1 < A_1^*$ (i.e., for all $\tau_1 \in (\tau_1^*, \frac{1}{3}]$). Thus, we can decrease τ_1 by a sufficiently small quantity, increase A_1 accordingly, and re-optimize q_2 accordingly such that the total quantity served increases. This contradicts the optimality of the given solution. Therefore, any solution with $A_1 < B$ cannot be optimal. \square

The following two results prove the correctness of the algorithm when $m \geq 3$.

Lemma 28. *Let $m \geq 3$ and $B < A_1^*$. Then, any solution with $A_1 < B$ and $A_1 \leq A_2$ cannot be optimal for the B -constrained problem.*

Proof. We will prove this claim by contradiction. Suppose we are given a candidate optimal solution to the constrained $mv1d$ problem $((\tau_1, A_1), (\tau_2, A_2), \dots, (\tau_m, A_m))$ with $m \geq 3$, $B > A_1$, and $A_2 \geq A_1$. By our previous discussion, we know that $\tau_1 \in (\tau_1^*, \frac{1}{3}]$.

Let $t_2 = \tau_1 + \tau_2$. We will show that we can slightly simultaneously perturb τ_1 and τ_2 (while leaving their sum t_2 unchanged) such that the total quantity served increases. Specifically, we wish to perform the following operations: decrease τ_1 by some small $\varepsilon > 0$, increase τ_2 by the same ε , increase A_1 by some δ_1 such that (B.13) is maintained, and decrease A_2 by some δ_2 such that (B.14) is maintained.

It remains to be seen whether the rate of increase of q_2 outpaces the rate of decrease of q_1 when we perform the above operations. With a slight abuse of notation, let us represent q_1 and q_2 as functions of τ_1 under the assumption that t_2 is fixed:

$$q_1(\tau_1) = \lambda A_1 \tau_1 = \lambda \left(\frac{1 - \tau_1}{c\sqrt{\tau_1}} \right) \tau_1 = \frac{\lambda}{c} (1 - \tau_1) \sqrt{\tau_1}, \quad (\text{B.20})$$

$$q_2(\tau_1) = \lambda A_2 \tau_2 = \lambda A_2 (t_2 - \tau_1) = \lambda \left(\frac{1 - t_2}{c\sqrt{t_2 - \tau_1}} \right) (t_2 - \tau_1) = \frac{\lambda}{c} (1 - t_2) \sqrt{t_2 - \tau_1}. \quad (\text{B.21})$$

Omitting the constant factors $\frac{\lambda}{c}$, the derivatives of both quantities with respect to τ_1 are

$$q_1'(\tau_1) = \frac{1}{2} \tau_1^{-1/2} - \frac{3}{2} \tau_1^{1/2}, \quad (\text{B.22})$$

$$q_2'(\tau_1) = \frac{t_2 - 1}{2\sqrt{t_2 - \tau_1}}. \quad (\text{B.23})$$

To show that the perturbation procedure increases the total quantity served by the first and second dispatches, we must prove that the sum of these derivatives evaluated at τ_1 is negative, i.e., that

$$h(\tau_1) = \frac{3}{2} \tau_1^{1/2} - \frac{1}{2} \tau_1^{-1/2} + \frac{1 - t_2}{2\sqrt{t_2 - \tau_1}} > 0. \quad (\text{B.24})$$

Since $A_2 \geq A_1$, by (B.13) and (B.14), it must hold that $\tau_2 \leq \tau_1$. This implies $t_2 \leq 2\tau_1$, which further implies

$$\frac{1-t_2}{2\sqrt{t_2-\tau_1}} \geq \frac{1-2\tau_1}{2\sqrt{2\tau_1-\tau_1}}. \quad (\text{B.25})$$

Therefore,

$$\begin{aligned} h(\tau_1) &= \frac{3}{2}\tau_1^{1/2} - \frac{1}{2}\tau_1^{-1/2} + \frac{1-t_2}{2\sqrt{t_2-\tau_1}} \\ &\geq \frac{3}{2}\tau_1^{1/2} - \frac{1}{2}\tau_1^{-1/2} + \frac{1-2\tau_1}{2\sqrt{2\tau_1-\tau_1}} \\ &= \frac{3}{2}\tau_1^{1/2} - \frac{1}{2}\tau_1^{-1/2} + \frac{1}{2}\tau_1^{-1/2} - \tau_1^{1/2} \\ &= \frac{1}{2}\tau_1^{1/2} \\ &> 0. \end{aligned}$$

Thus, if $A_2 \geq A_1$, we can find sufficiently small $\varepsilon, \delta_1, \delta_2 > 0$ such that $((\tau_1 - \varepsilon, A_1 + \delta_1), (\tau_2 + \varepsilon, A_2 - \delta_2), \dots, (\tau_m, A_m))$ is an improved feasible solution. Hence, by contradiction, if $B < A_1^*$, then the optimal solution to the constrained problem must have either $A_1 = B$ or $A_1 > A_2$. \square

Applying induction implies that the B -constrained optimal solution must have $B \geq A_1 \geq A_2 \geq \dots \geq A_m$.

Lemma 29. *Let $m \geq 3$ and $B < A_1^*$. Then, in an optimal solution to the constrained m -vehicle problem, $A_1 = B$.*

Proof. We proceed by induction with the result in Lemma 27 as the base case. Assume that the claim is true for $m-1$ vehicles. For the purposes of contradiction, suppose we are given a candidate optimal solution to the constrained m -vehicle problem $((\tau_1, A_1), (\tau_2, A_2), \dots, (\tau_m, A_m))$ with $m \geq 3$ and $B > A_1$. Observe first that $\tau_1 > \tau_1^*$. By Lemma 28, we may assume that $A_1 > A_2$.

Because the full solution is assumed optimal, the final $m - 1$ are also optimized over the truncated service day induced by τ_1 . By the induction hypothesis, if the final $m - 1$ vehicles were optimized with respect to the constrained problem *but not* the unconstrained problem, it would hold that $A_2 = B$. However, because $A_2 < B$, we know that the the final $m - 1$ vehicles must be optimized with respect to the unconstrained $(m - 1)$ -vehicle problem as well. Therefore, by Theorem 13, $\tau_2 = \theta_{m-1}(1 - \tau_1)$, where θ_{m-1} is the optimal first dispatch time in the unconstrained $(m - 1)$ -vehicle problem. Consequently, $A_1 > A_2$ implies

$$\frac{1 - \tau_1}{c\sqrt{\tau_1}} > \frac{1 - \tau_1 - \theta_{m-1}(1 - \tau_1)}{c\sqrt{\theta_{m-1}(1 - \tau_1)}} \quad (\text{B.26})$$

which in turn implies

$$\tau_1 < \frac{1}{\frac{1}{\theta_{m-1}} - 1 + \theta_{m-1}} < \frac{\theta_{m-1}}{1 - \theta_{m-1}}. \quad (\text{B.27})$$

Let z_{m-1} represent the optimal total quantity served in the unconstrained $(m - 1)$ -vehicle problem, and let $\frac{\lambda}{c}\hat{z}_m = z_m$. Theorem 13 implies that the total quantity as a function of τ_1 is

$$q(\tau_1) = \frac{\lambda}{c}(1 - \tau_1)\sqrt{\tau_1} + \frac{\lambda}{c}\hat{z}_{m-1}(1 - \tau_1)^{3/2} \quad (\text{B.28})$$

when $A_1 > A_2$. Our goal is to show that $q'(\tau_1) < 0$ for all $\tau_1 \in (\tau_1^*, \frac{\theta_{m-1}}{1 - \theta_{m-1}})$ so that we can slightly reduce τ_1 (equivalently, slightly increase A_1) and improve the total quantity served. As such, we henceforth ignore the scaling factor $\frac{\lambda}{c}$.

Differentiating gives

$$q'(\tau_1) = -\frac{3}{2}\tau_1^{1/2} + \frac{1}{2}\tau_1^{-1/2} - \frac{3}{2}\hat{z}_{m-1}(1 - \tau_1)^{1/2}. \quad (\text{B.29})$$

Note that, by definition, $q'(\tau_1^*) = 0$. Therefore, it suffices to show that $q''(\tau_1) < 0$ for all $\tau_1 \in (0, \frac{\theta_{m-1}}{1 - \theta_{m-1}}]$.

Differentiating twice gives

$$q''(\tau_1) = -\frac{3}{4}\tau_1^{-1/2} - \frac{1}{4}\tau_1^{-3/2} + \frac{3}{4}\hat{z}_{m-1}(1-\tau_1)^{-1/2}, \quad (\text{B.30})$$

and differentiating thrice gives

$$q'''(\tau_1) = \frac{3}{8}\tau_1^{-3/2} + \frac{3}{8}\tau_1^{-5/2} + \frac{3}{8}\hat{z}_{m-1}(1-\tau_1)^{-3/2} > 0. \quad (\text{B.31})$$

Observe that $\lim_{\tau_1 \downarrow 0} q''(\tau_1) = -\infty$. Additionally,

$$\begin{aligned} q''\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right) &= -\frac{3}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-1/2} - \frac{1}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-3/2} + \frac{3}{4}\hat{z}_{m-1}\left(1 - \frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-1/2} \\ &= -\frac{3}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-1/2} - \frac{1}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-3/2} + \frac{3}{4}\hat{z}_{m-1}\left(\frac{1-2\theta_{m-1}}{1-\theta_{m-1}}\right)^{-1/2} \\ &= \frac{3}{4}\left(\hat{z}_{m-1}\sqrt{\frac{1-\theta_{m-1}}{1-2\theta_{m-1}}} - \sqrt{\frac{1-\theta_{m-1}}{\theta_{m-1}}}\right) - \frac{1}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-3/2} \\ &\leq -\frac{1}{4}\left(\frac{\theta_{m-1}}{1-\theta_{m-1}}\right)^{-3/2} \\ &< 0, \end{aligned}$$

where the penultimate step is implied by Lemma 26. It follows that $q''(\tau_1) < 0$ for all $\tau_1 \in (0, \frac{\theta_{m-1}}{1-\theta_{m-1}}]$. Because $q'(\tau_1^*) = 0$, it follows that $q'(\tau_1) < 0$ for all $\tau_1 \in (\tau_1^*, \frac{\theta_{m-1}}{1-\theta_{m-1}}]$. Thus, we can decrease τ_1 by a sufficiently small quantity, increase A_1 accordingly, and re-optimize the remaining dispatches accordingly such that the total quantity served increases. This contradicts the optimality of the given solution. Therefore, any solution with $A_1 < B$ cannot be optimal. \square

B.10 Proof of Property 19

Given a set of positive accumulation times, $\{\tau_1, \tau_2, \dots, \tau_D\}$, consider the d -th dispatch, where $d < D$. Inequality (3.7c) constrains the service area by $A_d \leq \frac{\tau_{d+1}}{c\sqrt{\tau_d}}$. As the objective value (3.7a) increases linearly with A_d , choosing the service area such that this inequality holds at equality will maximize the quantity served on the d -th dispatch for all $d < D$. Now, consider the last dispatch. Inequality (3.7b) constrains the service area by: $A_D \leq \frac{1 - \sum_{\delta=1}^D \tau_\delta}{c\sqrt{\tau_D}}$. As the objective value (3.7a) scales linearly with A_D , we choose this service area such that this inequality holds at equality in order to serve the maximal number of orders served by the D -th dispatch, which completes the proof. \square

B.11 Proof of Property 20

For a set of accumulation times to be a part of an optimal dispatching solution to the 1vDd model formulated by (3.8a)-(3.8c), by first order conditions it must be true that both

$$\tau_1^* = \frac{(\tau_2^*)^2}{4\tau_D^*}$$

and

$$\tau_d^* = \frac{(\tau_{d+1}^*)^2}{4(\sqrt{\tau_D^*} - \sqrt{\tau_{d-1}^*})^2}$$

for all $d \in \{2, \dots, D-1\}$. By Property 19, these equations imply that $A_1^* = \frac{2}{c}\sqrt{\tau_D^*}$, and

$$A_d^* = \frac{2}{c} \left(\sqrt{\tau_D^*} - \sqrt{\tau_{d-1}^*} \right) \leq \frac{2}{c} \sqrt{\tau_D^*}$$

for all $d \in \{2, \dots, D-1\}$. \square

B.12 Proof of Theorem 21

By Property 20, we have that $A_d^* \leq \frac{2}{c}\sqrt{\tau_D^*}$ for all $d < D$. Thus, we can consider a relaxation of the original problem defined by (3.7a)-(3.7e) where each of the first $D - 1$ dispatches serves an area of $A_d = \frac{2}{c}\sqrt{\tau_D}$ without any regard for returning to the depot in time for the next dispatch. That is, consider the relaxation:

$$\max \quad \lambda A_D \tau_D + \sum_{d=1}^{D-1} \lambda \frac{2\sqrt{\tau_D}}{c} \tau_d \quad (\text{B.32a})$$

$$\text{s.t.} \quad \sum_{\delta=1}^D \tau_\delta + c A_D \sqrt{\tau_D} \leq 1, \quad (\text{B.32b})$$

$$A_D \geq 0, \quad (\text{B.32c})$$

$$\tau_d \geq 0. \quad \forall d \quad (\text{B.32d})$$

In this relaxed system, it is a strictly dominant strategy for the final dispatch to serve an area large enough such that the vehicle will arrive back to the depot exactly at the end of the service day, implying that $A_D = \frac{1 - \sum_{\delta=1}^D \tau_\delta}{c\sqrt{\tau_D}}$. Without loss of optimality, as each of the first $D - 1$ dispatches serve the same area, and themselves have no explicit concerns of arriving back before a future dispatch, everything can be served on the first dispatch while the remaining $D - 2$ dispatches serve nothing. Thus, this relaxed problem can be re-formulated as:

$$\max \quad \lambda \left(\frac{1 - \tau_1 - \tau_D}{c\sqrt{\tau_D}} \right) \tau_D + \lambda \left(\frac{2\sqrt{\tau_D}}{c} \right) \tau_1 \quad (\text{B.33a})$$

$$\text{s.t.} \quad \tau_1 + \tau_D \leq 1, \quad (\text{B.33b})$$

$$\tau_1 \geq 0, \quad (\text{B.33c})$$

$$\tau_D \geq 0. \quad (\text{B.33d})$$

Since this objective function is equivalent to $\frac{\lambda}{c} (1 + \tau_1 - \tau_D) \sqrt{\tau_D}$, we see that $\tau_1 +$

$\tau_D \leq 1$ will hold at equality for an optimal dispatching policy, so after a substitution of $\tau_1 = 1 - \tau_D$ we arrive at the problem:

$$\max_{\tau_D} \frac{2\lambda}{c}(1 - \tau_D)\sqrt{\tau_D} \tag{B.34a}$$

$$\text{s.t. } \tau_D \in [0, 1]. \tag{B.34b}$$

This problem is identical to the optimization problem presented in Section 3.3 for the 1v1d model with the caveat that the objective value is exactly twice as large. Therefore, this relaxation of the 1vDd model has an optimal objective value of $\frac{\lambda}{c} \frac{4}{3\sqrt{3}}$, as desired. \square

APPENDIX C

TECHNICAL SUPPLEMENTATION FOR CHAPTER 4

C.1 Proof of Theorem 23

Observe the following set of inequalities regarding the cost of some arbitrary route:

$$\begin{aligned} c_r &= \sum_{i=1}^{|r|} q_{r_i} \left(\frac{\ell_r}{2m_r} + \sum_{j=0}^{i-1} (c_{(r_j, r_{j+1})} + s_1) \right) \\ &\geq \sum_{i=1}^{|r|} q_{r_i} \left(\sum_{j=0}^{i-1} (c_{(r_j, r_{j+1})} + s_1) \right) \\ &\geq \sum_{i=1}^{|r|} (q_{r_i} (c_{(0, r_i)} + s_1)). \end{aligned}$$

From here it is clear that any subset of routes $R \subseteq \mathcal{R}^+$ which partitions the set of demand nodes has cost given by $\sum_{r \in R} c_r \geq \sum_{r \in R} \sum_{i=1}^{|r|} (q_{r_i} (c_{(0, r_i)} + s_1)) = \sum_{i=1}^n (q_i (c_{(0, i)} + s_1))$. Thus any set of routes which feasibly satisfies (4.4a)-(4.4e) will return an objective value bounded below by $\sum_{i=1}^n (q_i (c_{(0, i)} + s_1))$.

C.2 Proof of Theorem 24

Consider the marginal reduction in cycle time when adding one additional vehicle to a route:

$$\begin{aligned}
c_{r(k)} - c_{r(k+1)} &= \sum_{i=1}^{|r|} q_{r_i} \left(\frac{\ell_r}{2k} + \sum_{j=0}^{i-1} \left(c_{(r_j, r_{j+1})} + s_1 \right) \right) \\
&\quad - \sum_{i=1}^{|r|} q_{r_i} \left(\frac{\ell_r}{2(k+1)} + \sum_{j=0}^{i-1} \left(c_{(r_j, r_{j+1})} + s_1 \right) \right) \\
&= \sum_{i=1}^{|r|} q_{r_i} \frac{\ell_r}{2k} - \sum_{i=1}^{|r|} q_{r_i} \frac{\ell_r}{2(k+1)} \\
&= \left(\frac{q_r \ell_r}{2} \right) \left(\frac{1}{k} - \frac{1}{k+1} \right) \\
&= \left(\frac{q_r \ell_r}{2} \right) \left(\frac{1}{k^2 + k} \right)
\end{aligned}$$

As the factor $\frac{1}{k^2+k}$ strictly decreases as k increases for all $k \geq 1$, we are done.

C.3 Details regarding the example illustrated in Figure 4.1

The three demand nodes of A, B, and C are located at the coordinates $(-8, 6)$, $(0, 21)$, and $(8, 6)$ respectively. The depot node is located at the origin, $(0, 0)$. The cost-metric between any two nodes on the graph are given as euclidean distances and no fixed setup or service times are considered. By symmetry, the permutations of $C \rightarrow B \rightarrow A$, $C \rightarrow A \rightarrow B$, and $B \rightarrow A \rightarrow C$ are equivalent to those given by $A \rightarrow B \rightarrow C$, $A \rightarrow C \rightarrow B$, and $B \rightarrow C \rightarrow A$ respectively (and are thus ignored).

The table below describes the minimal fleet size necessary to serve demand as well as the cost of the route given a fleet size (for up to 3 vehicles). This data assumes that package demands are given by $q_A = q_C = 1$ and $q_B = 10$ and service vehicles have a capacity of $Q = 350$.

Table C.1: Supporting information for Figure 4.1.

	m_r^{min}	$c_r m_r = 1$	$c_r m_r = 2$	$c_r m_r = 3$
$A \rightarrow B \rightarrow C$	1	648	486	432
$A \rightarrow C \rightarrow B$	2	∞	658	594
$B \rightarrow C \rightarrow A$	2	∞	494	430

C.4 Details regarding the generated inputs in Section 4.5

The generated demand points, labeled 1 through 10, were located at $(-5, 2)$, $(-2, -9)$, $(1, -4)$, $(-4, 6)$, $(0, 2)$, $(1, -10)$, $(2, -4)$, $(-9, -1)$, $(6, 4)$, and $(4, -7)$, respectively. As a reminder the depot location is given at the origin, $(0, 0)$. These demand nodes have a demands of 2, 3, 5, 1, 1, 2, 5, 1, 3, and 5 packages per hour respectively.

REFERENCES

- [1] United States Census Bureau, *Annual retail trade survey: Sales estimates (1992-2016)*, URL <https://www.census.gov/data/tables/2016/econ/arts/annual-report.html>. [Online; accessed 18-November-2018], 2018.
- [2] P. Borkowski, M. Jądzewska-Gutta, and A. Szmelter-Jarosz, “Lockdowned: Everyday mobility changes in response to COVID-19,” *Journal of Transport Geography*, vol. 90, p. 102 906, 2021.
- [3] J. De Vos, “The effect of COVID-19 and subsequent social distancing on travel behavior,” *Transportation Research Interdisciplinary Perspectives*, vol. 5, p. 100 121, 2020.
- [4] United States Census Bureau, *Quarterly retail e-commerce sales 1st quarter 2020*, URL <https://www2.census.gov/retail/releases/historical/ecommm/20q1.pdf>, 2020.
- [5] ———, *Quarterly retail e-commerce sales 1st quarter 2021*, URL <https://www2.census.gov/retail/releases/historical/ecommm/21q1.pdf>, 2021.
- [6] A. Inc., *Amazon launches same day delivery in seven major cities and expands saturday delivery options [press release]*, URL <https://press.aboutamazon.com/news-releases/news-release-details/amazon-launches-same-day-delivery-seven-major-cities-and-expand>. [Online; accessed 18-November-2018], 2009.
- [7] ———, *Amazon expands free shipping to everyone for the holiday season [press release]*, URL <https://press.aboutamazon.com/news-releases/news-release-details/amazon-expands-free-shipping-everyone-holiday-season>. [Online; accessed 18-November-2018], 2018.
- [8] C. Ratcliff, *15 fascinating insights from econsultancy’s 2014 reports*, URL <https://econsultancy.com/15-fascinating-insights-from-econsultancy-s-2014-reports/>. [Online; accessed 18-February-2021], 2014.
- [9] F. Reichheld and P. Schefter, *The economics of e-loyalty*, URL <https://hbswk.hbs.edu/archive/the-economics-of-e-loyalty>. [Online; accessed 18-February-2021], 2000.
- [10] M. Klapp, “Dynamic optimization for same-day delivery operations,” Ph.D. dissertation, Georgia Institute of Technology, 2016.
- [11] M. Klapp, A. Erera, and A. Toriello, “The dynamic dispatch waves problem for same-day delivery,” *European Journal of Operations Research*, vol. 271, no. 2, pp. 519–534, 2018.

- [12] ———, “The one-dimensional dynamic dispatch waves problem,” *Transportation Science*, vol. 52, no. 2, pp. 402–415, 2018.
- [13] A. Nabila, M. Gendreau, and J. Potvin, “A dynamic vehicle routing problem with multiple delivery routes,” *Annals of Operations Research*, vol. 199, pp. 103–112, 2012.
- [14] W. Van Heeswijk, M. Mes, and J. Schutten, “The delivery dispatching problem with time windows for urban consolidation centers,” *Transportation Science*, vol. 53, no. 1, pp. 203–221, 2017.
- [15] M. Ulmer, B. Thomas, and D. Mattfeld, “Preemptive depot returns for a dynamic same-day delivery problem,” *EURO Journal on Transportation and Logistics*, vol. 8, pp. 327–361, 4 2018.
- [16] S. Voccia, A. Campbell, and B. Thomas, “The same-day delivery problem for online purchases,” *Transportation Science*, vol. 53, no. 1, pp. 167–184, 2017.
- [17] B. Brunk, J. Neville, K. Morris, D. Naumann, W. Deacon, and K. Fischer, “Digital commerce survey: Annual survey 2017,” BRP Consulting, Tech. Rep., 2017, URL <https://brpconsulting.com/download/2017-digital-commerce-survey/>. [Online; accessed 18-November-2018].
- [18] A. Inc., *Prime free same-day delivery expands to 11 new metro areas [press release]*, URL <https://press.aboutamazon.com/news-releases/news-release-details/prime-free-same-day-delivery-expands-11-new-metro-areas>. [Online; accessed 18-November-2018], 2016.
- [19] S. Liu, L. He, and Z. Shen, “On-time last mile delivery: Order assignment with travel time predictors,” *Management Science*, forthcoming. URL <https://dx.doi.org/10.2139/ssrn.3179994>, 2018.
- [20] F. Corp., *Same-day delivery*, URL <https://www.fedex.com/en-us/shipping/same-day.html>. [Online; accessed 19-February-2019], 2019.
- [21] A. Inc., *Order with prime free same-day delivery*, URL <https://www.amazon.com/gp/help/customer/display.html?nodeId=201631620>. [Online; accessed 19-February-2019], 2019.
- [22] D. Reyes, A. Erera, and M. Savelsbergh, “Complexity of routing problems with release dates and deadlines,” *European Journal of Operational Research*, vol. 266, no. 1, pp. 29–34, 2018.
- [23] A. Franceschetti, O. Jabali, and G. Laporte, “Continuous approximation models in freight distribution management,” *TOP*, vol. 25, pp. 413–433, 2017.

- [24] J. Beardwood, J. Halton, and J. Hammersley, “The shortest path through many points,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 55, no. 4, pp. 299–327, 1959.
- [25] C. Daganzo, “The length of tours in zones of different shapes,” *Transportation Research Part B: Methodological*, vol. 18, no. 2, pp. 135–145, 1984.
- [26] G. Newell and C. Daganzo, “Design of multiple-vehicle delivery tours I: A ring-radial network,” *Transportation Research Part B*, vol. 20, no. 5, pp. 345–363, 1986.
- [27] D. Johnson, L. McGeoch, and E. Rothberg, “Asymptotic experimental analysis for the held-karp traveling salesman bound,” *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 341–350, 1996.
- [28] E. Çeven and K. Gue, “Optimal wave release times for order fulfillment systems with deadlines,” *Transportation Science*, vol. 51, no. 1, pp. 52–66, 2017.
- [29] M. Albareda-Sambola, E. Fernández, and G. Laporte, “The dynamic multiperiod vehicle routing problem with probabilistic information,” *Computers & Operations Research*, vol. 48, pp. 31–39, 2014.
- [30] A. Campbell and B. Thomas, “Probabilistic traveling salesman problem with deadlines,” *Transportation Science*, vol. 42, no. 1, pp. 1–21, 2008.
- [31] S. Voccia, A. Campbell, and B. Thomas, “The probabilistic traveling salesman problem with time windows,” *EURO Journal on Transportation and Logistics*, vol. 2(1-2), pp. 89–107, 2013.
- [32] R. Bent and P. Van Hentenryck, “Scenario-based planning for partially dynamic vehicle routing with stochastic customers,” *Operations Research*, vol. 52, no. 6, pp. 977–987, 2004.
- [33] M. Ulmer, “Delivery deadlines in same-day delivery,” *Logistics Research*, vol. 10, no. 3, pp. 1–15, 2017.
- [34] ———, “Dynamic pricing and routing for same-day delivery,” *Transportation Science*, vol. 54, pp. 1016–1033, 4 2020.
- [35] M. Ulmer and B. Thomas, “Same-day delivery with a heterogeneous fleet of drones and vehicles,” *Networks*, vol. 72, no. 4, pp. 475–505, 2018.
- [36] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *The traveling salesman problem: a computational study*. Princeton University Press, 2011, ISBN: 9781400841103.

- [37] H. Wang and A. Odoni, “Approximating the performance of a “last mile” transportation system,” *Transportation Science*, vol. 50, no. 2, pp. 659–675, 2014.
- [38] M. Figliozzi, “Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications,” *Transportation Research Part B*, vol. 41, no. 9, pp. 1014–1032, 2007.
- [39] J. Carlsson and S. Song, “Coordinated logistics with a truck and a drone,” *Management Science*, vol. 64, no. 9, pp. 4050–4069, 2017.
- [40] J. Carlsson, “Dividing a territory among several vehicles,” *INFORMS Journal on Computing*, vol. 24, no. 4, pp. 565–577, 2011.
- [41] M. Lim, H. Mak, and Z. Shen, “Agility and proximity considerations in supply chain design,” *Management Science*, vol. 63, no. 4, pp. 1026–1041, 2016.
- [42] United States Census Bureau, *Census explorer*, URL <https://www.census.gov/censusexplorer/censusexplorer.html>. [Online; accessed 06-November-2019].
- [43] G. M. Platform, *Distance matrix api*, URL <https://developers.google.com/maps/documentation/distance-matrix/intro>. [Online; accessed 06-November-2019].
- [44] J. Bursztunsky, *Some amazon customers will now be able to get orders delivered in five hours or less*, URL <https://www.cnbc.com/2020/03/03/amazon-expands-same-day-delivery-after-building-fulfillment-centers.html>. [Online; accessed 15-December-2020], 2020.
- [45] ———, *Walmart and instacart partner for same-day u.s. delivery in fight against amazon’s whole foods*, URL <https://www.cnbc.com/2020/08/11/walmart-and-instacart-partner-in-fight-against-amazons-whole-foods.html>. [Online; accessed 15-December-2020], 2020.
- [46] Instacart, *Introducing michaels to the instacart marketplace*, URL <https://news.instacart.com/introducing-michaels-to-the-instacart-marketplace-e95bd66e48ce>, 2021.
- [47] T. Walk-Morris, *Sephora enlists instacart for same-day delivery*, URL <https://www.retaildive.com/news/sephora-enlists-instacart-for-same-day-delivery/585997/>. [Online; accessed 15-December-2020], 2020.
- [48] S. Dolan, *The challenges of last mile delivery logistics & the technology solutions cutting costs*, URL <https://www.businessinsider.com/last-mile-delivery-shipping-explained>. [Online; accessed 15-December-2020], 2018.

- [49] M. Joerss, J. Schroder, F. Neuhaus, C. Klink, and F. Mann, “Parcel delivery: The future of last mile,” McKinsey & Company, Tech. Rep., 2016, URL https://www.mckinsey.com/~media/mckinsey/industries/traveltransportandlogistics/ourinsights/howcustomerdemandsarereshapingleastmiledelivery/parcel_delivery_the_future_of_last_mile.ashx. [Online; accessed 14-November-2020].
- [50] D. Ingold and S. Soper, *Amazon doesn't consider the case of its customers. should it?* URL <https://www.bloomberg.com/graphics/2016-amazon-same-day/>. [Online; accessed 16-January-2021], 2016.
- [51] S. Soper, *Amazon to bring same-day delivery to bronx, chicago after outcry*, URL <https://www.bloomberg.com/news/articles/2016-05-01/amazon-pledges-to-bring-same-day-delivery-to-bronx-after-outcry>. [Online; accessed 16-May-2021], 2016.
- [52] ———, *Amazon to bring same-day delivery to roxbury after outcry*, URL <https://www.bloomberg.com/news/articles/2016-04-26/amazon-to-bring-same-day-delivery-to-roxbury-after-outcry>. [Online; accessed 16-May-2021], 2016.
- [53] A. Stroh, A. Erera, and A. Toriello, “Tactical design of same-day delivery systems,” *Management Science*, forthcoming. URL http://www.optimization-online.org/DB_HTML/2019/06/7243.html, 2021.
- [54] D. Banerjee, A. Erera, and A. Toriello, “Fleet sizing and service region partitioning for same-day delivery systems,” In review, URL http://www.optimization-online.org/DB_HTML/2020/12/8183.html. [Online; accessed 20-January-2021], 2020.
- [55] J. Côté, T. de Queiroz, F. Gallesi, and M. Iori, “Optimization methods for the same-day delivery problem,” in *Advances in optimization and decision science for society, services and enterprises*, M. Paolucci, A. Sciomachen, and P. Uberti, Eds., Springer, 2019, pp. 335–349.
- [56] M. Klapp, A. Erera, and A. Toriello, “Request acceptance in same-day delivery,” *Transportation Research Part E*, vol. 143, p. 102083, 2020.
- [57] M. Ulmer and S. Streng, “Same-day delivery with pickup stations and autonomous vehicles,” *Computers & Operations Research*, vol. 108, pp. 1–19, 2019.
- [58] I. Dayarian, M. Savelsbergh, and J. Clarke, “Same-day delivery with drone resupply,” *Transportation Science*, vol. 54, no. 1, pp. 229–249, 2020.
- [59] B. Yao, C. McLean, and H. Yang, “Robust optimization of dynamic route planning in same-day delivery networks with one-time observation of new demand,” *Networks*, vol. 73, no. 4, pp. 434–452, 2019.

- [60] J. Oyola, H. Arntzen, and D. Woodruff, “The stochastic vehicle routing problem, a literature review, part II: Solution methods,” *EURO Journal on Transportation and Logistics*, vol. 6, no. 4, pp. 349–388, 2017.
- [61] ———, “The stochastic vehicle routing problem, a literature review, part I: Models,” *EURO Journal on Transportation and Logistics*, vol. 7, pp. 193–221, 2018.
- [62] V. Pillac, M. Gendreau, C. Guéret, and A. Medaglia, “A review of dynamic vehicle routing problems,” *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013.
- [63] H. Psaraftis, M. Wen, and C. Kontovas, “Dynamic vehicle routing problems: Three decades and counting,” *Networks*, vol. 67, no. 1, pp. 3–31, 2016.
- [64] A. Snoeck and M. Winkenbach, “A discrete simulation-based optimization algorithm for the design of highly responsive last-mile distribution networks,” Massachusetts Institute of Technology working paper, URL <https://hdl.handle.net/1721.1/125591>, 2020.
- [65] X. Chen, T. Wang, B. Thomas, and M. Ulmer, “Same-day delivery with fairness,” working paper, URL <https://arxiv.org/abs/2007.09541>, 2020.
- [66] X. Chen, M. Ulmer, and B. Thomas, “Deep q-learning for same-day delivery with vehicles and drones,” working paper, URL <https://arxiv.org/abs/1910.11901>, 2019.
- [67] C. Daganzo, “The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application,” *Transportation Science*, vol. 18, no. 4, pp. 331–350, 1984.
- [68] G. Newell and C. Daganzo, “Design of multiple vehicle delivery tours - II other metrics,” *Transportation Research Part B*, vol. 20, no. 5, pp. 365–376, 1986.
- [69] D. Merchán and M. Winkenbach, “An empirical validation and data-driven extension of continuum approximation approaches for urban route distances,” *Networks*, vol. 73, no. 4, pp. 418–433, 2019.
- [70] S. Ansari, M. Başdere, X. Li, Y. Ouyang, and K. Smilowitz, “Advancements in continuous approximation models for logistics and transportation systems: 1996 – 2016,” *Transportation Research Part B*, vol. 107, pp. 229–252, 2018.
- [71] United States Census Bureau, *Patterns of metropolitan and micropolitan population change: 2000 to 2010*, URL <https://www.census.gov/library/visualizations/2012/dec/c2010sr-01-density.html>. [Online; accessed 19-January-2021], 2012.

- [72] K. Moore, *What is customer retention + 16 proven retention strategies*, URL <https://www.helpscout.com/blog/customer-retention-strategies-that-work/>. [Online; accessed 18-February-2021], 2020.
- [73] I. Kara, B. Kara, and M. Yetis, “Energy minimizing vehicle routing problem,” *Combinatorial Optimization and Applications. COCOA 2007. Lecture Notes in Computer Science*, vol. 4616, pp. 62–71, 2007.
- [74] M. Desrochers, J. Desrosiers, and M. Solomon, “A new optimization algorithm for the vehicle routing problem with time windows,” *Operations Research*, vol. 40, no. 2, pp. 342–354, 1992.
- [75] R. Fukasawa, Q. He, and Y. Song, “A branch-cut-and-price algorithm for the energy minimization vehicle routing problem,” *Transportation Science*, vol. 50, no. 1, pp. 23–34, 2016.
- [76] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998.
- [77] M. Fischetti, G. Laporte, and S. Martello, “The delivery man problem and cumulative matroids,” *Operations Research*, vol. 41, no. 6, pp. 1055–1064, 1993.
- [78] F. Afrati, S. Cosmadakis, C. Papadimitriou, G. Papageorgiou, and N. Papakostantinou, “The complexity of the travelling repairman problem,” *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, vol. 20, no. 1, pp. 79–87, 1986.
- [79] I. Mendez-Diaz, P. Zabala, and A. Lucena, “A new formulation for the traveling deliveryman problem,” *Discrete Applied Mathematics*, vol. 156, no. 17, pp. 3223–3237, 2008.
- [80] N. Mladenovic, D. Urosevic, and S. Hanafi, “Variable neighborhood search for the travelling deliveryman problem,” *4OR*, vol. 11, pp. 57–73, 2013.
- [81] G. Onder, I. Kara, and T. Derya, “New integer programming formulation for multiple traveling repairmen problem,” *Transportation Research Procedia*, vol. 22, pp. 355–361, 2017.
- [82] F. Angel-Bello, Y. Cardona-Valdes, and A. Alvarez, “Mixed integer formulations for the multiple minimum latency problem,” *Operational Research*, vol. 19, pp. 369–398, 2019.

- [83] A. Salehipour and M. Sepehri, “Exact and heuristic solutions to minimize total waiting time in the blood products distribution problem,” *Advances in Operations Research*, vol. 2012, p. 393 890, 2012.
- [84] S. Wang, M. Liu, and F. Chu, “Approximate and exact algorithms for an energy minimization traveling salesman problem,” *Journal of Cleaner Production*, vol. 249, p. 119 433, 2020.
- [85] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Graduate School of Industrial Administration, Carnegie Mellon University, Technical Report 388, 1976.
- [86] P. Gilmore and R. Gomory, “A linear programming approach to the cutting-stock problem,” *Operations Research*, vol. 9, no. 6, pp. 849–859, 1961.
- [87] L. Coelho, J. Cordeau, and G. Laporte, “Thirty years of inventory routing,” *Transportation Science*, vol. 48, no. 1, pp. 1–19, 2014.
- [88] A. Campbell and M. Savelsbergh, “Delivery volume optimization,” *Transportation Science*, vol. 38, no. 2, pp. 210–223, 2004.
- [89] C. Archetti and L. Bertazzi, “Recent challenges in routing and inventory routing: E-commerce and last-mile delivery,” *Networks*, vol. 77, no. 2, pp. 255–268, 2021.
- [90] P. Francis, K. Smilowitz, and M. Tzur, “The period vehicle routing problem and its extensions,” in *The vehicle routing problem: latest advances and new challenges*. Springer US, 2008, pp. 73–102.
- [91] ———, “The period vehicle routing problem with service choice,” *Transportation Science*, vol. 40, no. 4, pp. 439–454, 2006.

VITA

Alexander Martin Stroh was born in Hutchinson, Minnesota in 1994 to Patrick and Debra Stroh. A lifelong student, in 2012 he enrolled in the College of Engineering at the University of Minnesota - Twin Cities. There he received his Bachelors of Industrial and Systems Engineering as well as his Bachelors of Science in Mathematics. In 2016, Alex moved to Atlanta to pursue doctoral studies in Operations Research at the Georgia Institute of Technology. During his studies, he achieved a Master of Science in Statistics. His research there focused on the design of last mile logistical systems, especially as they pertain to tactical level decision making. His broader research interests include transportation logistics, optimization theory, data analytics, and healthcare decision making. Teaching has always been a significant part of Alex's academic journey. He started out as a tutor and an undergraduate teaching assistant while at the University of Minnesota, which eventually lead to becoming a graduate teaching assistant and full instructor while at the Georgia Institute of Technology. On a personal level, Alex is fortunate to have a supportive inner circle of lifelong friends including his wife Bridget who he happily married in 2021.