Exact and Heuristic Approaches to Middle and Last Mile Logistics

Larissa Prates Guimarães Petroianu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Zelda B. Zabinsky, Chair

Mauricio G. C. Resende, Chair

Archis Ghate

James F. Cowan

Program Authorized to Offer Degree:
Industrial and Systems Engineering

University of Washington

Abstract

Exact and Heuristic Approaches to Middle and Last Mile Logistics

Larissa Prates Guimarães Petroianu

Co-Chairs of the Supervisory Committee:
Professor Zelda B. Zabinsky
Industrial and Systems Engineering

Affiliate Professor Mauricio G. C. Resende
Industrial and Systems Engineering

Logistics is a well-studied field in operations research. Numerous authors have done extensive work in this area, especially in the domain of routing problems. However there are still aspects of routing to be better explored. For example, routing considering different types of uncertainties, penalties, or multiple objectives in middle and last mile delivery. The main focus of this research is to develop an optimization methodology that improves middle and last mile distribution, maximizing the accessibility of goods to the final customer. This research considers the impact of uncertainty as well as multiple objectives to inform decision makers during route planning. It develops optimization models and methods that solve middle and last mile distribution while considering uncertainty and multiple objectives, thereby, providing a strategic plan that is well-positioned to improve the accessibility of goods to the final customer. Finally, this research is motivated by two realistic applications, vaccine distribution in Africa and improvement of planning for delivery routes and product distribution for commercial companies. This research adapts and generalizes a Vehicle Routing and Scheduling Algorithm (VeRSA) that embeds a heuristic in an exact method to address the need for fast and efficient computation while maintaining information on an optimality gap.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I would like to express my gratitude to Professor Zelda B. Zabinsky and Mauricio G.C. Resende for all their support during my Ph.D. and for giving me the chance to be their mentee. I will take their teaching and guidance in my professional and personal life forever. I would also like to thank my committee members, Professors Archis Ghate, Kenneth Sherr, and James F. Cowan for the helpful comments and insights, and the willingness in the final defense. I would also like to extend my gratitute to Professor Christina Mastrangelo for her teaching and research opportunities. My research was funded by the Industrial and Systems Engineering department at the University of Washington, for which I just have to thank all the incredible team, especially Deborah, Jennifer, Neelu, and Sheila. I also had financial support from the Brazilian National Council for Scientific and Technological Development (CNPq), VillageReach, Philips, and Amazon. I am proud to represent each of these institutions in my research. I would also like to acknowledge my colleagues for their friendship, companionship, and support during these five years. In special Aven, Aziz, Michelle, Ting-Yu, Tolu, and Victoria. I extend my thanks to Yi Chu for his collaboration in part of this research. I made wonderful friends in Seattle, who helped me feel a little more at home and brought “brasilidade” and joy during these years. Thanks to the Brazilian community that we created around the PUB-Seattle (Brazilian group of researchers and scholars in Seattle), and to the FICA capoeira group. Particularly David, Eduardo, Flávia, Juliana, Nigini, and Mestre Silvinho. My special gratitude is to my parents, Rita and Andy, for all their support on this journey. They always believed in me, even when they did not understand what I was doing. Thanks for showing your love, and for enduring my frequent calls. I also want to thank my beautiful family, uncles, cousins, and grandparents for the

great care and support. And I cannot forget my friends back in Brazil and around the world who are always there for me. These last five years had many challenges, growth, and overcoming. Everyone in my live helped me reach this moment. It was not an easy time, but every sweat, tear, victory, smile, and laugh made it worthwhile and prepared me for the next challenges that life will present me!

DEDICATION

To my parents and grandparents, who always encouraged me and helped me to become a better person.

Chapter 1

INTRODUCTION

In supply chain, logistics and distribution are well studied fields. Routing problems have long been the focus of researchers (Gendreau et al., 1996; Laporte and Louveaux, 2002), resulting in algorithms such as the simplex method for minimum cost flow (Orlin, 1997), Dijkstra's algorithm for shortest path (Dijkstra, 1959), and 2-opt and 3-opt local search for the traveling salesman problem (Mersmann et al., 2012). However, most studies in the field of routing concentrate on a deterministic version of the problem. Of those that address uncertainty, most only consider one type, limiting themselves to study uncertainty in transit time or demand (Ritzinger et al., 2016). In addition to uncertainty, it is also important to consider multiple objectives, whereas most papers are limited to a single objective, such as minimization of transit time or cost, ignoring, for example, the effectiveness of the delivery. By considering uncertainty and multiple objectives, planning the routes and vehicles can incorporate risk and contingency planning, thus affecting the guarantee of a successful delivery.

This research considers the impact of uncertainty as well as multiple objectives to inform decision makers during route planning.

It is also important to understand the various stages of the distribution process, including first mile, middle mile, and last mile. The former considers transportation of products from the supplier to the fulfillment or distribution centers. Middle mile transportation consists of moving goods between facilities (e.g., with aircraft, rail, and/or trucks) while satisfying several constraints, including time windows for pickup and delivery. Last mile transportation is delivering the product to the final client (e.g., with trucks, vans, cars, and/or bicycles). Desrosiers et al. (1995) is an extensive survey on this subject. Often optimization models are separated into parts (e.g., middle mile and last mile) transportation problems (Lin et al., 2010; Zhou et al., 2018) due to their different constraints and objectives. It is important to recognize that the different mile-problems have dependencies. For example, cost-effective solutions to the middle mile problem may use as few vehicles as possible and centralize the intermediate storage, which may make the last mile problem costly or even infeasible. Uncertainty also impacts both problems. For example, weather causing delay in aircraft moving goods between fulfillment centers will impact the last mile effectiveness. Thus, it is important to evaluate and develop middle and last mile solutions, since the result of middle mile routing affects the inputs to the last mile, and, on the other hand, the demand of last mile is important to establish the needs of the middle mile routing.

This research develops optimization models and methods that integrate middle and last mile distribution while considering uncertainty and multiple objectives, thereby, providing a strategic plan that is well-positioned to improve the accessibility of goods to the final customer.

This research creates optimization models and methods that are motivated by two realistic applications. The first is vaccine distribution in Africa. Vaccines should reach the entire population-in-need by way of an efficient and low-cost distribution. Vaccines have an additional consideration of maintaining a proper temperature, typically by transportation in a "cold box." Problems in humanitarian logistics have numerous objectives, constraints, and uncertainties. These include uncertainty in demand forecast, road conditions, and time constraints. The main goal in humanitarian logistics is accessibility of products, rather than cost reduction.

The second motivating application area is for commercial companies to improve planning for delivery routes and product distribution. Fast and efficient distribution of goods, both regular and perishable goods, is important for companies such as Amazon. Companies that aim to deliver products at a low cost and in a short time have a competitive advantage that has been a differential for customers. Companies need to consider perishability of products, as well as integration of middle mile with last mile optimization models.

Another challenge to optimizing distribution is the computational difficulty of solving large-scale vehicle routing problems. It is well-known that most vehicle routing problems are NP-hard (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012), so many applied problems cannot be solved to optimality in reasonable time. Thus, much research focuses on exact and heuristic methods for vehicle routing problems, where there is a trade-off between fast solution techniques and confirmation of optimality (Laporte and Louveaux, 2002).

This research adapts and generalizes a Vehicle Routing and Scheduling Algorithm (VeRSA) (Zabinsky et al., 2019a) that embeds a heuristic in an exact method to address the need for fast and efficient computation while maintaining information on an optimality gap.

Research Objective and Approach

The primary objective of this research is to create optimization models and methods that provide insight and solutions to middle and last mile problems under uncertainty in an easy, fast, and broad way. The research objective is broken down into two specific objectives: create optimization models; and design solution methods.

Optimization Models and Applications The first research objective is to formulate optimization models that incorporate realistic constraints and address uncertainty and multiple objectives. To achieve this research objective, the first formulation concerns humanitarian logistics with a detailed application of vaccine distribution in Mozambique. The second formulation considers commercial distribution of perishable goods, and the third formulation integrates middle mile and last mile for commercial distribution.

Humanitarian logistics (e.g., vaccine distribution) optimization: The first optimization model and application is for the distribution of vaccines and medical supplies in Mozambique. Although vehicle routing is classically applied to commercial enterprises, it is important for non-profit organizations and governments to also seek to optimize their humanitarian logistics. The considerations for designing a routing plan includes specific constraints related to cold chain and transit time. Uncertainties in transit time, demand, and possibility of violating the temperature range for proper storage of vaccines makes this a challenging problem. For this problem, not only is the demand uncertain, but also the road condition, that may be affected by the weather or geographical location. Also, different transportation modes (e.g., bicycle, motorcycle, car, truck) have to be considered and evaluated. Moreover, since vaccines are perishable, the optimization formulation must include multiple objectives to balance the speed of distribution with the risk of waste. The goal is to distribute the vaccines and medical supplies to meet the demand in a timely manner at a minimum cost and risk of waste.

In this case, the distribution is between a federal storage house, regional centers, and health facilities; incorporating both middle and last mile. The medicines arrive at a federal government storage room and are distributed between regional distribution centers (middle mile) and local clinics (last mile). Data provided by VillageReach for distribution in Mozambique included two provinces, with 21 regional districts with one distribution center per district and 6 to 18 local clinics per district.

The optimization model includes two objectives and uses penalty parameters to represent the possibility of wasting vaccines due to temperature violation. For the optimization approach to be useful for routing planners in Mozambique, it was necessary for the computation time to be very fast. Instead of using classic mixed-integer programming solvers, such as Gurobi, CPLEX, GLPK or CBC, a new version of an indexing-based method (based on a Vehicle Routing and Scheduling Algorithm (VeRSA) (Zabinsky et al., 2019a)) was applied to this problem.

A combination of modeling and adaptation of VeRSA resulted in the development of a light-touch routing optimization tool for vaccine and medical supply distribution in Mozambique which is described in Chapter 3. The Route Optimization Tool (RoOT) is available on GitHub, in English at Petroianu (2019a) and in Portuguese at Petroianu (2019b). A paper describing RoOT was submitted for publication in the International Transactions on Operational Research (ITOR) (Petroianu et al., 2020b), and is a finalist in the IFORS prize for "OR in Development 2020" (Guajardo, 2020).

Optimization of perishable goods with commercial logistics (e.g., Amazon): With the increase of demand for perishable products needing a cold chain distribution, this research created several optimization models that consider uncertainty and multiple objectives with the use of robust optimization and stochastic programming with recourse.

The demand and transportation time are uncertain and will affect the allocation of products to the distribution centers and final customers. As with the situation with vaccine distribution, two objective functions were considered, the transit/transportation time and a metric for the loss of freshness to reflect quality of products on delivery. An efficient frontier is constructed to allow decision makers to consider tradeoffs between transit time (or cost) and perishability of goods. The robust optimization model provides a conservative solution, while the stochastic program provides a recourse action of adding expensive vehicles to handle extreme scenarios that would otherwise be infeasible. This is described in Chapter 4, and has been submitted for publication (Petroianu et al., 2020a).

Optimization of middle mile with last mile for commercial logistics (e.g., Amazon):

The third optimization model combines considerations for middle-mile logistics with last-mile distribution. As is common in commercial distribution, similar to Amazon, products are transported between different distribution centers by full workload trucks (middle mile) and then delivered to the final customer (last mile). The optimization model incorporates uncertainty of demand and delivery, considering time windows and changes in orders.

One of the main difficulties of this type of routing is that the trucks have to be allocated without the certainty of demand. This situation can lead to last minute cancellations or ad hoc demands that jeopardize planning. In this case, in addition to uncertain demand of the customers in the last mile, the middle-mile distribution is uncertain. For example, a truck may arrive at a distribution center expecting to pick up an order to transport, only to discover that the order has been cancelled. It is also possible that this cancelled order is rescheduled for a later time, thus the optimization model must consider order cancellations and additional ad hoc orders. This is discussed in Chapter 5.

Optimization Methodology and Solution Technique While mathematical models are important to describe the problems, the solution technique is also important to convey a meaningful solution. There are exact solution techniques available to solve small instances (e.g., small number of customers) and calculate bounds on the optimal solution. Thus exact methods can confirm optimality and provide an optimality gap on intermediate solutions. However, for larger instances, that include most of the realistic situations, exact solution techniques are too computationally expensive to be useful. Therefore, solving these problems using heuristics is important since they typically provide solutions that are good or near-optimal in a practical time. However, most heuristics sacrifice the optimality guarantee and cannot provide an optimality gap. In this dissertation, VeRSA (Zabinsky et al., 2019a) has been extended and generalized to solve large-scale routing problems. VeRSA combines an indexing approach with a quickly calculated lower bound, to provide a hybrid method that combines a heuristic with an exact lower bound, providing a good feasible solution with an optimality gap quickly.

Chapter 6 includes a description of VeRSA and how it has been adapted and generalized to include different constraints and objectives. Benchmark problems are included to demonstrate its scalability. A specific implementation of VeRSA is integrated in RoOT and used for vaccine and medical supply distribution in Mozambique.

The following chapters present the work done to achieve these research objectives. Chapter 3 gives a mathematical model and an indexing method to plan last mile distribution for vaccine and medical supplies. It also presents an open source tool developed to aid planners responsible for vaccine and medical supply routing in rural regions. Chapter 4 discusses different mathematical models that incorporate uncertainty of transit time, demand and loss of freshness for perishable products in last mile distribution. Chapter 5 gives a stochastic formulation for the middle mile routing problem with uncertainty of demand, considering cancellations and ad hoc additional orders. Chapter 6 presents a generalization of VeRSA for vehicle routing problems with time windows and compares its performance to state-of-art algorithms. Lastly, Chapter 7 gives concluding remarks and future research.

Chapter 2

BACKGROUND AND LITERATURE REVIEW

2.1 Vehicle Routing Problem

Vehicle routing problems have long been the focus of researchers (Gendreau et al., 1996; Laporte and Louveaux, 2002). Vehicle routing problems provide optimization models and solution techniques that are used in this dissertation research *to create optimization models and methods that provide insight and solutions to middle and last mile problems under uncertainty in an easy, fast, and broad way.*

Laporte (1992) presents a comprehensive review of the problem on vehicle routing problems (VRP). There are many variations of the VRP that are studied, such as the addition of time constraints including time windows for delivery, total time for each route (Kohl and Madsen, 1997; Solomon and Desrosiers, 1988), and capacity constraints (Laporte and Louveaux, 2002; Sungur et al., 2008). VRP problems and their variations are discussed in Chapters 3, 4, and 5.

VRP is in general NP-hard and is difficult to solve exactly for instances with more than 50 customers (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012). Research on solving large scale vehicle routing problems often focuses on heuristic solution approaches, which do not guarantee optimality but can find good solutions to large scale problems quickly. Cordeau et al. (2002) and Vidal et al. (2012b) present extensive comparisons of heuristics applied to solve vehicle routing problems. Chapter 6 discusses solution methods to solve large scale problems.

2.2 Vehicle Routing Problem with Uncertainty, Multiple Objectives, and Perishable Goods

Most studies in VRP concentrate on a deterministic version of the problem. Of those that address uncertainty, most only consider one type, limiting themselves to study uncertainty in transit time or demand (Ritzinger et al., 2016). In addition to uncertainty, it is also important to consider multiple objectives. Most papers are limited to a single objective, such as minimization of transit time or cost. By considering uncertainty and multiple objectives, planning the routes and vehicles can incorporate risk and contingency planning, thus affecting the guarantee of a successful delivery.

Uncertainties are often present in VRPs. A VRP that considers uncertainties is called a stochastic vehicle routing problem (SVRP). Ritzinger et al. (2016) provides a comprehensive survey of SVRPs. The most common type of uncertainties considered in SVRPs are demand and transit time, where demand is the most studied (Gendreau et al., 1996; Gounaris et al., 2013). One of the earliest papers on SVRP with probabilistic demand is Tillman (1969). Uncertain demand and time are typically represented as random variables, with a probability distribution, or with scenarios. Fixed uncertainty intervals are also used to capture a range of uncertain values. Different strategies are used to solve SVRPs, such as probabilities of satisfying a constraint (Dror and Trudeau, 1986), penalties if a constraint is not satisfied (Lee et al., 2012; Dror and Trudeau, 1986), and probabilistic scenarios (Crama et al., 2018; Petroianu et al., 2019; Sahinidis, 2004).

Three common ways to model uncertainty in optimization are with chance constraints, robust optimization, and stochastic programming with recourse. Chance constraints use a threshold for failure and a probability of satisfying this threshold. Robust optimization is also commonly used to handle uncertainty (Gounaris et al., 2013; Sungur et al., 2008; Ben-Tal et al., 2011; Lee et al., 2012). It can approach uncertainties considering with intervals of uncertainty for satisfying constraints (Gounaris et al., 2013; Sungur et al., 2008). Stochastic programming with recourse is well-studied with many applications (Birge and Louveaux, 2011). In stochastic programming with recourse, the initial stage considers known information to make an initial decision, and the recourse determines a response to uncertainty (Dror and Trudeau, 1986; Zhang et al., 2016; Gendreau et al., 1996; Laporte and Louveaux, 2002; Laporte, 1992; Güner et al., 2012). Chapter 4 has a detailed discussion on SVRP.

An important factor to consider in modeling VRP or SVRP is the presence of multiple objectives. According to Jozefowiez et al. (2008), common goals in VRP include minimization of total transit time, total time required, total cost, and fleet size, or maximization of quality of service and profit. When more than one of these objectives is important, a multi-objective approach should be used.

Incorporating uncertainty into a multi-objective problem has additional challenges. Ide and Schöbel (2016) present an extensive survey of robust optimization with multiple objectives and uncertainty. The robust optimization for a single objective considers predefined uncertainty sets or scenarios. Those can be “strictly” robust, in which the solution has to be feasible in all scenarios, or “lightly” robust, in which the solution is feasible in the most likely scenario. Considering multiple objectives, a solution is “partially” robust if it is non-dominated for at least one scenario (Ide and Schöbel, 2016). Ide and Schöbel (2016) present different concepts and manners to evaluate solution robustness when the feasibility set does not change considering different scenarios.

Another challenge in VRP or SVRP is the existence of perishable goods to be distributed. A poor route plan can affect the quality of the goods, and lead to spoilage (Song and Ko, 2016). Perishable products need a cold chain distribution, important for maintaining an appropriate temperature for the products during distribution (Hsu et al., 2007).

The need of a cold chain is an important aspect to consider in perishable distribution, since inappropriate refrigeration can result in waste. According to Nahmias (1982), perishability has two categories: fixed lifetime and random lifetime. Fixed lifetime has a known lifetime, after which the product cannot be used. Products with random lifetime have an unknown lifetime that usually follows a distribution with an exponential decay. The quality of these products or customer value diminishes over time. Examples are fruits and vegetables

(Coelho and Laporte, 2014).

In addition, the challenge of transporting perishable products varies according to the region. In big cities, where demand is growing rapidly, the main uncertainty for distribution is transit time, which varies according to the time of day and can be affected by road infrastructure. In rural regions and low income countries, other challenges and uncertainties are also important, especially when we consider distribution of perishable vaccines and medicines. In this case, not only the demand is uncertain due to lack of accurate population estimates but also road accessibility is uncertain due to weather conditions and road infrastructure. Moreover, the number of disasters and affected populations have been growing, which increases the demand for blood products (Nagurney et al., 2012) and other perishable medical supplies. Chapter 4 has a detailed discussion on perishable goods distribution.

2.3 Humanitarian Logistics

Another important problem for supply chain and VRP is humanitarian logistics, more specifically, vaccine distribution. Vaccine distribution is a difficult problem for governments around the world, but it is especially challenging in poor neighborhoods and low and middle-income countries, where the demand is uncertain due to lack of accurate population estimates, and road infrastructure is poor, even inaccessible under some weather conditions. Chan et al. (2013) discuss the problem that low and middle-income countries have of adopting new vaccines. For example, 98% of newborns in low-income countries do not receive pneumococcal conjugate vaccines, according to their government plan, while in high-income countries, the number is 13%. The geography of many low-income countries, such as the lack of proper roads or transportation methods to reach populations in need (Chan et al., 2013), is an important factor. Therefore, it is important to understand this type of humanitarian logistics problem and adapt how other fields achieve efficient distribution under these conditions.

Van Wassenhove (2006) discusses the existing gap between supply chain tools for humanitarian organizations and those used in the private sector. Humanitarian organizations have begun to realize the value of logistics and supply-chain management tools used in the private sector to improve their operations. Consequently, humanitarian organizations have begun to adopt private sector practices in their operations (Sarley et al., 2017; Lee et al., 2016).

Considering the distribution of vaccines, it can be considered a capacitated vehicle routing problem (CVRP) (Laporte and Louveaux, 2002). The need of a cold chain is also an important aspect to consider in vaccine distribution. Vaccines are perishable and inappropriate refrigeration outside of ideal storage temperatures results in waste. The problem also includes timing constraints, such as a constraint on the time from departure to the time of delivery and a constraint on the time for a driver to complete a route as in Bräysy and Gendreau (2005); Chen et al. (2006); Grasas et al. (2014); Laporte (1992); Laporte et al. (1992); and Zabinsky et al. (2019a).

Although much has been written about the importance of transportation for humanitarian logistics, infrastructure risks such as information technology, financial systems, and transportation are rarely addressed, and those risks are responsible for most of the network disruptions. Since transportation is fundamental to humanitarian logistics, its risks should be properly accounted for (Baharmand et al., 2017). Road risks should be considered in the objective function, along with cost, travel time, and demand satisfaction (Özdamar and Ertem, 2015). Road failure, caused by flooding, road sink, or bridge collapse, could make a calculated route longer than expected or even infeasible (Hamedi et al., 2012). Penalty parameters are also used to incorporate a failure probability as in Hamedi et al. (2012).

Chapter 3 has a detailed discussion on humanitarian logistics and vaccine distribution.

2.4 Middle Mile Routing

Middle mile transportation consists of moving goods between facilities (e.g., with aircraft, rail, and/or trucks) while satisfying several constraints, including time windows for pickup and delivery Desrosiers et al. (1995).

Middle-mile routing is not an area as well studied as last-mile routing, especially vehicle routing problem with full workload (Arunapuram et al., 2003). However, its importance has increased significantly due to e-commerce. Companies such as Amazon and Walmart have warehouses and distribution centers located in different parts of the United States, and distributing and allocating products between them is a common practice to balance inventory near customers and reduce the time between shopping and delivery. The products are often transported between those warehouses using trucks. Usually, while routing these trucks, demand is specified as an order for a full truckload to transport goods between two centers. Those problems are typically modeled as a vehicle routing problem with time windows (VRPTW) or vehicle routing problem with pickup and delivery (VRPPD). Uncertainty also impacts middle mile transportation. However, most of the research done in the field considers deterministic models in which the orders are known in advance. Moreover, their main goal is to minimize empty loads or lanes (Arunapuram et al., 2003; Gronalt et al., 2003; Zolfagharinia and Haughton, 2017). According to Zolfagharinia and Haughton (2017), empty loads are one of the main operational issues in the transportation industry, and load cancellations is one of the causes of this problem. In addition, few papers present multi-depot models (Liu et al., 2010). Since the problem is NP-hard (Liu et al., 2010), most of the research done focuses on finding heuristic methods to solve it, especially in the presence of uncertainty (Arunapuram et al., 2003; Liu et al., 2010; Goel and Irnich, 2017). Zolfagharinia and Haughton (2016; 2017) present surveys with multiple methods to solve this VRPTW. One of the first papers to present stochastic demand for full workload transportation was Powell (1986). Some authors evaluate rescheduling caused by routing disruption due to vehicle breakdown or traffic conditions (Li et al., 2009a;b; Mu et al., 2011). Chapter 5 has a detailed discussion on middle mile distribution.

2.5 Solution Methods for Vehicle Routing Problem

Due to the complexity of solving a vehicle routing problem (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012), most of the recent papers in the field are focused on heuristic and metaheuristics approaches that give near-optimal solutions for large-scale problems in a practical time. Cordeau et al. (2002) and Vidal et al. (2013a) present an extensive review of different heuristic methods to solve VRPs.

All those heuristic methods present good results and are state-of-art in heuristics for VRPTW. However, they do not guarantee optimal solutions. Therefore, another branch of research combines heuristics and mathematical models to reach the optimal solution or to present an optimality gap, although they do not present results as quickly as heuristics (Prescott-Gagnon et al., 2009; Jepsen et al., 2006; Solomon, 1987).

Our VeRSA method is based on Zabinsky et al. (2019a), and combines a heuristic indexing method to find feasible good solutions in a timely manner with an exact branch-and-bound method, that offers optimality gaps and an optimal solution eventually. It calculates the lower bound using a weighted minimum spanning tree problem. Other ways of calculating lower bounds are relaxed linear programming and Lagrangian multiplier (Liu et al., 2010; Gronalt et al., 2003; Adulyasak et al., 2015; Kolen et al., 1987). Chapter 6 has a detailed discussion on solution methods.

Chapter 3

VEHICLE ROUTING PROBLEM FOR VACCINE DELIVERY USING VERSA

This chapter *formulates optimization models that incorporate realistic constraints and address multiple objectives, and presents a solution technique* to solve the distribution of vaccines and medical supplies in Mozambique. It presents a model and an indexing method for distributing vaccines and medical supplies in rural areas. Most of the chapter appeared in Petroianu et al. (2019; 2020b).

Planning vaccine distribution in rural and urban poor communities is challenging, due in part to inadequate vehicles, limited cold storage, road availability, and weather conditions. The University of Washington and VillageReach jointly developed and tested a user-friendly, Excel spreadsheet-based optimization tool for routing and scheduling to efficiently distribute vaccines and other medical commodities to health centers across Mozambique. This chapter describes the tool and the process used to define the problem and obtain feedback from users during the development. The distribution and routing tool, named Route Optimization Tool (RoOT), uses an indexing algorithm to optimize the routes under constrained resources. Numerical results are presented using three realistic datasets. RoOT can be used in routine or emergency situations, and may be easily adapted to include other products, regions, or logistic problems. RoOT is available on GitHub, in English at Petroianu (2019a) and in Portuguese at Petroianu (2019b).

3.1 Introduction

Distribution of vaccines in rural Mozambique faces many challenges, such as inadequate vehicles, limited cold storage, road availability, and variable weather conditions. Moreover, different from most routing problems, cost reduction is not the sole or main goal. Timely delivery of available medicines within constrained resources is the primary objective. Government and non-governmental organizations require tools for planning and scheduling distribution that are easy to use and easy to update when circumstances change. This chapter presents joint work between VillageReach, a nonprofit organization that transforms health care delivery to reach everyone, including the most rural and remote communities (VillageReach, 2019), and the University of Washington, Department of Industrial & Systems Engineering, to optimize delivery routes that can improve the efficiency of vaccine distribution when considering issues such as vehicle availability and reliability, road conditions, and weather. VillageReach (VR) and the University of Washington (UW) are working with the Mozambican Ministry of Health (MoH) to develop and test a user-friendly, Excel spreadsheet-based optimization tool, called the Route Optimization Tool (RoOT), for routing and scheduling to effectively distribute vaccines and other medical commodities to health centers across the country. RoOT is designed to be easily updated and executed, and considers the availability of roads, vehicles, and medical products to distribute. The tool can be used periodically for routine operations, or in emergency situations. RoOT can also be used for strategic planning when exploring the effect of changes in the situation (such as new or closed health centers, additional or fewer vehicles, new medical supplies, or new refrigerators) on distribution plans.

3.1.1 Outline of the chapter

Section 3.2 contains background material and a brief literature review of humanitarian logistics with a brief discussion of vehicle routing problems and algorithms. Section 3.3 includes a detailed description of RoOT. The mixed-integer optimization model is given in Section 3.4, and the indexing algorithm is discussed in Section 3.5. Numerical results comparing the performance of several solvers using three realistic datasets are presented in Section 3.6, and finally, conclusions are drawn in Section 3.7, followed by future work in Section 3.8.

3.2 Background and literature review

3.2.1 Humanitarian logistics

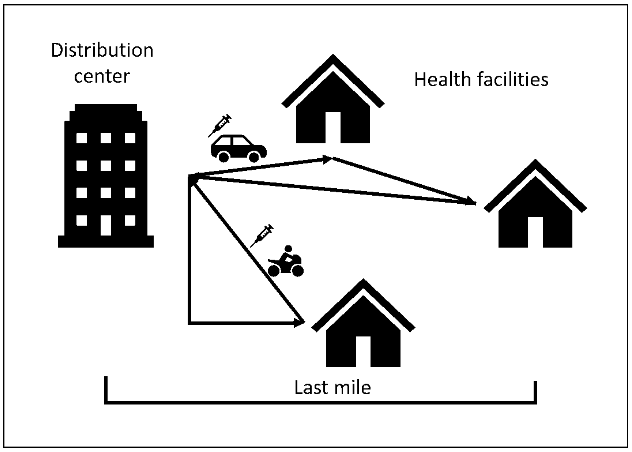


Figure 3.1: Last mile of vaccine delivery

Vaccine distribution is a difficult problem for governments around the world, but it is especially challenging in poor neighborhoods and low and middle-income countries, where the demand is uncertain due to lack of accurate population estimates, and road infrastructure is poor, even inaccessible under some weather conditions. Chan et al. (2013) discuss the problem that low and middle-income countries have of adopting new vaccines. For example, 98% of newborns in low-income countries do not receive pneumococcal conjugate vaccines, according to their government plan, while in high-income countries, the number is 13%. The geography of many low-income countries, such as the lack of proper roads or transportation methods to reach populations in need (Chan et al., 2013), is an important factor. Therefore, it is important to understand this type of humanitarian logistics problem and adapt how other fields achieve efficient distribution under these conditions.

Van Wassenhove (2006) discusses the existing gap between supply chain tools for humanitarian organizations and those used in the private sector. Humanitarian organizations have begun to realize the value of logistics and supply-chain management tools used in the private sector to improve their operations. Consequently, humanitarian organizations have begun to adopt private sector practices in their operations. For example, by using supply-chain practices from the private sector, Nigeria has increased its immunization coverage by about 30%, with a cost reduction of about 15% (Sarley et al., 2017). Through the use of computer simulation, Lee et al. (2016) redesigned the vaccine distribution process in two provinces in

Mozambique, as a joint effort with VillageReach. The redesign increased availability by 27% and 8%, while decreasing costs by 40% and 37%, respectively. However, some differences between a private sector supply chain and a humanitarian relief chain should be noted. In the private sector, the network configuration is more stable with respect to supply and demand (quantities and entities involved), while they are challenging to predict and are less consistent in humanitarian logistics (Manopiniwes and Irohara, 2014). In addition, cost is often the sole objective in the private sector whereas a humanitarian relief chain may prioritize rapid distribution using available resources. These differences motivate the development of a routing tool that is designed to address vaccine distribution in a developing country.

This project concentrates on last mile vaccine distribution. According to Laseinde and Mpofu (2017a), the last mile is the transportation of goods from a hub to the final destination. Figure 3.1 exemplifies the last mile for vaccine delivery.

3.2.2 Current tools

Limited technology is an important consideration for humanitarian logistics, such as poor Internet connectivity, lack of real-time data, and outdated computers. Supply chain tools need to be simple for users, and may not be adopted by the end users if they are complex to use, require data that is not easily available, require time to update, or need a fast reliable Internet connection.

There is a lack of tools for routing that are easy to use and address desired issues that were highlighted during interviews with VillageReach staff and Mozambican Ministry of Health (Vitoriano et al., 2013). We evaluated 33 commercial software products listed in Tables 3.1 and 3.2, where Table 3.1 lists 15 that focus on humanitarian logistics and Table 3.2 lists 18 that focus on routing. Most of the software that provide decision support systems for humanitarian logistics focus on inventory control and are primarily used for disaster preparedness and response but do not incorporate routing. These systems also primarily address management, but not operational issues (Vitoriano et al., 2013). Therefore, this project focused on developing a routing optimization tool that is simple to use, and in which the user interacts only with Excel files. The optimization algorithm is run in Python in the background to create the distribution routes and schedules.

3.2.3 Vehicle routing problem

The problem addressed in this project is a vehicle routing problem (VRP) in which routes between centers are planned such that each center is visited once by a vehicle, and each vehicle starts at the distribution center (often termed a depot) and returns to it by the end of its route. This vehicle routing problem is widely studied and Laporte (1992) presents a comprehensive review of the problem. There are many variations of the VRP that are studied, such as the addition of time constraints including time windows for delivery, total time for each route (Kohl and Madsen, 1997; Solomon and Desrosiers, 1988), and capacity constraints (Laporte and Louveaux, 2002; Sungur et al., 2008).

The problem considered in this chapter includes capacity constraints on the vehicles. Therefore, it is considered a capacitated vehicle routing problem (CVRP) (Laporte and Louveaux, 2002). The need of a cold chain is also an important aspect to consider in vaccine distribution. Vaccines are perishable and inappropriate refrigeration outside of ideal storage temperatures results in waste. Since the vehicles in Mozambique use “cold boxes” as passive containers to keep the vaccines within the proper temperature range, we consider the capacity of cold storage by vehicle type. The size of the vehicle (e.g., motorcycle, car, truck) determines the limited total capacity for its passive container for cold storage and other medical supplies that do not require refrigeration. Therefore, we incorporate two types of capacity constraints per vehicle, called cold and dry capacities.

Large and small cold boxes can maintain the proper temperature for a specified maximum amount of time before the vaccines are either used or transferred to a refrigerator at a health center. In this model, the total time of a route is constrained so that the cold box will preserve the vaccine until final delivery. We do not allow transfer of products between health centers, as it would require intermediary storage and some regions do not have electricity. Therefore, in addition to the vehicle capacity constraint, we also include timing constraints, such as a constraint on the time from departure to the time of delivery and a constraint on the time for a driver to complete a route (typically eight hours) as in Bräysy and Gendreau (2005); Chen et al. (2006); Grasas et al. (2014); Laporte (1992); Laporte et al. (1992), and Zabinsky et al. (2019a).

Different factors can lead to vial wastage in vaccine distribution. In addition to refrigeration or cold boxes keeping the vaccines within a proper temperature range, closed vial wastage may be due to breakage of vials (Hanson et al., 2017). Duttagupta et al. [20] show that the wastage rate for a 10-dose vial can be up to 62%. We include the risk of breakage of vials due to poor road condition or poor vehicle condition with a penalty parameter.

In humanitarian logistics, most routing problems focus on disaster preparedness, e.g., earthquakes (Ahmadi et al., 2015; Mete and Zabinsky, 2010; Tofighi et al., 2016). These problems are commonly modeled as classical vehicle routing or dynamic network problems, having as objective the minimization of total travel time, unmet demand, or cost (Özdamar and Ertem, 2015). Hoyos et al. (2015) consider the use of operations research in disaster operations management. In 48 papers using mathematical models, the most common goals were to minimize cost (31%), minimize unmet demand (21%), and maximize regional coverage (19%).

The model presented in this chapter has two objective functions, i.e., minimization of transportation time and minimization of risk factor for spoilage and breakage of vaccine using penalties for use of certain roads and vehicles. Although much has been written about the importance of transportation for humanitarian logistics, infrastructure risks such as information technology, financial systems, and transportation are rarely addressed, and those risks are responsible for most of the network disruptions. Since transportation is fundamental to humanitarian logistics, its risks should be properly accounted for (Baharmand et al., 2017). According to Özdamar and Ertem (2015), road risks should be considered in the objective function, along with cost, travel time, and demand satisfaction. Furthermore, road failure, caused by flooding, road sink, or bridge collapse, could make a calculated route longer than expected or even infeasible (Hamedi et al., 2012). An option is to consider the reliability of the transportation scheme, such as the probability of not completing a route. Penalty parameters are also used to incorporate a failure probability as in Hamedi et al. (2012). The use of penalty parameters reduces the amount of traffic (e.g., number of vehicles) on unreliable roads (Hamedi et al., 2012). Another way to calculate risk is to estimate the probability a road between two centers is inaccessible. Nolz et al. (2011) identify critical roads that could be bottlenecks in a tour. Avoiding the bottleneck decreases the total risk. In one example, using a minimum risk approach, the risk range decreased from 0.97-0.98 to 0.81-0.83. However, the travel time range increased from 0.25-0.61 hour to 0.59-0.98 hour. Since there is a trade off between risk and travel time, the users of RoOT will decide whether to minimize risk (e.g., penalties) or minimize transit time, or minimize a weighted sum of both objectives.

While analyzing risks for humanitarian logistics is not typically addressed in the literature (Baharmand et al., 2017), risk minimization is one of the main goals when transporting hazardous materials. Since the 1970s, the National Transportation Safety Board recommends a risk-based approach for transporting hazardous materials (List et al., 1991). One of the reasons there is a large quantity of research done on risk minimization is that fatalities due to hazmat-related traffic accidents are considered unacceptable (Akgün et al., 2007). Among multiple objectives for routing hazardous materials, minimization of risk should be the main objective (Patel and Horowitz, 1994).

Some approaches for risk analysis of hazardous material transportation incorporate an evaluation of accident rates by mode, carrier type, vehicle type, and road classification (List et al., 1991). According to List et al. (1991), private vehicles have lower accident rates than for-hire vehicles, and accident rates due to the time of day and weather conditions depend on the roadway type. In addition, accident rates may consider the road classification (expressways, arterials, collectors, ramps), the designed speed, the surface condition, and visibility (Saccomanno and Chan, 1985).

Considering the weather and time of day (e.g., daylight or night) is also important. Weather affects not only the transit time but also the risk of an accident. This includes the risk of the harm that a hazardous spill can do to the nearby population (Akgün et al., 2007). Moreover, other risks that may be addressed are the probability of an accident or delay at a facility, the accident rate en route, and the probability of an accident due to speed and road condition (Batta and Chiu, 1988).

Therefore, to minimize the spoilage and breakage of vaccines, the methodology used for hazardous materials was applied and vehicles and road were classified according to their conditions, assigning corresponding penalties to each. The corresponding objective function representing risk is the minimization of the sum of these penalties.

3.2.4 Exact methods and heuristics for solving vehicle routing problems

The vehicle routing problem is in general NP-hard and is difficult to solve exactly for instances with more than 50 customers (Laporte and Louveaux, 2002). Exact methods that guarantee an optimal solution usually start with a relaxation of the linear problem, followed by a presolve phase to reduce the problem. Then they apply branch-and-bound, branch-and-cut, or cutting-plane algorithms to solve the problem exactly (Forrest et al., 2018; Gurobi, 2019; GNU, 2019; Martin, 2010). Research on solving large scale vehicle routing problems often focuses on heuristic solution approaches, which do not guarantee optimality but can find good solutions quickly to large scale problems. Cordeau et al. (2002) and Vidal et al. (2012b) present extensive comparisons of heuristics applied to solve vehicle routing problems. Common heuristics are Tabu Search, Genetic Algorithms, and Greedy Randomized Adaptive Search Procedure (GRASP) (Berbeglia et al., 2010a; Gendreau et al., 1999; Grasas et al., 2014; Hanson et al., 2017; Kontoravdis and Bard, 1995; Taillard et al., 1997; Vidal et al., 2012a).

To address large scale problems, this chapter applies a variant of the Vehicle Routing and Scheduling Algorithm (VeRSA) presented in Zabinsky et al. (2019a). VeRSA is an exact method that embeds an indexing rule to prioritize pickups on different routes in a branch-and-bound framework, dynamically constructing the branches to be traversed. Therefore, it is possible to reach a near-optimal feasible solution quickly, while guaranteeing an optimal solution if the user runs it long enough. In Zabinsky et al. (2019a), the performance of VeRSA compared favorably to a commercial solver and a genetic algorithm. In this chapter, we adapted the indexing algorithm used in VeRSA to our vaccine distribution and routing problem, as discussed in Section 3.5.

3.3 Description of the Route Optimization Tool (RoOT)

3.3.1 Problem definition

The routing model developed in this paper was extensively discussed with the users, in this case, team members in VillageReach and the Mozambican government.

Initially, a team of UW and VR members held several teleconferences to discuss objectives of a light-touch routing tool for potential government users in Mozambique, and considered the current problems with existing tools. After several conference calls between VR team members in Seattle and Mozambique, a UW doctoral student (Petroianu) visited Mozambique to determine important features of a light-touch tool by seeing how routes are conducted. The objectives and constraints were defined by interviewing and interacting with the final users in the Mozambican government, and with VillageReach staff supporting the government for vaccine distribution planning. This interaction was important for the acceptance of the tool by the future users, since they were a part of its development, and they understand that the tool was created to meet their needs.

3.3.2 Model approach

The routing optimization model in RoOT is a mixed integer program (MIP) with two objectives and constraints tailored to the vaccine distribution problem in Mozambique. The participation of all team members in the modeling effort aided in identifying important considerations. Discussions of how to incorporate data from existing sources into Excel spreadsheets were also critical for the light-touch tool to be accepted and used.

In every province in Mozambique, vaccine distribution is done by district. Three data sets are included in this chapter, with one district in each of three provinces (Tete, Maputo, and Sofala). For example, one district in the Sofala province has 16 health centers, 5 vehicles, and 13 products.

From discussions with the stakeholders, it was decided that the model should consider the following:

- *Multiple objectives with different weights.* The users agreed that two objectives are important. The first is to minimize total transit time and the second to minimize the penalties for using vehicles or roads that are not in good condition. The users can decide how to combine the two objective functions by choosing an appropriate set of weights. The penalty parameter was also discussed. Penalty parameters are used to represent unreliable vehicles and poor road conditions that can jeopardize the delivery of viable vaccines (breakage and temperature range). Since it is difficult for users to scale penalty parameters appropriately, the Excel input file has dropdown menus and the users select the appropriate vehicle and road condition (e.g., there are four options for a vehicle ranging from very reliable to unreliable). The penalty parameter values are determined in the Python code to maintain appropriate scaling.

- *Supply and demand issues.* It was discussed that the demand projections can exceed the supply, and also that demand can exceed the storage capacity at a health center. Usually, vaccines are distributed once a month in Mozambique, so it was agreed that if there is insufficient supply to meet all of the monthly demand, then the input demand will be scaled back until it reaches the available supply. In this case, the recommendation is to do more than one delivery during the month, when the products arrive. Sometimes the demand at a health center exceeds the current storage capacity at the center, however, it is anticipated that there will be sufficient storage when needed (perhaps a refrigerator is awaiting repair). In this case, a warning is issued but the optimization can still be run.

- *Allow routes in parallel.* The user inputs all available vehicles, and each vehicle is assigned a route. In addition, each route has a maximum duration time (typically eight hours, specified by the user). If needed to achieve the complete distribution, vehicles can be assigned more than one route. However, all available vehicles should be used before one is reused. For example, if two vehicles are available, each will be assigned a route that may be completed on the same day. However, if the two vehicles do not have enough carrying capacity or time to deliver all of the vaccines, they can be assigned an additional route to complete on another day.

- *Cost.* Although the users do not want to necessarily minimize cost in the objective function, they are still interested in the cost of the routes and distribution plan. Cost calculations are provided in the output file based on input cost parameters. It should be noted that the costs are calculated after defining the routes, and are not part of the optimization model.

3.3.3 Usability

It is important that the tool be easy to use, and usability of the route optimization tool led to the decision to use Excel spreadsheets for inputs and outputs. The optimization algorithm is run by clicking on executable Python code, which allows a user to browse and select an input file. An Excel output file is created containing the routes with details on the types and quantities of vaccines and medical supplies to be delivered.

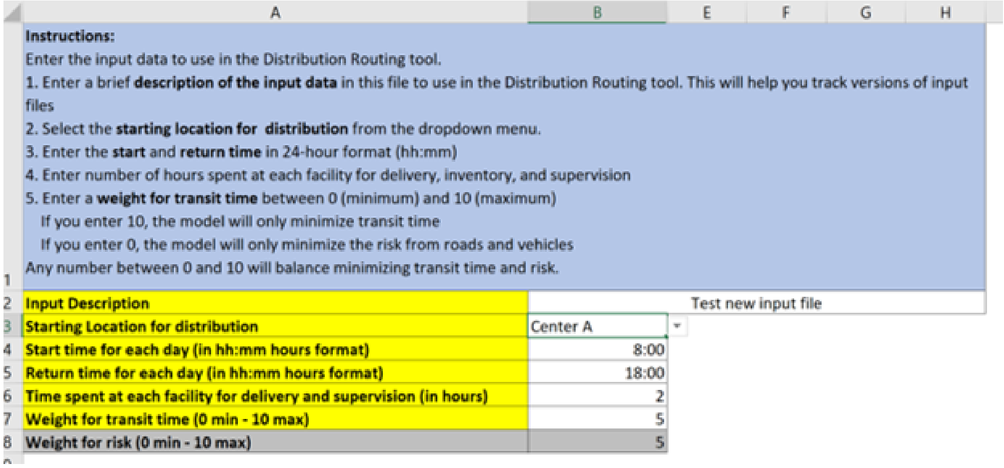
VillageReach evaluated the first prototype according to usability, using methods from Nielsen and Mack (1994). As a result, it was determined that there were too many possibilities for typos by the users, so dropdown menus were incorporated in the input file for many parameters. Adding a new health center necessitates changes in several of the spreadsheets, so the input file was designed so that user only enters the name of the new health center once, and it is automatically replicated to the other sheets to avoid errors. Similarly, new vaccines, medical supplies, or vehicles are entered once and then automatically replicated to sheets with connecting cells. Worksheets and cells with calculations that are used by the Python program but are not important to the user are locked and hidden. Sections 3.4 and 3.5 present the input and output files.

3.3.4 Input file

The input file is an Excel file with seven spreadsheets. Each sheet has brief instructions in the first line, The headings that are highlighted in yellow require input from the user, and some of the input cells have drop down menus. The seven input sheets are:

- parameters
- products
- center_capacities
- demand
- vehicle
- distance_data
- road_condition

The parameters sheet (Figure 3.2) defines the general parameters of the vehicle routing problem, including: run description, start and final location, start time and return time for the route, drop-off time, and weights for each objective function. The weights are used to balance the objective of minimizing transit time with the objective of minimizing penalties for using roads or vehicles that can be risky to the product due to their condition. Any number between 0 and 10 will balance minimization transit time and risk. The total is always 10. For example, the user can input 6 for the transit time weight, and the sheet automatically calculates 4 for the weight on the penalties. If the user enters 10, the model will only minimize transit time. If the user enters 0, the model will only minimize the penalties for roads and vehicles (risk to vaccines).



Instructions:
Enter the input data to use in the Distribution Routing tool.
1. Enter a brief **description of the input data** in this file to use in the Distribution Routing tool. This will help you track versions of input files
2. Select the **starting location for distribution** from the dropdown menu.
3. Enter the **start** and **return time** in 24-hour format (hh:mm)
4. Enter number of hours spent at each facility for delivery, inventory, and supervision
5. Enter a **weight for transit time** between 0 (minimum) and 10 (maximum)
If you enter 10, the model will only minimize transit time
If you enter 0, the model will only minimize the risk from roads and vehicles
Any number between 0 and 10 will balance minimizing transit time and risk.

Input Description	Test new input file
Starting Location for distribution	Center A
Start time for each day (in hh:mm hours format)	8:00
Return time for each day (in hh:mm hours format)	18:00
Time spent at each facility for delivery and supervision (in hours)	2
Weight for transit time (0 min - 10 max)	5
Weight for risk (0 min - 10 max)	5

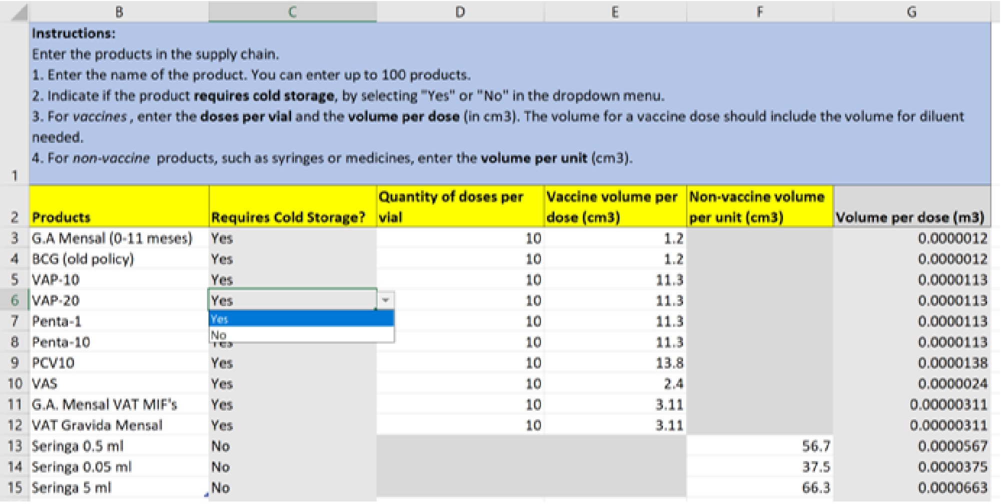
Figure 3.2: parameters sheet – input

In the products sheet (Figure 3.3), the user enters the products to be distributed and their volume and storage characteristics, such as doses per vial or number of syringes. The user also specifies if the product needs cold storage.

The center_capacities sheet (Figure 3.4) has health center information (name and type), and storage capacities for cold and dry products. When a user enters a new center name on this sheet, it is automatically added to the other sheets.

In the demand sheet (Figure 3.5), the user enters the demand for each product to be distributed to each center, in doses or units. If the demand exceeds the center capacity, the warning column will turn from green to red. If there is a warning, the user should adjust the demand or increase the storage capacity to make sure the center can store the delivered vaccines and supplies. Note that the optimization can still be run even if there is a warning.

The vehicle sheet (Figure 3.6) has the vehicle information that is used for delivering vaccines, their availability, and their characteristics, such as average velocity, fuel consumption, fuel costs, storage capacity, and personnel per diem costs for distribution.



Instructions:
Enter the products in the supply chain.
1. Enter the name of the product. You can enter up to 100 products.
2. Indicate if the product **requires cold storage**, by selecting "Yes" or "No" in the dropdown menu.
3. For *vaccines* , enter the **doses per vial** and the **volume per dose** (in cm3). The volume for a vaccine dose should include the volume for diluent needed.
4. For *non-vaccine* products, such as syringes or medicines, enter the **volume per unit** (cm3).

	B	C	D	E	F	G
2	**Products**	**Requires Cold Storage?**	**Quantity of doses per vial**	**Vaccine volume per dose (cm3)**	**Non-vaccine volume per unit (cm3)**	**Volume per dose (m3)**
3	G.A Mensal (0-11 meses)	Yes	10	1.2		0.0000012
4	BCG (old policy)	Yes	10	1.2		0.0000012
5	VAP-10	Yes	10	11.3		0.0000113
6	VAP-20	Yes	10	11.3		0.0000113
7	Penta-1		10	11.3		0.0000113
8	Penta-10	Yes	10	11.3		0.0000113
9	PCV10	Yes	10	13.8		0.0000138
10	VAS	Yes	10	2.4		0.0000024
11	G.A. Mensal VAT MIF's	Yes	10	3.11		0.00000311
12	VAT Gravida Mensal	Yes	10	3.11		0.00000311
13	Seringa 0.5 ml	No			56.7	0.0000567
14	Seringa 0.05 ml	No			37.5	0.0000375
15	Seringa 5 ml	No			66.3	0.0000663

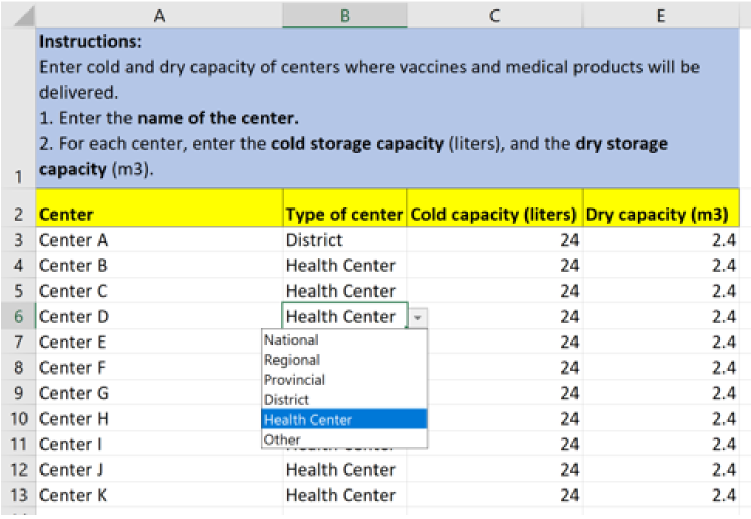
Figure 3.3: product sheet – input

The distance_data sheet (Figure 3.7) displays the distance between centers as a matrix. To provide flexibility in representing one-way roads, the distance matrix does not need to be symmetric.

Finally, the road_condition sheet (Figure 3.8) defines the condition of the road between centers using a dropdown menu, for the model to assess the risk of using that road. Options include: Fully paved, partially paved, dirt road (good quality), dirt road (rough quality), boat access only, foot access only, not accessible.

3.3.5 Output file

The Excel output file has two sheets: routes and products to be delivered. The routes sheet, in Figure 3.9, gives the recommended routes, including the distances traveled, fuel and personnel costs, utilized vehicle and its condition, utilized capacity per vehicle, dry and cold capacities, and the centers visited, giving the time to leave each of the centers and the road condition between them. The products sheet, in Figure 3.10, gives the quantity of products distributed to each center in a route, by dose or unit. It assigns a vehicle to the route and provides the utilized capacity, dry and cold, by center.



Instructions:
Enter cold and dry capacity of centers where vaccines and medical products will be delivered.
1. Enter the **name of the center.**
2. For each center, enter the **cold storage capacity** (liters), and the **dry storage capacity** (m3).

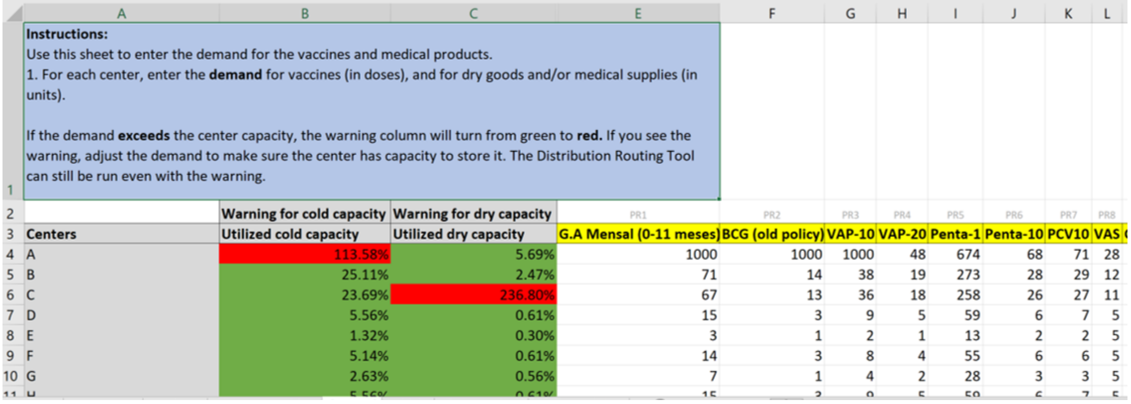
Center	Type of center	Cold capacity (liters)	Dry capacity (m3)
Center A	District	24	2.4
Center B	Health Center	24	2.4
Center C	Health Center	24	2.4
Center D	Health Center	24	2.4
Center E		24	2.4
Center F		24	2.4
Center G		24	2.4
Center H		24	2.4
Center I		24	2.4
Center J	Health Center	24	2.4
Center K	Health Center	24	2.4

Figure 3.4: center_capacities sheet – input

3.3.6 Use cases

While developing the tool, VR team members shared questions or use cases that the Mozambican government users often asked. RoOT was designed so that these questions can be easily answered. The questions include:

- What if my main distribution center changes location?
- What if a new vaccine is added for distribution?
- What if a new facility is added to my current list?
- What if one of my vehicles breaks down?
- What if I add a new vehicle to my fleet?
- What if the cold storage capacity at a health center is reduced?
- What if new refrigerators arrive?

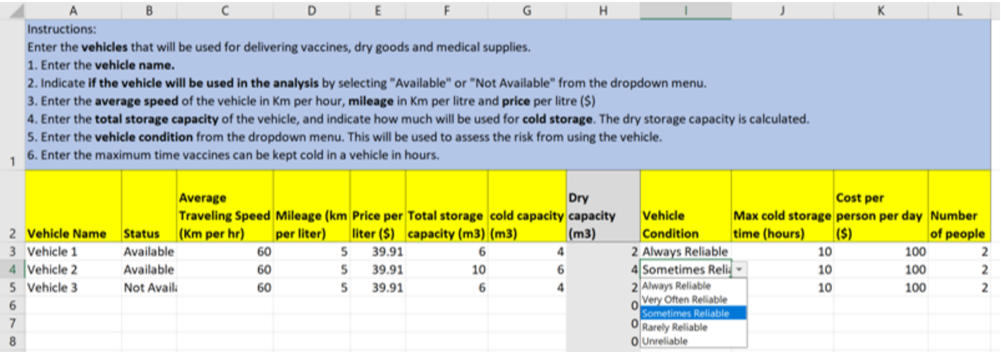


Instructions:
Use this sheet to enter the demand for the vaccines and medical products.
1. For each center, enter the **demand** for vaccines (in doses), and for dry goods and/or medical supplies (in units).

If the demand **exceeds** the center capacity, the warning column will turn from green to **red.** If you see the warning, adjust the demand to make sure the center has capacity to store it. The Distribution Routing Tool can still be run even with the warning.

	Warning for cold capacity	Warning for dry capacity	PR1	PR2	PR3	PR4	PR5	PR6	PR7	PR8
Centers	**Utilized cold capacity**	**Utilized dry capacity**	**G.A Mensal (0-11 meses)**	**BCG (old policy)**	**VAP-10**	**VAP-20**	**Penta-1**	**Penta-10**	**PCV10**	**VAS**
A	🟥 113.58%	🟩 5.69%	1000	1000	1000	48	674	68	71	28
B	🟩 25.11%	🟩 2.47%	71	14	38	19	273	28	29	12
C	🟩 23.69%	🟥 236.80%	67	13	36	18	258	26	27	11
D	🟩 5.56%	🟩 0.61%	15	3	9	5	59	6	7	5
E	🟩 1.32%	🟩 0.30%	3	1	2	1	13	2	2	5
F	🟩 5.14%	🟩 0.61%	14	3	8	4	55	6	6	5
G	🟩 2.63%	🟩 0.56%	7	1	4	2	28	3	3	5

Figure 3.5: demand sheet – input



Instructions:
Enter the **vehicles** that will be used for delivering vaccines, dry goods and medical supplies.
1. Enter the **vehicle name.**
2. Indicate **if the vehicle will be used in the analysis** by selecting "Available" or "Not Available" from the dropdown menu.
3. Enter the **average speed** of the vehicle in Km per hour, **mileage** in Km per litre and **price** per litre ($)
4. Enter the **total storage capacity** of the vehicle, and indicate how much will be used for **cold storage**. The dry storage capacity is calculated.
5. Enter the **vehicle condition** from the dropdown menu. This will be used to assess the risk from using the vehicle.
6. Enter the maximum time vaccines can be kept cold in a vehicle in hours.

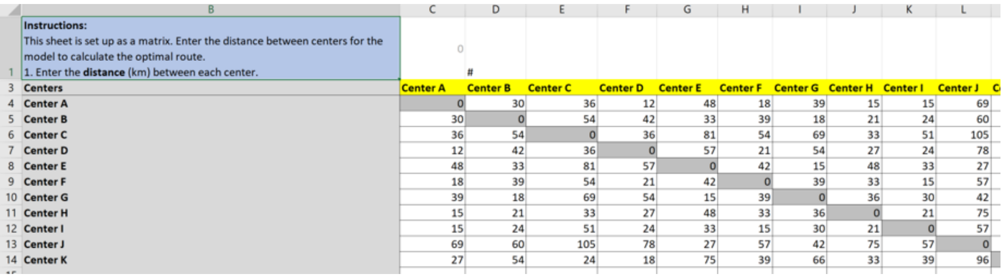
Vehicle Name	Status	Average Traveling Speed (Km per hr)	Mileage (km per liter)	Price per liter ($)	Total storage capacity (m3)	cold capacity (m3)	Dry capacity (m3)	Vehicle Condition	Max cold storage time (hours)	Cost per person per day ($)	Number of people
Vehicle 1	Available	60	5	39.91	6	4	2	Always Reliable	10	100	2
Vehicle 2	Available	60	5	39.91	10	6	4	Sometimes Reli	10	100	2
Vehicle 3	Not Avail	60	5	39.91	6	4	2		10	100	2
							0				
							0				
							0				

Figure 3.6: vehicle sheet – input

- What if there is an outbreak and a need for immediate distribution?

- What if a road is unavailable?

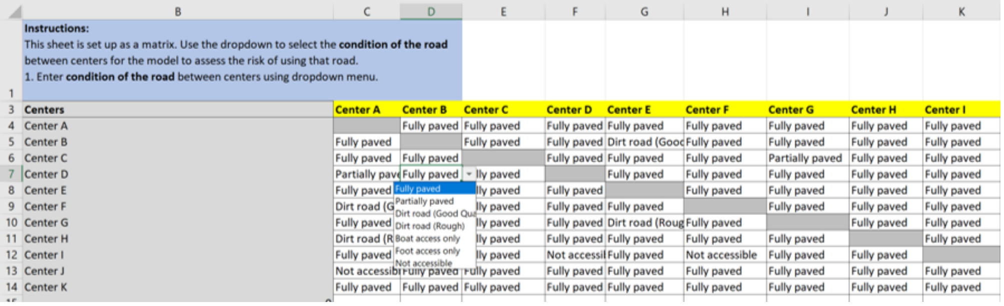
The user guide explains how to address each of these use cases. The need to easily add a health center, a new vehicle, a new product, or change capacities was instrumental in designing the input sheets. The Excel input file allows the addition in one place that is replicated across sheets.



Instructions:
This sheet is set up as a matrix. Enter the distance between centers for the model to calculate the optimal route.
1. Enter the **distance** (km) between each center.

Centers	Center A	Center B	Center C	Center D	Center E	Center F	Center G	Center H	Center I	Center J
Center A	0	30	36	12	48	18	39	15	15	69
Center B	30	0	54	42	33	39	18	21	24	60
Center C	36	54	0	36	81	54	69	33	51	105
Center D	12	42	36	0	57	21	54	27	24	78
Center E	48	33	81	57	0	42	15	48	33	27
Center F	18	39	54	21	42	0	39	33	15	57
Center G	39	18	69	54	15	39	0	36	30	42
Center H	15	21	33	27	48	33	36	0	21	75
Center I	15	24	51	24	33	15	30	21	0	57
Center J	69	60	105	78	27	57	42	75	57	0
Center K	27	54	24	18	75	39	66	33	39	96

Figure 3.7: distance_data sheet – input



Instructions:
This sheet is set up as a matrix. Use the dropdown to select the **condition of the road** between centers for the model to assess the risk of using that road.
1. Enter **condition of the road** between centers using dropdown menu.

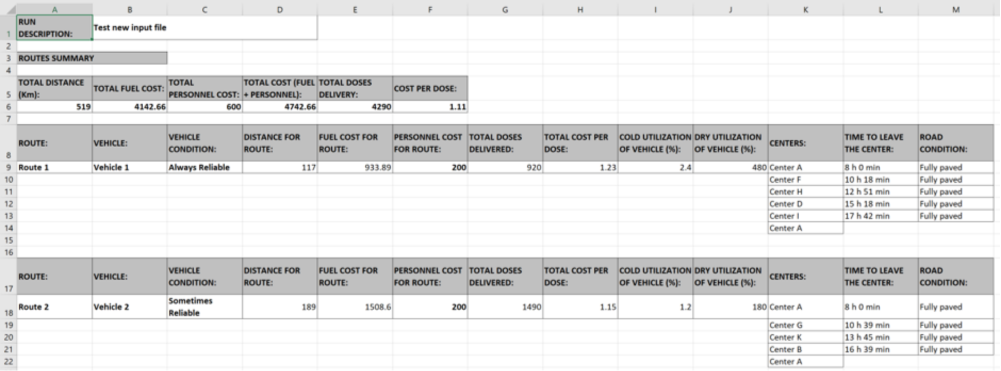
Centers	Center A	Center B	Center C	Center D	Center E	Center F	Center G	Center H	Center I
Center A		Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved
Center B	Fully paved		Fully paved	Fully paved	Dirt road (Gooc	Fully paved	Fully paved	Fully paved	Fully paved
Center C	Fully paved	Fully paved		Fully paved	Fully paved	Fully paved	Partially paved	Fully paved	Fully paved
Center D	Partially pave	Fully paved	lly paved		Fully paved	Fully paved	Fully paved	Fully paved	Fully paved
Center E	Fully paved		lly paved	Fully paved		Fully paved	Fully paved	Fully paved	Fully paved
Center F	Dirt road (G		lly paved	Fully paved	Fully paved		Fully paved	Fully paved	Fully paved
Center G	Fully paved		lly paved	Fully paved	Dirt road (Roug	Fully paved		Fully paved	Fully paved
Center H	Dirt road (R		lly paved	Fully paved	Fully paved	Fully paved	Fully paved		Fully paved
Center I	Fully paved		lly paved	Not accessil	Fully paved	Not accessible	Fully paved	Fully paved	
Center J	Not accessib	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved
Center K	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved	Fully paved

Figure 3.8: road_condition sheet – input

3.4 *Mathematical model*

The mathematical optimization model defining the problem is described in this section. Tables 3.3 and 3.4 presents the sets, decision variables and parameters of the model. It is based on the preliminary work presented in Petroianu et al. (2019).

In preliminary computational experiments, the number of products adversely affected the computation time. To speed up computation, products are classified as needing refrigeration (cold) or not needing refrigeration (dry) in a pre-processing phase. By grouping products into only two categories, the number of variables in the model is reduced and so is the computation time. This pre-processing, and similar post-processing are invisible to the user. The inputs can allow any number of products, and the outputs describe the products delivered along each route, that require different storage conditions. This approach reduces



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	RUN DESCRIPTION:	Test new input file											
2													
3	ROUTES SUMMARY												
4													
5	TOTAL DISTANCE (Km):	TOTAL FUEL COST:	TOTAL PERSONNEL COST:	TOTAL COST (FUEL + PERSONNEL):	TOTAL DOSES DELIVERY:	COST PER DOSE:							
6	519	4142.66	600	4742.66	4290	1.11							
7													
8	ROUTE:	VEHICLE:	VEHICLE CONDITION:	DISTANCE FOR ROUTE:	FUEL COST FOR ROUTE:	PERSONNEL COST FOR ROUTE:	TOTAL DOSES DELIVERED:	TOTAL COST PER DOSE:	COLD UTILIZATION OF VEHICLE (%):	DRY UTILIZATION OF VEHICLE (%):	CENTERS:	TIME TO LEAVE THE CENTER:	ROAD CONDITION:
9	Route 1	Vehicle 1	Always Reliable	117	933.89	200	920	1.23	2.4	480	Center A	8 h 0 min	Fully paved
10											Center F	10 h 18 min	Fully paved
11											Center H	12 h 51 min	Fully paved
12											Center D	15 h 18 min	Fully paved
13											Center I	17 h 42 min	Fully paved
14											Center A		
15													
16													
17	ROUTE:	VEHICLE:	VEHICLE CONDITION:	DISTANCE FOR ROUTE:	FUEL COST FOR ROUTE:	PERSONNEL COST FOR ROUTE:	TOTAL DOSES DELIVERED:	TOTAL COST PER DOSE:	COLD UTILIZATION OF VEHICLE (%):	DRY UTILIZATION OF VEHICLE (%):	CENTERS:	TIME TO LEAVE THE CENTER:	ROAD CONDITION:
18	Route 2	Vehicle 2	Sometimes Reliable	189	1508.6	200	1490	1.15	1.2	180	Center A	8 h 0 min	Fully paved
19											Center G	10 h 39 min	Fully paved
20											Center K	13 h 45 min	Fully paved
21											Center B	16 h 39 min	Fully paved
22											Center A		

Figure 3.9: routes sheet – output

the model complexity, once instead of routing over 10 products, it only considers two, P_r and P_d.

$$\min \; W_t \sum_{i\in C}\sum_{j\in C}\sum_{v\in V} e^{-h_{ijv}/\mu_h} y_{ijv} + W_p \sum_{i\in C}\sum_{j\in C}\sum_{v\in V} (e^{-\gamma_{ij}/\mu_\gamma} + e^{-\beta_v/\mu_\beta}) y_{ijv} \tag{3.1}$$

subject to

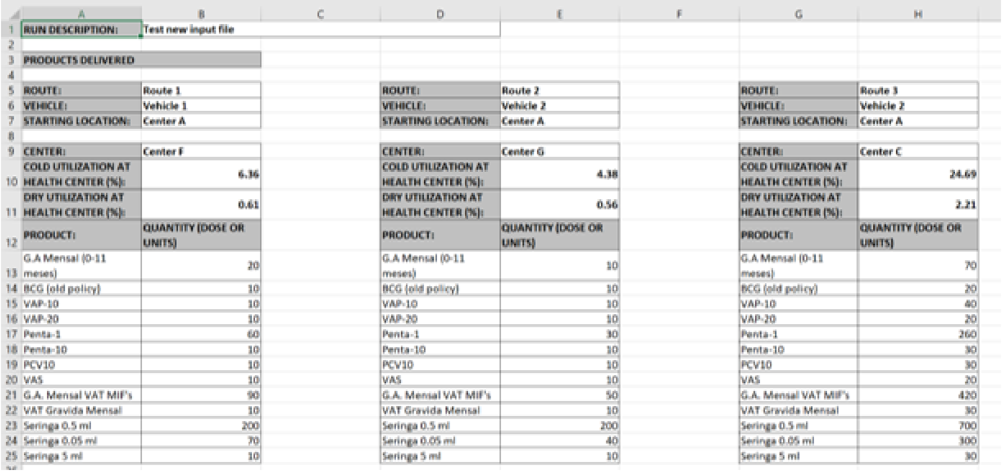
$$\sum_{i\in C}\sum_{v\in V} x_{ijvp} - x_{jivp} = d_{jp} \quad \forall j \in C, p \in P \tag{3.2}$$

$$\sum_{j\in C\setminus\{o\}}\sum_{p\in P_r} x_{ojvp} k_p \le c_v^r \quad \forall v \in V \tag{3.3}$$

$$\sum_{j\in C\setminus\{o\}}\sum_{p\in P_d} x_{ojvp} k_p \le c_v^d \quad \forall v \in V \tag{3.4}$$

$$t_{jv} - t_{iv} + M(1 - y_{ijv}) \ge h_{ijv} + W \quad \forall i \in C\setminus\{j\}, j \in C\setminus\{o\}, v \in V \tag{3.5}$$

$$t_{iv} + W(1 - y_{iov}) + y_{ijv}(h_{ijv} + h_{jov}) \le l \quad \forall i \in C, j \in C, v \in V \tag{3.6}$$



	A	B	C	D	E	F	G	H
1	RUN DESCRIPTION:	Test new input file						
2								
3	PRODUCTS DELIVERED							
4								
5	ROUTE:	Route 1		ROUTE:	Route 2		ROUTE:	Route 3
6	VEHICLE:	Vehicle 1		VEHICLE:	Vehicle 2		VEHICLE:	Vehicle 2
7	STARTING LOCATION:	Center A		STARTING LOCATION:	Center A		STARTING LOCATION:	Center A
8								
9	CENTER:	Center F		CENTER:	Center G		CENTER:	Center C
10	COLD UTILIZATION AT HEALTH CENTER (%):	6.36		COLD UTILIZATION AT HEALTH CENTER (%):	4.38		COLD UTILIZATION AT HEALTH CENTER (%):	24.69
11	DRY UTILIZATION AT HEALTH CENTER (%):	0.61		DRY UTILIZATION AT HEALTH CENTER (%):	0.56		DRY UTILIZATION AT HEALTH CENTER (%):	2.21
12	PRODUCT:	QUANTITY (DOSE OR UNITS)		PRODUCT:	QUANTITY (DOSE OR UNITS)		PRODUCT:	QUANTITY (DOSE OR UNITS)
13	G.A Mensal (0-11 meses)	20		G.A Mensal (0-11 meses)	10		G.A Mensal (0-11 meses)	70
14	BCG (old policy)	10		BCG (old policy)	10		BCG (old policy)	20
15	VAP-10	10		VAP-10	10		VAP-10	40
16	VAP-20	10		VAP-20	10		VAP-20	20
17	Penta-1	60		Penta-1	30		Penta-1	260
18	Penta-10	10		Penta-10	10		Penta-10	30
19	PCV10	10		PCV10	10		PCV10	30
20	VAS	10		VAS	10		VAS	20
21	G.A. Mensal VAT MIF's	90		G.A. Mensal VAT MIF's	50		G.A. Mensal VAT MIF's	420
22	VAT Gravida Mensal	10		VAT Gravida Mensal	10		VAT Gravida Mensal	30
23	Seringa 0.5 ml	200		Seringa 0.5 ml	200		Seringa 0.5 ml	700
24	Seringa 0.05 ml	70		Seringa 0.05 ml	40		Seringa 0.05 ml	300
25	Seringa 5 ml	10		Seringa 5 ml	10		Seringa 5 ml	30

Figure 3.10: products sheet – output

$$\sum_{i\in C}\sum_{j\in C} y_{ijv} - My_{ojv} \leq 0 \quad \forall v \in V \tag{3.7}$$

$$\sum_{i\in C}\sum_{j\in C} x_{ijvp} - Mx_{ojvp} \leq 0 \quad \forall v \in V, p \in P \tag{3.8}$$

$$\sum_{i\in C} y_{ijv} - y_{jiv} = 0 \quad \forall j \in C, v \in V \tag{3.9}$$

$$y_{ijv} \leq a_{ij} \quad \forall i \in C \backslash \{j\}, j \in C, v \in V \tag{3.10}$$

$$y_{iiv} = 0 \quad \forall i \in C, v \in V \tag{3.11}$$

$$My_{ijv} - x_{ijvp} \geq 0 \quad \forall i \in C, j \in C, v \in V, p \in P \tag{3.12}$$

$$y_{ijv} \in \{0, 1\} \quad \forall i \in C, j \in C, v \in V \tag{3.13}$$

$$x_{ijvp} \geq 0 \quad \forall i \in C, j \in C, v \in V, p \in P \tag{3.14}$$

$$t_{iv} \geq 0 \quad \forall i \in C, v \in V \tag{3.15}$$

The objective function (3.1) is a weighted sum of total transit time and total sum of all penalties for chosen vehicles and roads during transit. In the objective function, the exponential and the division by the means are used to normalize the parameter values and consider them in the same scale. Constraints (3.2) guarantee that the center demand at each center is met. Constraints (3.3) and (3.4) limit the amount or quantity that vehicle v can carry of cold and dry products, respectively. Constraints (3.5) give the time sequence between two sequential health centers. This means that if health center j follows i, the time that vehicle v leaves j has to be greater than the time that it passed by i plus the transit time between the centers and the time for product drop off. Constraints (3.6) guarantee that the vehicle will always have time to return to the supply node o, while respecting the maximum time for the route l. Constraints (3.7) and (3.8) ensure that each vehicle must depart from the main health center o. In these constraints, M is a large number. Constraints (3.9) require that if a vehicle enters a health center, it has to leave it, and constraints (3.10) forbid a vehicle from returning to a center immediately after leaving it. Constraints (3.11) limit the routes to the available routes, and constraints (3.12) ensure that a vehicle v traverses arc (i, j) whenever products are carried between them. Constraints (3.13) - (3.15) define binary variables y_{ijv} and non-negative variables x_{ijvp} and t_{iv}.

3.5 Indexing method

The VRP is NP-hard problem. Therefore, in general it is difficult to find an optimal solution for large instances of the problem, such as most real problems. In addition, when it is possible to reach an optimal solution, its calculation can take hours or even days. Therefore, in real situations, such as the ones from the distribution routing tool, it is more important to have a good solution in a timely manner than reaching an optimal solution in hours. The route planners need to define good routes fast, for different scenarios, such as daily planning or emergency situation. Therefore, developing a method that supports their decisions and analysis, and is feasible to be used is important.

The indexing algorithm used in RoOT is a variation of VeRSA, presented in Zabinsky et al. (2019a). VeRSA demonstrated better and faster results in comparison to Genetic Algorithm methods (Zabinsky et al., 2019a). The method is based on branch-and-bound (BnB) with an indexing rule to prioritize pickups on different routes, defining the BnB branches to be traversed. The indexing method created for this problem is based on the mathematical model objective functions and constraints. It is divided into two stages. The first stage defines which vehicle will be used. The index value for choosing a vehicle is calculated according to

$$e^{-\beta_v/\mu_\beta} + e^{c_v^r/\mu_{c^r}} + e^{c_v^d/\mu_{c^d}} + e^{v_v/\mu_v} \tag{3.16}$$

where β_v is the penalty for vehicle v, c_v^r is the capacity of vehicle v for refrigerated vaccines, c_v^d is the capacity of vehicle v for dry goods, and $\mu_\beta, \mu_{c^r}, \mu_{c^d}$, and μ_v are the averages, respectively, to scale appropriately. The vehicle with the largest index value is assigned a route next. When all available vehicles are assigned a route, in order of the index, then they may be assigned a second route.

After deciding the vehicle, its route is created using the index calculated in

$$W_t e^{-h_{ijv}/\mu_h} + W_p e^{-\gamma_{ij}/\mu_\gamma} \tag{3.17}$$

where i is the current center, j is the next center to visit, v is vehicle, h_{ijv} is the transit time from i to j using vehicle v, γ_{ij} is the penalty for road (i, j), β_v is the vehicle penalty, and W_t

and W_p are the weights for transit time and penalty objective functions, respectively. The averages, μ_h, μ_γ, and μ_β are used to scale appropriately. The node with the highest index value is added to the route.

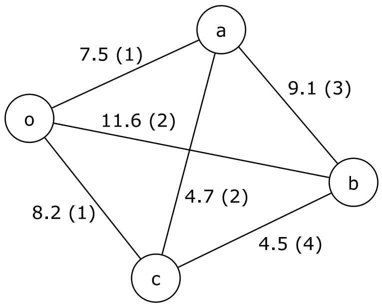
The choices of vehicle and next node to add to the route have to respect vehicle capacities, time limits, and road availability. These constraints are considered in a feasibility check that is performed every time a new node or vehicle is added to the route. The indexing algorithm constructs a feasible solution by assigning routes until the delivery of products is complete.

The indexing algorithm allows routes in parallel and uses all available vehicles before reusing any of them. This constraint does not appear in the mathematical model (3.1) – (3.15). However, in the numerical results, there was no need to use available vehicles more than once and the vehicle characteristics were similar, so the results are comparable.

To illustrate how to construct a feasible solution using the index, consider an example with four health centers ($o,\ a,\ b,\ c$), where o is the supply node (i.e., depot). Figure 3.11a gives the transit time, h_{ijv}, and the penalties, γ_{ij}, between centers. In this example, there is one vehicle with penalty, β_v, equal to one.

In Figure 3.11b, the index values calculated using (3.17) leaving node o and going to $a,\ b,\ c$ are shown in the first row. The largest value, 0.49, indicates that node b is visited next, as illustrated in Figure 3.11c. Then the index from b to $a,\ c$ is calculated, and the largest value of 0.46 indicates that node a is added to the route. Every time that a center is added, the index is calculated for all remaining centers, and the feasible center with highest index is added to the route, as shown in Figure 3.11.

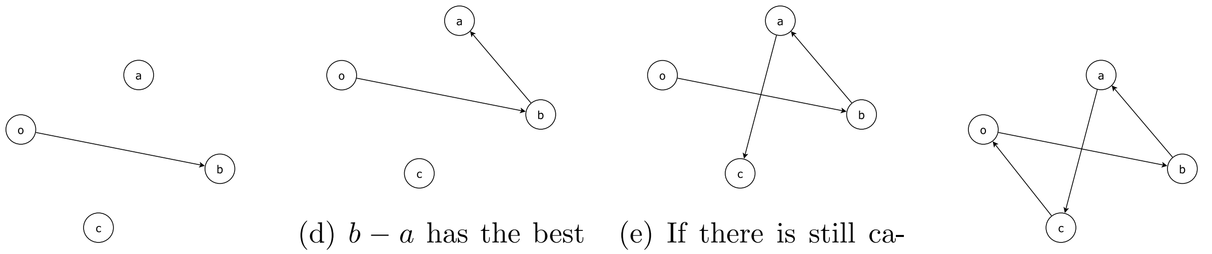
In the indexing algorithm presented in this chapter, the algorithm traverses the tree by calculating an incumbent solution for each initial branch of the tree, a depth evaluation of the branch. Then it creates an elite set with 12% of the incumbent solutions. This elite set is divided into two blocks of solutions. From its total, 50% come from the best incumbent solutions and 50% are solutions considering the largest solution uncertainty intervals. The uncertainty interval is the local best solution minus local lower bound. Each solution of this elite set represents a complete branch that will be explored in search of better solution and



(a) Transit time, h_{ijv}, and (penalties, γ_{ij})

	a	b	c
o	0.69	**0.49**	0.67
	a	c	
b	**0.46**	0.54	

(b) Index calculation considering available arcs



(c) $o - b$ has the best index

(d) $b - a$ has the best index of the remaining arcs

(e) If there is still capacity and time, the vehicle goes to c

(f) The vehicle returns to the depot o

Figure 3.11: Example of how to construct a feasible solution using the index

will help to traverse the tree. The global lower bound of the problem is defined as a minimum spanning tree in which the cost is related to the index in (3.17). The local lower bound is calculated exploring the current branch up to the its current level.

This indexing algorithm presents good results in comparison to the solvers for the same mathematical model, finding the optimal solution promptly for small datasets, and obtaining better optimality gaps for larger sets, as is demonstrated in Section 3.6.

3.6 Numerical results

In this section, three realistic datasets with information from Mozambican provinces were used as inputs to perform computational experiments. District A has 11 health centers, 13 products, and 1 vehicle. District B has 16 health centers, 13 products, and 2 vehicles. District C has 13 health centers, 12 products, and 6 vehicles, as shown in Table 3.5. The three datasets have the same penalties for all vehicles and roads. As mentioned in Section 4, the products are grouped into two types: refrigerated and non-refrigerated. Therefore, the models consider only two type of products while optimizing the routes. It is valid for all datasets.

The computational experiments compared the runtime and solution found by solving the same MIP with Gurobi 8.0.1, CBC, GLPK, and RoOT. All the tests were run on a Dell XPS13 computer, Intel CORE i7, with 16 GB of RAM. The weighted objective function used $W_t = W_p = 0.5$ for all tests.

Due to the problem sizes, optimality can be found in less than four hours only for District A. Therefore, three smaller datasets were created to check if the indexing algorithm could discover an optimal solution that was confirmed using Gurobi. The smaller data sets are based on a subset of centers in each district, called District A-small, District B-small, and District C-small. The size of each dataset is given in Table 3.5.

Two large test datasets with 50 health centers and 5 vehicles were also created to compare the solvers on large instances. The first large instance, called 50 centers simple, has identical vehicles and the penalties for all the road are the same. The second instance, called 50 centers modified, has five different vehicles and different penalties for the roads. All the instances are available on-line, see Petroianu (2019a).

For the small datasets, Gurobi was able to solve the MIP to optimality, as shown in Table 3.6. The indexing algorithm in RoOT found the same optimal solutions quickly for the three small data sets, but did not confirm optimality within 30 minutes of runtime. The open-source solver CBC also discovered the same optimal solutions, although taking more time, and did not confirm optimality within 30 minutes of runtime. The other open-source solver GLPK was able to discover the same optimal solutions for two of the three datasets, but reported an infeasible solution as "optimal" for the District A-small dataset.

The realistic datasets (District A, District B, and District C) were tested running the solvers for 30 minutes (i.e., 1800 seconds). The objective of the tool is to find good solutions in a timely manner. Therefore, the user will not run it more than 30 minutes, except if an optimal solution is needed. Figures 3.12 – 3.14 provide plots of solution versus runtime (in seconds) for District A, District B, and District C, respectively. Note that the runtime is plotted in logarithmic scale. An end-user typically expects a near-optimal solution in less than two minutes (i.e., 120 seconds). Table 3.7 summarizes the best solution found in 30 minutes, with its lower bound, to provide the optimality gap.

Gurobi found the optimal solution for District A, and RoOT had a performance similar to the CBC solver for that instance. For dataset District B, Gurobi, CBC, and RoOT had similar performances running for 30 minutes. For dataset District C, containing 6 vehicles, RoOT had better performance than all solvers, with small improvement over Gurobi.

However, as discussed in Section 3.2.4, vehicle routing problems are difficult to solve for large instances. To test the performance of RoOT in this situation, we created two test datasets with 50 centers and five vehicles. The distances are from the instance belgium-road-km-d2-n50-k10 (Smet, 2017).

The first dataset had five identical vehicles with the same penalty values, and all the roads also had the same penalty values. Figure 3.15 shows solver performances. CBC and GLPK could not find a feasible solution in 30 minutes. RoOT, found a feasible solution

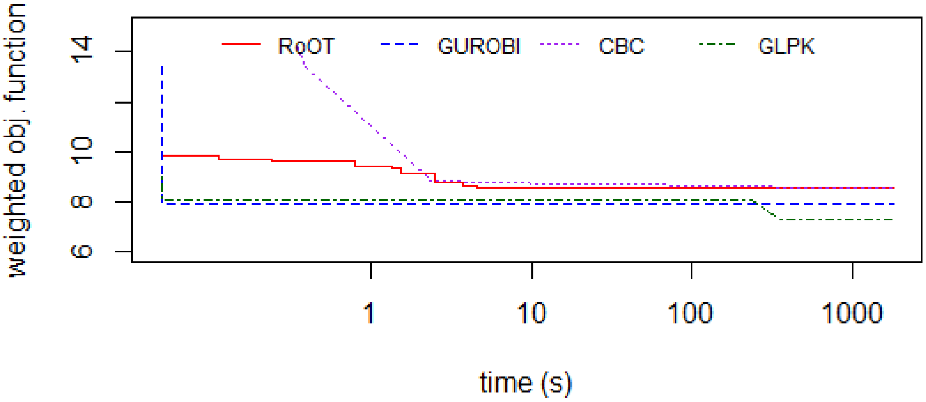


Figure 3.12: Solution comparison: District A (GLPK gave an infeasible solution smaller weighted objective function than the optimal calculated by Gurobi)

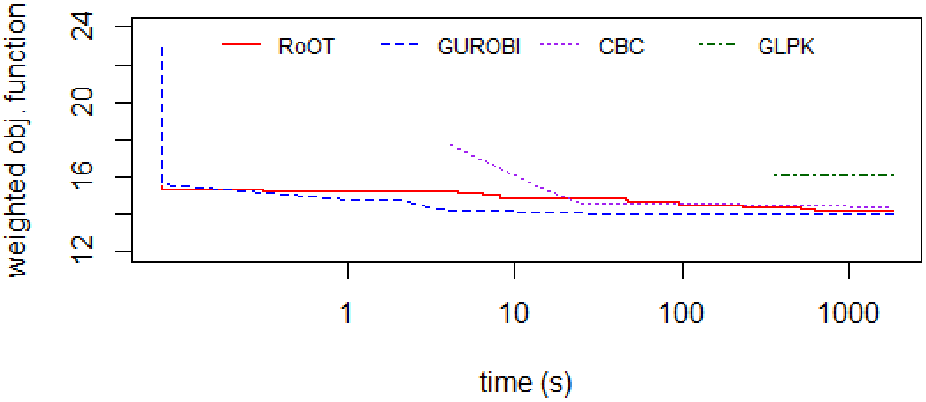


Figure 3.13: Solution comparison: District B

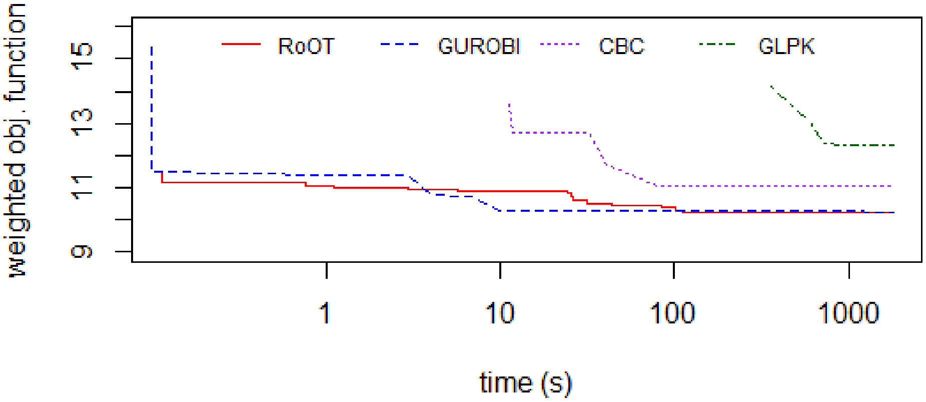


Figure 3.14: Solution comparison: District C

earlier than Gurobi, and Gurobi only reached the RoOT solution after 8.33 minutes (i.e.,500 seconds). However, Gurobi improved RoOT's solution and had a better solution at the end of 30 minutes. The difference in Gurobi's solution and RoOT's solution at 30 minutes was not significant.

The second dataset had five different vehicles, with different capacities and penalties. Moreover, the penalties for the roads were also different. Again CBC and GLPK could not find a solution in 30 minutes. RoOT found a feasible solution and converged earlier than Gurobi, and by the end of the 30 minutes, Gurobi had not reached the RoOT solution.

3.7 Conclusion

The main goal of this project is to create an easy-to-use routing tool that meets the needs of the users. The interactions between team members from the University of Washington, VillageReach, and the Mozambican Ministry of Health were essential to reach this objective. Through many discussions and meetings, we developed the tool presented in this paper. RoOT was initially presented to the users to obtain feedback, and the revision presented in

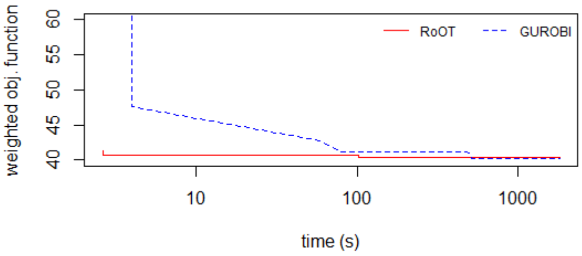


Figure 3.15: Solution comparison: 50 centers simple

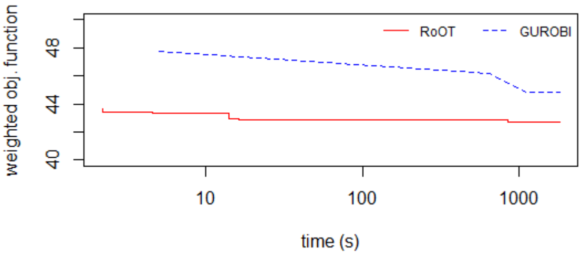


Figure 3.16: Solution comparison: 50 centers modified

2019 was favorably received. Continued training for MoH users will result in a final version in 2020 that will be publicly available on Github.

RoOT gives good solutions in a timely manner. The final users do not have time or resources available to run an optimization model for hours or days to find the optimal solution. They want a good solution in one or two minutes, and RoOT is capable of that, as shown in this paper. Moreover, RoOT obtained good solutions within two minutes on the 50 center datasets. Scalability and speed are important factors for the users.

Conclusion:

- tool easy to use
- importance of work with stakeholders
- good solution fast
- better for larger and more complex datasets

3.8 *Future work*

Considerations for future route optimization versions of RoOT include multiple day routes, with mixed transportation modes (e.g., land vehicles and boats) and island deliveries. This will require discussion on how intermediary storage of vaccines may be handled over multiple days. There is a risk of breakage and temperature range violation when unpacking and repacking vaccines in mid-route for intermediary refrigeration. Mixed modes also present issues of coordination of timing as well as capacity issues.

One of the main challenges in preparing the data for the tool is to define the distance matrix. There is an opportunity to develop another tool that uses information from mapping and map APIs to populate the matrix, using names of locations, postal codes, or geographic coordinates. Professor Zabinsky and I, Larissa P.G. Petroianu, mentored an undergraduate student that developed this tool (Snyder et al., 2020).

RoOT has a version translated to Portuguese for the Mozambican users. Both Portuguese and the English version are available on Github (Petroianu, 2019a) for other users from NGOs, government and academia.

The computational experience with RoOT has provided insights and ideas for future improvements. We will continue improving the indexing algorithm used in RoOT, to find better solutions faster and reducing the optimality gap with a tighter lower bound. We intend to test RoOT to evaluate its performance on more complex datasets with hundreds of centers. The improvements in this indexing algorithm and the tests in larger datasets are presented in Chapter 6.

Table 3.1: Humanitarian logistics tools/software

	Tool/Software name
1	LLamasoft – Supply Chain Guru – Cloud-Based Supply Chain Design Software
2	HERMES – Highly Extensible Resource for Modeling Event-Driven Supply Chains
3	GLC – Global Logistic Competence
4	SUMA
5	LSS
6	Fritz institute - Humanitarian Logistics Software (PRSRM-HLS)
7	HELIOS
8	Sahana
9	Chevinfleet
10	Logistimo
11	Parcel Project
12	UNICEF
13	ELIST
14	DMIS
15	LOGITIX

Table 3.2: Vehicle routing tools/software

	Tool/Software name
1	ClearD Optima
2	DISC
3	Intelligent Routing
4	JOpt
5	ODL Studio
6	OptimoRoute
7	Optrak4
8	Routist
9	Routyn
10	Scientific Logistics Cloud-based Route Optimization
11	StreetSync Pro
12	Locus Dispatcher
13	Workwave Route Manager
14	Onfleet
15	Routific
16	Loginext
17	Track POD
18	Cro software solutions

Table 3.3: Model notation - sets and decision variables

Sets	
Health centers – C	$i \in C$, o is the supply node
Vehicles – V	$v \in V$
Refrigerated products – P_r	$p \in P_r$
Non-refrigerated (dry) products – P_d	$p \in P_d$
Products – P	$p \in P$
Decision Variables	
y_{ijv}	binary variable: equals 1 if products are transported from i to j using vehicle v; and equals 0 otherwise
x_{ijvp}	quantity of product p transported from i to j using vehicle v
t_{iv}	time that vehicle v leaves health center i

Table 3.4: Model notation - parameters

Parameters	
W_t	weight for minimizing the total transit time, in $[0, 1]$ interval
W_p	weight for minimizing the total penalties, $W_p = 1 - W_t$
μ_h	average of all transit times
μ_β	average of all vehicle penalties
μ_γ	average of all road penalties
d_{ip}	demand at health center i for product p
h_{ijv}	average transit time between i and j using vehicle v
c_v^r	transportation capacity of vehicle v carrying cold products r
c_v^d	transportation capacity of vehicle v carrying dry products d
l	maximum time for a route
k_p	volume of product p
a_ij	route availability; equals 1 if route (i, j) is available; and equals 0 otherwise
γ_{ij}	penalty for driving between i and j
β_v	penalty for driving with vehicle v
W	time for product drop off
M	big number

Table 3.5: Size of test datasets

Dataset	Quantity of centers	Quantity of vehicles
District A-small	8	1
District B-small	8	2
District C-small	8	3
District A	11	1
District B	16	2
District C	13	6
50 centers simple	50	5
50 centers modified	50	5

Table 3.6: Computational comparison for the small datasets

	Optimal solution	MIP[a] Gurobi (s)	MIP[b] CBC (s)	MIP[b] GLPK (s)	RoOT[b]
District A-small	5.90	3.48	13.23	4.40[c]	1.02
District B-small	6.69	4.31	100.61	60.00	1.88
District C-small	6.20	12.35	13.23	540.20	0.90

[a] Time for optimal solution.

[b] Time to first discover the optimal solution.

[c] GLPK gave a different solution in comparison to the other 3 solvers: 4.63.

Table 3.7: Computational comparison for five datasets

	District A		District B		District C		50 centers simple		50 centers modified	
	Best solution	Lower bound	Best solution	Lower bound	Best solution	Lower bound	Best solution	Lower bound	Best solution	Lower bound
RoOT	8.57	6.16	14.24	6.11	10.23	2.93	40.46	21.36	42.73	16.03
Gurobi	7.93	7.93	14.06	10.11	10.24	8.03	40.24	24.26	44.81	21.05
CBC	8.57	4.69	14.40	2.47	11.05	1.29	-	-	-	-
GLPK	7.30[a]	4.81	16.13	7.93	12.34	1.07	-	-	-	-

[a] GLPK gave a solution smaller than the optimal calculated by Gurobi

Chapter 4

DISTRIBUTION OF PERISHABLE PRODUCTS UNDER UNCERTAINTY

This chapter *formulates optimization models that incorporate realistic constraints, uncertainties, and address multiple objectives.* It presents different mathematical models that incorporate uncertainty in transit time, demand and loss of freshness for perishable products in last mile distribution. Most of the chapter appeared in Petroianu et al. (2020a).

On-line shopping has become popular and its demand is increasing due to the convenience of receiving a product in the comfort of one's own home. Perishable goods shopping is also increasing, as more supermarkets and companies offer to deliver grocery shopping to the person's house. However, a poor route plan can affect the quality of the perishable goods, their freshness, even leading to spoilage. Therefore, good route planning and scheduling under uncertainty is especially important for perishable products. In this chapter, we model a stochastic vehicle routing problem for perishable products delivery with uncertainty of demand, transit time, and freshness time. We use min-max robust optimization utilizing the *a posteriori* approach, and stochastic programming with recourse. The models have three objective functions, i.e., minimization of transportation time, minimization of potential spoilage, and minimization of loss of freshness. We compare and contrast how the models incorporate uncertainty and construct an efficient frontier to evaluate the objective functions presenting numerical results and discussing differences between the models.

4.1 Introduction

The prevalence of e-commerce has changed how customers shop for products. The convenience of receiving a product in the comfort of one's own home increased the popularity of on-line shopping. According to Zhou et al. (2018), 20.6 billion packages were delivered in China in 2015, with an expectation of increasing to two billion packages daily by 2022. In addition, perishable goods shopping, such as for groceries (Song and Ko, 2016), is also affected by the increase of e-commerce. This trend increased the importance of good route planning and scheduling under uncertainty, and it is especially important for perishable products. A poor route plan can affect the quality of the goods, their freshness, even leading to spoilage (Song and Ko, 2016). Perishable products need a cold chain distribution, important for maintaining an appropriate temperature for the products during distribution (Hsu et al., 2007). Zhou et al. (2018) note that the *last mile*, the transportation of goods from a hub to the final destination (Laseinde and Mpofu, 2017b), is the least efficient and most expensive part of the total supply chain process. We concentrate our work on the last mile of cold/perishable product distribution.

Perishable products are goods that should be discarded after a limited lifetime (Nagurney et al., 2012). Besides food, there is a large range of different products that can be considered perishable. This includes newspapers, flowers, chemical materials, clothing (Coelho and Laporte, 2014), and medical supplies, including blood products (Amorim and Almada-Lobo, 2014; Grasas et al., 2014), and vaccines (Petroianu et al., 2019; 2020b). Therefore, last mile distribution is a challenge not only for industries, but also for healthcare organizations. Companies need their products (e.g., flowers and food) to be fresh and of high-quality, while healthcare organizations need effective and reliable medicines, vaccines, and blood products.

The need of a cold chain is an important aspect to consider in perishable distribution, since inappropriate refrigeration can result in waste. According to Nahmias (1982), perishability has two categories: fixed lifetime and random lifetime. Fixed lifetime has a known lifetime, after which the product cannot be used. Examples are calendars and electronics, being more related to obsolescence than perishability (Coelho and Laporte, 2014). Medicine is often considered to have a fixed lifetime, although some research shows that sometimes it can be used after the defined lifetime and that the efficacy degrades over time (Kellicker, 2006; Balaton et al., 1999). Products with random lifetime have an unknown lifetime that usually follows a distribution with an exponential decay. The quality of these products or customer value diminishes over time. Examples are fruits and vegetables (Coelho and Laporte, 2014).

Wang et al. (2012) explain that the transportation of perishable products, such as refrigerated food, differs from transportation of generic products because its quality is "vulnerable to time, distance, modes of transportation and other transportation conditions," such as temperature maintenance. According to Crama et al. (2018), 52% of sales revenue in grocery retail comes from perishable products and 10% of these products are lost before sale.

In addition, the challenge of transporting perishable products varies according to the region. In big cities, where demand is growing rapidly, the main uncertainty for distribution is transit time, which varies according to the time of day and can be affected by road infrastructure. In rural regions and low income countries, other challenges and uncertainties are also important, especially when we consider distribution of perishable vaccines and medicines. In this case, not only the demand is uncertain due to lack of accurate population estimates but also road accessibility is uncertain due to weather conditions and road infrastructure. Moreover, the number of disasters and affected populations have been growing, which increases the demand for blood products (Nagurney et al., 2012) and other perishable medical supplies.

The distribution of perishable goods can be thought of as a vehicle routing problem (VRP). Usually, a VRP takes as input vehicle capacity, transit times, and total time limit. These problems are, in general, NP-hard and are difficult to solve for instances with more than 50 customers (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012). Much of the research on VRP focuses on fast and efficient solution techniques (exact and heuristic).

In this chapter, we do not address solution methods. Instead, we focus on optimization models to aid in distribution of perishable goods under uncertainty.

In the literature of VRP for perishable products, many authors focus on improving customer satisfaction as a goal. Amorim and Almada-Lobo (2014) investigate VRP with time windows (VRPTW) considering the perishable characteristics of the product to be transported. Their objective is to minimize the total route cost and maximize the *average* freshness of all requests. According to them, freshness is the deterioration of shelf-life, considering the time, after leaving the depot. For large instances, they apply a hybrid multi-objective genetic algorithm to address the problem. Osvald and Stirn (2008) also focus on freshness. However, instead of maximizing freshness, they minimize the cost that includes loss of quality. According to them, loss of quality is related to food degradation that gradually affects its selling price, until it has to be discarded. They measure this loss as a function of load size and time and modeled fresh food distribution in Slovenia as a vehicle routing problem with time windows and time-dependent travel times. Song and Ko (2016) present a nonlinear mathematical model and a heuristic algorithm for vehicle routing, also aiming to maximize customer satisfaction (product freshness), related to the time to deliver the product from a depot. They consider both refrigerated and non-refrigerated vehicles.

Wang et al. (2014) also consider a heterogeneous fleet, as in Song and Ko (2016). They model the problem as an integer model with objective to minimize the total cost of distribution, considering time windows. The importance of a heterogeneous fleet is discussed by Hsu et al. (2007), since refrigerated vehicles with cold storage equipment are more expensive and consume more fuel than regular vehicles. In our research we consider a heterogeneous fleet with different capacities for cold storage and vehicle characteristics. For example, a motorcycle may be able to travel on dirt roads but may only have a small capacity for a "cold box" with dry ice. We also consider a truck that can only travel on paved roads but may have room to carry refrigerated and non-refrigerated products.

Uncertainties are often present in VRPs. When they arise, the problem is called a stochastic vehicle routing problem (SVRP). Ritzinger et al. (2016) provides a comprehensive survey of SVRP. The most common type of uncertainties considered in SVRPs are demand and transit time, where demand is the most studied (Gendreau et al., 1996; Gounaris et al., 2013). One of the earliest papers on SVRP with probabilistic demand is Tillman (1969). Gao and Chabini (2006) note that different factors cause variability in travel time, such as congested traffic and disturbances in the traffic network (e.g., vehicle breakdown, work zones, and bad weather). Uncertain demand and time are typically represented as random variables, with a probability distribution, or with scenarios. Fixed uncertainty intervals are also used to capture a range of uncertain values. Different strategies are used to solve SVRP, such as probabilities of satisfying a constraint (Dror and Trudeau, 1986), penalties if a constraint is not satisfied (Lee et al., 2012; Dror and Trudeau, 1986), and probabilistic scenarios (Crama et al., 2018; Petroianu et al., 2019; Sahinidis, 2004). Some authors incorporate uncertainty in perishable products distribution. Hsu et al. (2007) considers time dependent travel within the time window constraint. The model is an SVRP with time windows, minimizing costs and considering penalties for deliveries outside time windows. They consider traffic conditions that are unpredictable and directly affect the product. Chen et al. (2009) analyze production scheduling and vehicle routing with time windows under stochastic demands. They maximize the expected total profit of the supplier, applying penalties for delivering out of the time window. They model the problem as an integer nonlinear program and later decompose the variables into two groups; where the first group defines a production scheduling optimization problem, and the second models a VRP with time windows (VRPTW).

In this chapter, we consider uncertainty of demand, transit time, and freshness/quality and spoilage of perishable goods.

Three common ways to model uncertainty in optimization are with chance constraints, robust optimization, and stochastic programming with recourse. Chance constraints use a threshold for failure and a probability of satisfying this threshold. Robust optimization is also commonly used to handle uncertainty. According to Gorissen et al. (2015), robust optimization is a young field that has been developed in the last 20 years with many potential applications in real-life situations. However, its potential has not yet been fully exploited. Stochastic programming with recourse is well-studied with many applications (Birge and Louveaux, 2011). In stochastic programming with recourse, the initial stage considers known information to make an initial decision, and the recourse determines a response to uncertainty.

Dror and Trudeau (1986) and Laporte (1992) model SVRP with stochastic demand or travel time using chance constraints. In Dror and Trudeau (1986), the demand is a random variable and they stipulate a probability of route failure due to exceeding the vehicle capacity. Laporte (1992) present an SVRP in which travel and service times are random variables. In their model, the probability that a route exceeds a predetermined duration should not surpass a threshold.

Robust optimization also handles demand and time uncertainties (Gounaris et al., 2013; Sungur et al., 2008; Ben-Tal et al., 2011; Lee et al., 2012). Gounaris et al. (2013) and Sungur et al. (2008) present a robust optimization approach with intervals of uncertainty for demand and vehicle capacities. Ben-Tal et al. (2011) model a humanitarian relief supply chain using robust optimization. They consider emergency response and evacuation with uncertainty of outgoing demand and time dependency. Lee et al. (2012) present a robust optimization model to solve an SVRP with time limits for customer and vehicles, and uncertainty of demand and travel times. Their objective is to minimize the sum of all routing distances. Lee et al. (2012) assume that the probabilistic distribution of demand and travel time in unknown, but the nominal and maximum deviation are specified and used to define the values of the robust intervals of uncertainty. They optimize the distribution using column generation, limiting the number of routes with time delay.

Stochastic programming with recourse, considering demand and transit time as random variables is another method for incorporating uncertainty (Dror and Trudeau, 1986; Zhang et al., 2016; Gendreau et al., 1996; Laporte and Louveaux, 2002; Laporte, 1992; Güner et al., 2012). In Dror and Trudeau (1986) the SVRP with uncertainty of demand is not only modeled as a chance constraint, but also as a stochastic program with recourse. In their model, Dror and Trudeau (1986) apply a penalty in the recourse part if the vehicle capacity constraint fails. Zhang et al. (2016) present a formulation in which there are probability distributions for the demand, and the possibility of exceeding the vehicle capacity triggers recourse actions, such as an additional route for the vehicle. Zhang et al. (2016) present three models with different objective functions. The objectives considered are minimization of total cost, maximization of on-time deliveries, and minimization of total cost subject to the assurance of a given level of service. Gendreau et al. (1996) exemplify a stochastic program with recourse having two stages. The first stage defines the routes, while the second defines the quantities distributed. Laporte and Louveaux (2002) solve a capacitated SVRP using a two-stage stochastic program in which the first stage plans the routes while the recourse stage considers return trips in case of failure to meet demand. Laporte (1992) modeled an SVRP with stochastic service and travel times using chance constraints and stochastic programming with recourse. In their two-stage model, the first stage defines the number of vehicles and routes, while the second stage considers two random variables: travel time and service time. Moreover, Laporte (1992) apply penalties for excess time. They present two stochastic programming models, one where all vehicles are similar and another where vehicles are different and should also be assigned to do the trip. Güner et al. (2012) aims to minimize travel time through stochastic dynamic optimization, affirming that the standard approach for dealing with traffic congestion is to add a time buffer into the trip. For example, during peak times the buffer could contribute around 40% of additional time.

In this chapter, we compare and contrast how robust optimization and stochastic programming with recourse incorporate uncertainty.

An important factor to consider in modeling VRP or SVRP is the presence of multiple objectives. According to Jozefowiez et al. (2008), common goals in VRP include minimization of total transit time, total time required, total cost, and fleet size, or maximization of quality of service and profit. When more than one of these objectives is important, a multi-objective approach should be used. A common approach to a multi-objective problem is to use weights and construct a single weighted objective function. Another approach is to explore the Pareto set of non-dominated solutions and construct an efficient frontier that provides trade-off information among objectives. Jozefowiez et al. (2008) summarize three approaches to multi-objective problems:

- *a priori* approach – the objective preference is given as a hierarchy or with weights,
- *interactive* approach – the choice of objective or weights is done during the solution process,

- *a posteriori* approach – non-dominated solutions are created and a set of best solutions is provided.

Incorporating uncertainty into a multi-objective problems has additional challenges. Ide and Schöbel (2016) present an extensive survey of robust optimization with multiple objectives and uncertainty. The robust optimization for a single objective considers predefined uncertainty sets or scenarios. Those can be "strictly" robust, in which the solution has to be feasible in all scenarios, or "lightly" robust, in which the solution is feasible in the most likely scenario. Considering multiple objectives, a solution is "partially" robust if it is non-dominated for at least one scenario (Ide and Schöbel, 2016). Ide and Schöbel (2016) present different concepts and manners to evaluate solution robustness when the feasibility set does not change considering different scenarios.

In this chapter, we apply a modification of min-max robust optimization, in which the objective is to minimize "the worst case of the objective function under all possible scenarios" (Ide and Schöbel, 2016), considering changes in the feasibility sets for each scenario. Since an extreme scenario may not be feasible with the available resources, we apply stochastic programming with recourse to provide a way to handle all scenarios. We also construct an efficient frontier using robust optimization and stochastic programming with recourse for comparison purposes.

We model an SVRP for perishable products delivery using expected value as a base case, min-max robust optimization utilizing the *a posteriori* approach, and stochastic programming with recourse. We compare the results between the different approaches and discuss the different insights provided. We consider uncertainties of both demand and transit time, plus the uncertainty associated with a cold chain. Moreover, distinct from most previous work, we do not consider a single objective of cost reduction or time minimization. We consider three objective functions, i.e., minimization of transportation time, minimization of loss of quality, and minimization of potential spoilage. The third objective function is analyzed using penalties for utilization of unreliable roads and vehicles (Hamedi et al., 2012;

List et al., 1991; Nolz et al., 2011).

By considering multiple objectives, we provide insights as to the trade-offs between objectives, providing an efficient frontier with non-dominated solutions.

The three models provide different types of insights, including the impact of uncertainty and possible recourse actions. This aids in designing an effective routing plan for distribution of perishable products and medical supplies.

4.2 Model Formulations

As mentioned in Section 4.1, the optimization models in this chapter consider three objectives:

- Minimize expected transit/transportation time,
- Minimize expected penalties for possibility of spoilage,
- Minimize loss of quality.

The first objective, minimize total transit time, is common in SVRPs. The second objective is used to penalize the possibility of spoilage due to unreliable vehicles and poor roads. To simplify the model and numerical analyses, we merge the first and second objective functions (minimize transit time plus penalties) into a single weighted objective function, since they are both directly related to the route chosen. The Pareto analysis with two objectives allows the decision maker to see a trade-off between transit time plus penalties versus a loss of quality objective, as is discussed later.

4.2.1 Baseline - Expected Value Model

The first model is used as a baseline for comparison, and it includes the expected values for all uncertain parameters. The decision variables include routing information and quantity carried in each vehicle, along with timing information. Table 4.1 presents notation used in the expected value model, including sets, decision variables and parameters. In the baseline model, the expected value of the uncertain parameters is identified with a superscript μ, which is later replaced by θ to denote scenario.

In this chapter we assume that there is only one supply node as the depot, denoted o, although the number of suppliers can be easily extended. The expected demand is denoted d^{μ}_{ip} for each center i and product p. The expected transit time is h^{μ}_{ijv} between two centers i and j using vehicle v. Each vehicle v has two different types of capacity, c^r_v for products that need refrigeration and c^d_v for the remaining products, where $c^t_v = c^r_v + c^d_v$ is the total capacity (refrigerated and non-refrigerated).

To minimize the possibility of spoilage of products, we define a penalty γ_{ijv}, based on the penalty of the arc, α_{ij}, and the penalty of the vehicle, β_v, chosen, that is, $\gamma_{ijv} = (\alpha_{ij} + \beta_v)/2$.

This penalty is greater for roads in bad condition and/or older vehicles. We assume that choosing better roads and reliable vehicles will decrease the chance of breakdowns or delays that lead to longer transit time than the product's cold life, resulting in spoilage.

To weigh the combination of transit time and penalties related to spoilage, both objectives are normalized by dividing the expected transit time h^{μ}_{ijv} and expected penalty γ^{μ}_{ijv} by their respective averages $\mu_{h^{\mu}}$ and μ_{γ} over all possible values. Specifically, $\mu_{h^{\mu}} = \sum_{v \in V} \sum_{i,j \in C} 2h^{\mu}_{ijv}/|V|\ |C|^2$, and similarly, $\mu_{\gamma} = \sum_{v \in V} \sum_{i,j \in C} \gamma^{\mu}_{ijv}/|V|\ |C|^2$.

We define a *delivery freshness time* s^{μ}_{iv}, given in (4.1), so that centers requesting products with a small freshness time, should receive a delivery early in a route. The delivery freshness time is defined for each center $i \in C$ and vehicle $v \in V$. This loss of freshness results in loss of product quality. To capture loss of freshness, we define a *product freshness time* f_{vp} that guarantees each product $p \in P_r$ will remain fresh if it is delivered by vehicle v before time f_{vp}. For example, if v has no refrigeration capability, a product p (e.g., ice cream) should be delivered within 30 minutes, whereas if v has refrigeration, the delivery time could be much longer. The product freshness time is calculated as the sum of the product freshness time without proper refrigeration g_p, and the vehicle refrigeration time z_v, that is, $f_{vp} = z_v + g_p$.

Table 4.1: Notation for sets, decision variables and parameters.

Sets	
Centers - C	$i \in C$, o is the supply node (i.e., depot)
Vehicles - V	$v \in V$
Refrigerated products - P_r	$p \in P_r$
Non-refrigerated (dry) products - P_d	$p \in P_d$
Products - $P = P_r \cup P_d$	$p \in P$
Decision variables	
y_{ijv}	binary variable; equals 1 if products are transported from i to j using vehicle v, and equals 0 otherwise
x_{ijvp}	quantity of product p transported from i to j using vehicle v
t_{iv}	time that vehicle v leaves center i
Parameters	
s_{iv}^{μ}	expected delivery freshness time for making a delivery to center i with vehicle v
d_{ip}^{μ}	expected demand at center i for product p
h_{ijv}^{μ}	expected transit time between i and j using vehicle v
$\mu_{h^{\mu}}$	average time considering all h_{ijv}^{μ}
c_v^r	capacity of vehicle v to carry products needing refrigeration
c_v^d	capacity of vehicle v to carry non-refrigerated (dry) products
c_v^t	total transportation capacity of vehicle v, $c_v^t = c_v^r + c_v^d$
u	maximum time for a route (e.g., 8 hours)
γ_{ijv}	penalty for driving between centers i and j with vehicle v
μ_{γ}	average time considering all γ_{ijv}
w	time for product drop off
M	large number, equals 600 (10 hours)

To calculate an average delivery freshness time s^{μ}_{iv} for making a delivery to center i utilizing vehicle v, we evaluate the minimum freshness time f_{vp} for all products demanded by center i. We also set an upper bound on delivery freshness time to be the limit on the total route time, denoted u. Thus, we have,

$$s^{\mu}_{iv} = \min[\min_{p \in P} f_{vp}, u] \quad \forall i \in C, v \in V. \tag{4.1}$$

The expected value model used as a baseline is given in (4.2)-(4.14).

$$\min \sum_{i \in C} \sum_{j \in C} \sum_{v \in V} (h^{\mu}_{ijv}/\mu_{h^{\mu}} + \gamma_{ijv}/\mu_{\gamma}) y_{ijv} \tag{4.2}$$

$$\min \sum_{i \in C} \sum_{v \in V} (1/s^{\mu}_{iv}) * t_{iv} \tag{4.3}$$

subject to

$$\sum_{i \in C \backslash \{l\}} \sum_{l \in C \backslash \{o\}} \sum_{v \in V} (x_{ijvp} - x_{jlvp}) = d^{\mu}_{jp} \quad \forall j \in C \backslash \{o\}, p \in P \tag{4.4}$$

$$\sum_{p \in P_r} x_{ijvp} \leq c^{r}_{v} \quad \forall i \in C, j \in C, v \in V \tag{4.5}$$

$$\sum_{p \in P} x_{ijvp} \leq c^{t}_{v} \quad \forall i \in C, j \in C, v \in V \tag{4.6}$$

$$t_{jv} - t_{iv} + M(1 - y_{ijv}) \geq h^{\mu}_{ijv} + w \quad \forall j \in C \backslash \{o\}, i \in C \backslash \{j\}, v \in V \tag{4.7}$$

$$t_{iv} + w \leq s^{\mu}_{iv} \quad \forall i \in C \backslash \{o\}, v \in V \tag{4.8}$$

$$t_{iv} + h^{\mu}_{iov} + w \leq u \quad \forall i \in C \backslash \{o\}, v \in V \tag{4.9}$$

$$\sum_{i \in C \backslash \{j\}} (y_{ijv} - y_{jiv}) = 0 \quad \forall j \in C, v \in V \tag{4.10}$$

$$y_{ijv} - x_{ijvp} \geq 0 \quad \forall i \in C, j \in C, v \in V, p \in P \tag{4.11}$$

$$\sum_{j \in C \backslash \{o\}} y_{ojv} \leq 1 \quad \forall v \in V \tag{4.12}$$

$$t_{ov} = 0 \quad v \in V, \theta \in \Theta \tag{4.13}$$

$$y_{iiv} = 0 \quad y_{ijv} \in \{0, 1\} \quad x_{ijvp} \geq 0 \quad t_{iv} \geq 0 \quad a_{iv} \geq 0 \quad \forall i \in C, j \in C, v \in V, p \in P. \tag{4.14}$$

The objective function in (4.2) minimizes the expected transit time and the penalties for wasted products during transit. The objective function in (4.3) minimizes a measure of loss of quality or loss of freshness, represented by the time taken to deliver the products within their delivery freshness time and scaled by $1/s^{\mu}_{iv}$.

Constraints (4.4) ensure that the demand in each center is met. The constraints in (4.5) and (4.6) ensure that a vehicle cannot transport more products than its cold and total capacities respectively. According to the constraints in (4.7), the time sequence between two sequential centers must be obeyed. This means that if center j follows i, the time that vehicle v leaves j has to be greater than the time that it passed by i plus the transit time between the centers and the drop off time. The constraints in (4.8) ensure that the delivery time at center i must respect the delivery freshness time using vehicle v. The constraints in (4.9) guarantee that the vehicle v respects the maximum time for a route u. The constraints in (4.10) ensure that if a vehicle enters a center, it has to leave it. Moreover, the constraints in (4.11) guarantee that if there are products being transported between centers, there will exist a route between them. The constraints in (4.12) limit one route per vehicle and the constraints in (4.13) define the initial time as zero. The constraints in (4.14) remove the possibility of a vehicle returning to the same center after leaving it by setting y_{iiv} to zero, and define the variables y_{ijv} to be binary, and the variables x_{ijvp} and t_{iv} to be non-negative.

4.2.2 Robust Optimization Model

Our robust optimization model applies a modified version of the min-max robust optimization, in which the objective is to minimize “the worst case of the objective function under all possible scenarios” (Ide and Schöbel, 2016) and different feasibility sets.

We consider three uncertainties: transit time h_{ijv}, demand d_{ip}, and delivery freshness time s_{iv}. Transit time and demand are common uncertainties in the literature, as discussed in Section 4.1, therefore, we could use probability distributions considered by other authors.

The transit time has a lognormal distribution with standard deviation equal to one and given mean values. This distribution was also presented in Oyola et al. (2018) and Wang et al. (2012). The demand has a triangular distribution as in Ramaekers and Janssens (2008); Schmitt et al. (2010); Yang et al. (2000) and we assumed the delivery freshness time has a uniform distribution. Their mean values are presented in Appendix 4.3.

To construct scenarios, we consider three values for transit time, called *min*, *mean*, and *max*. We only consider two values for demand, mean and max, because the optimal solution was not sensitive to the minimum value. Similarly, we only consider two values for delivery freshness time, min and mean, which were the more conservative values. This results in 12 scenarios. Table 4.2 represents the 12 scenarios evaluated. More specifically, the 12 scenarios combine the following values for each uncertain value:

- transit time h_{ijv} – minimum value equals 40% of the cumulative probability distribution, mean value, and maximum value equals 60% of the cumulative probability distribution;

- demand d_{ip} – mean value, and the maximum demand (according to the triangular distribution) equals 2.5×mean value;

- delivery freshness time s_{iv} – minimum value equals 80% of the mean value, and mean value.

As shown in Table 4.2, Scenario 6 is equivalent to the baseline and Scenario 11 is the most conservative, considering the maximum transit time, maximum demand, and minimum delivery freshness time.

4.2.3 Chance Constraint Model

A chance constraint model that uses the probability of satisfying constraints can be viewed as one of the scenarios in robust optimization, where each scenario gives a threshold to a

Table 4.2: Scenarios evaluated in the robust optimization model. Scenario 6 represents the expected value scenario used in the baseline, and Scenario 11 represents the most conservative one.

Scenario θ	Transit Time (h_{ijv}^{θ})	Demand (d_{ip}^{θ})	Delivery Freshness Time (s_{iv}^{θ})
Scenario 1	min	mean	min
Scenario 2	min	mean	mean
Scenario 3	min	max	min
Scenario 4	min	max	mean
Scenario 5	mean	mean	min
Scenario 6	**mean**	**mean**	**mean**
Scenario 7	mean	max	min
Scenario 8	mean	max	mean
Scenario 9	max	mean	min
Scenario 10	max	mean	mean
Scenario 11	**max**	**max**	**min**
Scenario 12	max	max	mean

probability for the uncertain parameters. For example, considering constraints in (4.8), if scenario θ considers the minimum value, then the probability of satisfying this constraint and delivering the products within their freshness time exceeds 0.80, i.e.,

$$P(t_{iv} + w \leq S_{iv}) \geq 0.80 \quad \forall i \in C \backslash \{o\}, v \in V.$$

Constraints in (4.9) represent another example. If the scenario θ considers the maximum value of the transit time, i.e., most conservative value for transit time, the probability of satisfying the constraint and finishing the route within the maximum time exceeds 0.60.

Therefore, the different robust scenarios can be considered equivalent to different probabilistic constraints with threshold probabilities.

4.2.4 Stochastic Programming with Recourse Model

The stochastic programming model considers the assignment of routes and vehicles as first stage decision variables. The recourse consists of assigning extra vehicles for the delivery, with quantity delivered and time for leaving each center, in each scenario. Table 4.3 presents the additional notation needed for the stochastic program.

$$\min \sum_{i\in C}\sum_{j\in C}\sum_{v\in V\cup V'}\sum_{\theta\in\Theta} p^{\theta}(h^{\theta}_{ijv}/\mu_{h^{\theta}} + \gamma_{ijv}/\mu_{\gamma})y_{ijv} + p^{\theta}(h^{\theta}_{ijv'}/\mu_{h^{\theta}} + \gamma_{ijv'}/\mu_{\gamma})ry^{\theta}_{ijv'} \tag{4.15}$$

$$\min \sum_{i\in C}\sum_{v\in V\cup V'}\sum_{\theta\in\Theta} p^{\theta}(1/s^{\theta}_{iv})(t^{\theta}_{iv}) \tag{4.16}$$

subject to

$$\sum_{i\in C\backslash\{l\}}\sum_{l\in C\backslash\{o\}}\sum_{v\in V\cup V'} (x^{\theta}_{ijvp} - x^{\theta}_{jlvp}) = d^{\theta}_{jp} \quad \forall j \in C\backslash\{o\}, p \in P, \theta \in \Theta \tag{4.17}$$

$$\sum_{p\in P_r} x^{\theta}_{ijvp} \le c^{r}_{v} \quad \forall i \in C, j \in C, v \in V \cup V', \theta \in \Theta \tag{4.18}$$

$$\sum_{p\in P} x^{\theta}_{ijvp} \le c^{t}_{v} \quad \forall i \in C, j \in C, v \in V \cup V', \theta \in \Theta \tag{4.19}$$

$$t^{\theta}_{jv} - t^{\theta}_{iv} + M(1 - y_{ijv}) \ge h^{\theta}_{ijv} + w \quad \forall j \in C\backslash\{o\}, i \in C\backslash\{j\}, v \in V, \theta \in \Theta \tag{4.20}$$

$$t^{\theta}_{jv'} - t^{\theta}_{iv'} + M(1 - y^{\theta}_{ijv'}) \ge h^{\theta}_{ijv'} + w \quad \forall j \in C\backslash\{o\}, i \in C\backslash\{j\}, v' \in V', \theta \in \Theta \tag{4.21}$$

$$t^{\theta}_{iv} + w \le s^{\theta}_{iv} \quad \forall i \in C\backslash\{o\}, v \in V \cup V', \theta \in \Theta \tag{4.22}$$

$$t^{\theta}_{iv'} + h^{\theta}_{iov'} + w \le u \quad \forall i \in C\backslash\{o\}, v' \in V \cup V', \theta \in \Theta \tag{4.23}$$

$$\sum_{i \in C \setminus \{j\}} (y_{ijv} - y_{jiv}) = 0 \quad \forall j \in C, v \in V \tag{4.24}$$

$$\sum_{i \in C \setminus \{j\}} (y^{\theta}_{ijv'} - y^{\theta}_{jiv'}) = 0 \quad \forall j \in C, v' \in V', \theta \in \Theta \tag{4.25}$$

$$y_{ijv} - x^{\theta}_{ijvp} \geq 0 \quad \forall i \in C, j \in C, v \in V, p \in P, \theta \in \Theta \tag{4.26}$$

$$y^{\theta}_{ijv'} - x^{\theta}_{ijv'p} \geq 0 \quad \forall i \in C, j \in C, v' \in V', p \in P, \theta \in \Theta \tag{4.27}$$

$$\sum_{j \in C \setminus \{o\}} y^{\theta}_{ojv'} \leq 1 \quad \forall v' \in V', \theta \in \Theta \tag{4.28}$$

$$\sum_{j \in C \setminus \{o\}} y_{ojv} \leq 1 \quad \forall v \in V \tag{4.29}$$

$$t^{\theta}_{ov} = 0 \quad v \in V \cup V', \theta \in \Theta \tag{4.30}$$

$$\begin{gathered} y_{iiv} = 0 \quad y_{ijv} \in \{0,1\} \quad y^{\theta}_{iiv} = 0 \quad y^{\theta}_{ijv} \in \{0,1\} \quad x^{\theta}_{ijvp} \geq 0 \\ t^{\theta}_{iv} \geq 0 \quad a^{\theta}_{iv} \geq 0 \quad \forall i \in C, j \in C, v \in V \cup V', p \in P \end{gathered} \tag{4.31}$$

The objective function (4.15) minimizes the expected transit time and the penalties for wasted products during transit. The objective function (4.16) minimizes the expected time taken to deliver the products considering their freshness time.

Constraints (4.17) ensure that the demand in each center is met. Constraints (4.18) and (4.19) ensure that a vehicle cannot transport more products than its cold and total capacities respectively. According to constraints (4.20) and (4.21), the time sequence between two sequential centers must be obeyed. This means that if center j follows i, the time that vehicle v leaves j has to be greater than the time that it passed by i plus the transit time

Table 4.3: Additional notation for sets, decision variables, and parameters for the stochastic programming model.

Sets	
Scenarios - Θ	$\theta \in \Theta$
Recourse vehicles (similar to V) - V'	$v \in V'$
Decision variables	
y_{ijv}	first stage binary variable; equals 1 if products are transported from i to j using vehicle $v \in V$; and equals 0 otherwise.
x^{θ}_{ijvp}	quantity of product p transported from i to j using vehicle $v \in V \cup V'$ for scenario θ.
t^{θ}_{iv}	time that vehicle $v \in V \cup V'$ leaves center i for scenario θ.
$y^{\theta}_{ijv'}$	binary variable for scenario θ; equals 1 if products are transported in the recourse from i to j using vehicle $v' \in V'$; and equals 0 otherwise.
Parameters	
s^{θ}_{iv}	time for worst freshness product in center i demand with vehicle v for scenario θ.
d^{θ}_{ip}	demand at center i for product p for scenario θ.
h^{θ}_{ijv}	transit time between i and j using vehicle v for scenario θ.
p^{θ}	probability of scenario θ.
r	cost of using an extra vehicle in the recourse, equals 100.

between the centers and the drop off time. The constraints (4.22) define that the time of the product with lower freshness period has to be respected while delivering the products to a center i. Constraints (4.23) guarantee that the vehicle v respects the maximum time for a route u. Constraints (4.24) and (4.25) ensure that if a vehicle enters a center, it has to leave it. Moreover, constraints (4.26) and (4.27) guarantee that if there are products being transported between centers, there will exist a route between them. Constraints (4.28) and (4.29) ensure only one route per vehicle. Constraints (4.30) set the initial time of a vehicle zero. Constraints (4.31) remove the possibility of a vehicle returning to the same center after leaving it and define the binary variables y_{ijv} and y^{θ}_{ijv}, and non-negative variables x^{θ}_{ijvp} and t^{θ}_{iv}.

4.2.5 Numerical results

Baseline

We prepared a test data set to illustrate the model. The expected transit times, h^{μ}_{ijv}, were calculated by multiplying the distances of the first 10 centers in the instance belgium-road-km-n50-k10 (Smet, 2017) by 0.5. We consider three vehicles to calculate the routes. Their characteristics and other parameters are described in Appendix 4.3.

Figure 4.1 illustrates a Pareto analysis of the expected value model (baseline). The vertical y-axis gives the objective function values of the transit time plus penalties and the horizontal x-axis the objective function values of loss of freshness. The line represents the efficient frontier for the two objective functions and the red dots represent non-dominated solutions. For example, if loss of freshness is constrained to be below 4.10, the best value for transit time plus penalties has a value of 16.20. On the other hand, if loss of freshness is allowed to increase to 4.60, transit time plus penalties will decrease significantly to below 15.40. This Pareto analysis shows that an intermediary solution between transit time and penalties could be more interesting than minimizing either objective function alone, since both increase at high rates.

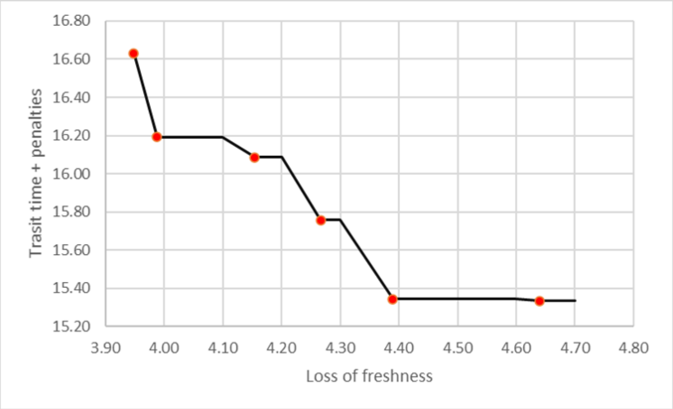


Figure 4.1: Comparison of objective functions on the baseline expected value test problem. The line represents the efficient frontier of the Pareto analysis and the red dots denote calculated solutions that are non-dominated.

Robust optimization

We present numerical results using the same test data set presented in Appendix 4.3, with the first 10 centers in the instance belgium-road-km-n50-k10 (Smet, 2017) and three vehicles.

For the first analysis, we consider a total route time of 10 hours and each of the three vehicles has a refrigeration time, z_v, of 6, 3, and 2 hours respectively, reflecting their refrigeration capabilities. Since 50% of the original 12 scenarios were infeasible (scenarios 5, 7, 9, 10, 11, 12), we modified the total route time to 12 hours. Now scenarios 10 and 12 became feasible. However, scenarios 5, 7, 9, and 11 were still infeasible. Therefore, we increased the refrigeration time in the last vehicle, z_3 from 2 to 3 hours. The new results are in Table 4.4, and only scenarios 9 and 11 are infeasible.

As shown in Table 4.4, the pairs of scenarios, 1 and 3, 2 and 4, 5 and 7, 6 and 8, and 10 and 12 have the same objective function results. In each pair of scenarios, the value of demand is *mean* and *max*, and the other values are the same, indicating that demand is not affecting the routes. This is because the carrying capacity of the vehicles is not restricting the routes. The limitations for delivering products within their freshness time and the total route time is creating multiple routes carrying smaller quantities of products. If those time constraints were not considered, then demand would affect the routes.

Table 4.4: Robust optimization model results, considering total route time of 12 hours, and vehicle refrigeration time of 6, 3, and 3 hours. Scenario 6 represents the expected value scenario and Scenario 11 represents the most conservative one.

Scenario θ	Minimize objective 1 first		Minimize objective 2 first	
	Objective 1	Objective 2	Objective 1	Objective 2
Scenario 1	14.78	4.70	16.63	3.62
Scenario 2	13.77	4.90	16.63	3.48
Scenario 3	14.78	4.70	16.63	3.62
Scenario 4	13.77	4.90	16.63	3.48
Scenario 5	15.61	4.97	16.63	4.12
Scenario 6	**15.34**	**4.64**	**16.63**	**3.95**
Scenario 7	15.61	4.97	16.63	4.12
Scenario 8	15.34	4.64	16.63	3.95
Scenario 9	infeasible			
Scenario 10	16.18	5.08	16.63	4.56
Scenario 11	**infeasible**			
Scenario 12	16.18	5.08	16.63	4.56

Since the worst or most conservative scenarios (Scenarios 9 and 11 with maximum transit time and minimum freshness time) remain infeasible, we use a modified version of the min-max approach presented by Ide and Schöbel (2016). In their paper, they assume that the feasibility set remains the same for all scenarios, which is not the case in this chapter, since the difference in transit time and freshness time modify the feasibility sets in each scenario. Since demand did not affect results, we know that maximum transit time and minimum freshness time are the most conservative values. However, these scenarios, 9 and 11 are infeasible, so no scenario is completely robust.

As in Ide and Schöbel (2016), we consider a lightly robust scenario as one whose solution is feasible for almost all scenarios. Here, the pairs of Scenarios 10 and 12, and 5 and 7, are considered lightly robust. Scenarios 10 and 12 consider maximum transit time and mean freshness time. Therefore, if it considers a route feasible, this route will also be feasible for Scenarios 2 and 4, and 6 and 8. Scenarios 5 and 7 consider mean transit time and minimum freshness time. Therefore, if it considers a route feasible, this route will also be feasible for Scenarios 1 and 3, and 6 and 8. Figure 4.2 provides the results for all feasible scenarios, and the black boxes highlight the lightly robust scenarios.

In Figure 4.2, blue circles represent the minimization of transit time plus penalties first. Red squares are used when loss of freshness is minimized first. The lines are used to identify the best values for the lightly robust scenarios.

The solutions associated with Scenarios 5 and 7 dominate the solutions associated with Scenarios 10 and 12, therefore, Scenarios 5 and 7 would be the best ones to use for planning purposes. An efficient frontier constructed using Scenarios 5 and 7 provides trade-offs while being as conservative as possible without becoming infeasible.

Stochastic programming with recourse

As discussed earlier, the most conservative scenario is not feasible in the robust approach. Therefore, we used the stochastic program with recourse model (Section 4.2.4) to account for all scenarios, considering the recourse action of using more vehicles to guarantee feasibility under all scenarios. Knowing that some scenarios give the same result, we set demand to the maximum value and consider the scenarios with probabilities in Table 4.5. Scenario S3 is equivalent to the baseline model and Scenario S4 is the most conservative.

In both Figures 4.3 and 4.4, there is only one recourse vehicle, and the cost of the recourse vehicle affects which center it visits. If $r = 2$, the recourse vehicle visits center $C2$, and when $r = 100$, the recourse vehicle visits $C10$, which is closer and thus incurs less cost. The type of

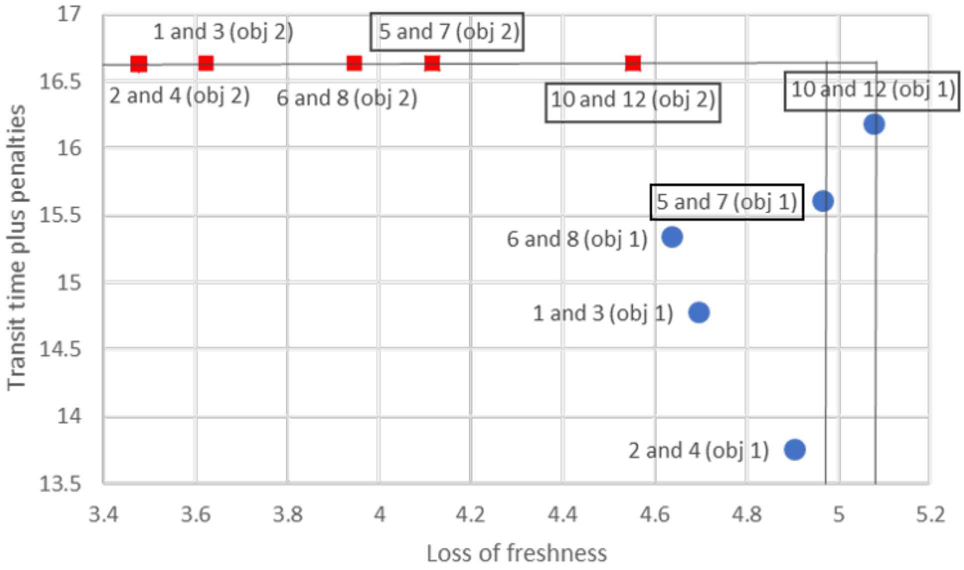


Figure 4.2: Objective function values for different scenarios. Blue circles represent the minimization of transit time plus penalties first. Red squares are used when loss of freshness is minimized first. Black boxes highlight the lightly robust scenarios.

recourse vehicle also changes by scenario depending on the freshness time and refrigeration capability.

Solving a stochastic program with recourse requires more computation time than solving the robust model. Also, we could only minimize the transit time plus penalties within a reasonable time; due to the characteristics of the data and model, minimizing loss of freshness could not be solved in a reasonable time.

4.3 Discussion and Conclusions

This chapter presents two models to tackle SVRP with uncertainties of transit time, demand, and freshness time for perishable products: robust optimization and stochastic program with recourse. We also consider two objectives and seek Pareto optimal solutions to describe trade-offs between transit time plus penalties for waste products during transit, and loss of

Table 4.5: Scenarios evaluated in the stochastic program with recourse model. Scenario 3 is equivalent to the expected value scenario and scenario 4 represents the most conservative one.

Scenario (θ)	Transit Time (h^{θ}_{ijv})	Demand (d^{θ}_{ip})	Delivery Freshness Time (s^{θ}_{iv})	Probability (p^{θ})
S0	min	max	min	1/16
S1	min	max	mean	3/16
S2	mean	max	min	2/16
S3	**mean**	**max**	**mean**	6/16
S4	**max**	**max**	**min**	1/16
S5	max	max	mean	3/16

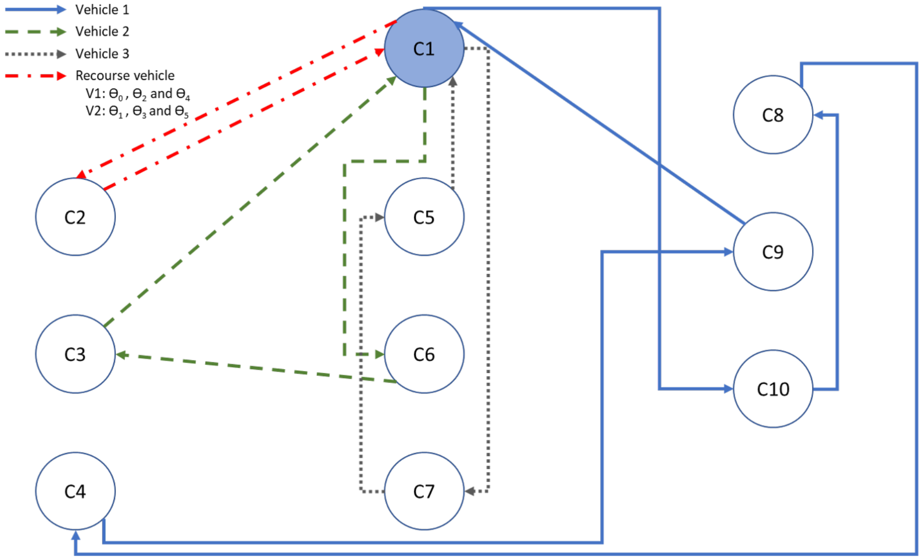


Figure 4.3: Optimal routes given cost of recourse vehicles, $r = 2$

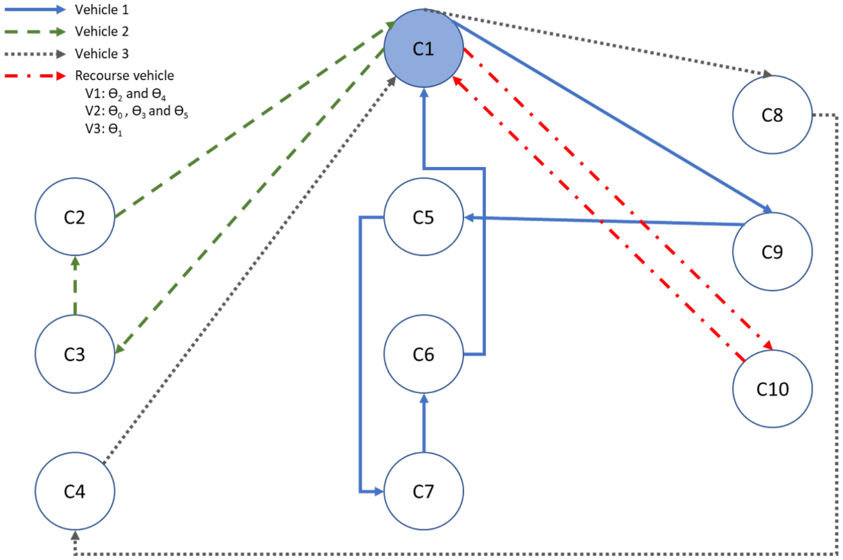


Figure 4.4: Optimal routes given cost of recourse vehicles, $r = 100$

freshness of products taking their delivery time into account.

Using a baseline model with expected values for uncertain parameters, we demonstrate the efficient frontier for a test data set in Figure 1 that provides valuable information regarding the trade-offs between transit time plus penalties and loss of freshness. This expected value scenario is useful for planning purposes when a strategic solution will be used many times. For example, allocating resources (such as vehicles) for well-established routes might find the baseline model useful.

However, when there is a large degree of uncertainty, it is important to consider several scenarios because a solution based solely on the expected value scenario will often result in an infeasible route. Therefore, the robust model provides a way to consider conservative scenarios and optimistic scenarios that account for uncertainty. In the test data set, the most conservative pair of scenarios (9 and 11) are infeasible, as shown in Table 4. The lightly robust scenarios (5 and 7, and 10 and 12) are feasible for the same number of scenarios, but are not feasible for all scenarios. As shown in Figure 2, the lightly robust Scenarios 5 and 7 considering an average transit time and a minimum freshness time provide better solutions than those from Scenarios 10 and 12, and can be used for planning purposes. The robust model, allowing for lightly robust scenarios, can augment the baseline model and provide conservative solutions along the efficient frontier.

There is still the possibility that Scenarios 9 and 11 occur, and the solutions under the robust model are infeasible. The stochastic program with recourse is an interesting option because it allows for a recourse vehicle in the case of an infeasible scenario. This model gives routes that are practical for all the scenarios, adding recourse vehicles when needed to guarantee that the problem is feasible in all scenarios. In addition, the quantity distributed and the time that the driver leaves the health centers are also scenario-based. Figures 3 and 4 provide two different solutions with different costs to illustrate the solutions with an additional vehicle that can handle Scenarios 9 and 11. A drawback to the stochastic programming model is that it requires more computational time in comparison to the other models.

In conclusion, the baseline expected model is good at giving an overall view of the efficient frontier and trade-offs between objectives. The robust model also provides trade-offs between objectives with more detail by scenario. It can also identify conservative situations when a solution is infeasible. However, the robust model cannot provide a way to compensate if there is no feasible solution for all extreme scenarios. The stochastic program with recourse provides guidance on how to handle extreme scenarios while keeping a stable (first stage) solution. It can be used for strategic planning and operational decisions, given sufficient computation time.

Appendix

Table 4.6: Penalties for roads, that are used to calculate penalty conditions $\gamma_{ijv} = (\alpha_{ij}+\beta_v)/2$

center/ center	o	2	3	4	5	6	7	8	9	10
o	0	1	1	3	3	3	2	2	1	1
2		0	1	2	2	4	4	3	1	1
3				3	3	3	3	2	2	2
4					4	4	4	2	2	2
5						1	1	2	3	3
6							2	2	1	2
7								1	2	1
8									1	1
9										1

Table 4.7: Average distance, h_{ij}, used to calculate $h^{\mu}_{ijv} = h_{ij} \cdot$ *average traveling speed*$_v$

center/ center	o	2	3	4	5	6	7	8	9	10
o	0	50.878	53.493	40.7135	37.231	37.278	51.1445	67.203	28.5835	45.7875
2		0	28.5815	91.111	56.6715	18.259	55.574	117.4965	69.0655	96.0805
3			0	89.601	44.701	38.1895	37.094	116.747	64.5635	98.7465
4				0	50.6015	77.4035	67.011	29.046	27.819	23.7845
5					0	52.8285	20.069	76.2495	24.6545	68.5605
6						0	55.9885	103.601	61.128	82.185
7							0	92.645	41.0505	84.9565
8								0	54.8975	29.3525
9									0	46.696

Table 4.8: Vehicle v parameters

	v_1	v_2	v_3
Penalty – β_v	2	2	3
Refrigeration time (hours)– z_v	6	3	3
Total storage capacity (m3) – c^t_v	2	1	1
Refrigerated storage capacity (m3) – c^r_v	2	1	1
Average traveling speed (km/hours)	40	60	60

Table 4.9: Average demand, d^{μ}_{ip}, for each center

center/product	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
o	3	2	5	8	0	10	9	5	7	0
2	4	3	10	7	0	4	4	0	6	1
3	10	5	8	5	7	0	10	10	7	6
4	4	2	10	1	4	1	2	0	5	3
5	7	7	9	0	9	5	6	7	0	10
6	6	1	9	0	5	8	2	10	7	0
7	5	6	9	8	2	0	6	6	8	8
8	5	0	9	3	3	10	10	5	5	6
9	10	10	8	10	0	9	5	5	3	8
10	9	7	2	0	10	1	3	0	7	2

Table 4.10: Freshness time, g_p, for each product without proper refrigeration, used to calculate product freshness time $f_{vp} = z_v + g_p$

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
g_p	1	1.5	2	3	4	10	10	10	10	10

Chapter 5

OPTIMAL MIDDLE MILE TRANSPORTATION WITH LOAD CANCELLATION AND PICK-UP AND DELIVERY TIME

This chapter *formulates optimization models that incorporate realistic constraints and uncertainties* to solve middle mile routing problems applied to commercial logistics problems. We develop a mixed-integer stochastic programming model of a planning and scheduling problem in middle mile transportation. Middle mile transportation consists of moving goods between facilities while satisfying several constraints, including time windows for pickup and delivery and total drive time. One challenge is that the demand is stochastic, i.e., demand can be canceled and ad-hoc demand created. Our model deals with demands that are canceled due to delays at the centers to prepare the orders. However, the truck only learns of the cancellation upon arrival. Those demands are rescheduled to later times and new routes are created to service all demands.

5.1 Introduction

Logistics is one of the main causes of expenses in industry. There is a continuous search for cost reduction, time decrease and delivery volume increase. In addition, deliveries that used to take months, now are done in days or even hours.

To guarantee that a product arrives to the final customer, the middle mile transportation has to be well planned ahead, even with uncertainty of demand. Sometimes, due to bad planning or delays, there are last minute cancellations in a route or ad-hoc demands are created. Many of the order cancellations are in fact delays in preparing the cargo. Therefore, this order needs to be postponed and rescheduled at a later time.

This chapter presents a stochastic mixed-integer programming model to describe a simplified version of the middle mile routing problem in which uncertainty of demand due to last minute cancellations and reschedules are considered. We minimize the total cost of the route, considering distance and cost of trucks for different cancellation scenarios.

5.2 Literature review

Middle-mile routing is not an area as well studied as last-mile routing, especially vehicle routing problem with full workload (Arunapuram et al., 2003). However, its importance has increased significantly due to e-commerce. Companies such as Amazon and Walmart have warehouses and distribution centers located in different parts of the United States, and distributing and allocating products between them is a common practice to balance inventory near customers and reduce the time between shopping and delivery. The products are transported between those warehouses using trucks. Usually, while routing these trucks, demand is considered as an order for a full truckload, to transport goods between two centers. These problems are typically modeled as a vehicle routing problem with time windows (VRPTW) or vehicle routing problem with pickup and delivery (VRPPD). Most of the research done in the field consider deterministic models in which the orders are known in advance. Moreover, their main goal is to minimize empty loads or lanes (Arunapuram et al., 2003; Gronalt et al., 2003; Zolfagharinia and Haughton, 2017). According to Zolfagharinia and Haughton (2017), empty loads are one of the main operational issues in the transportation industry, and load cancellations is one of the causes of this problem. In addition, few papers present multi-depot models (Liu et al., 2010). This problem is NP-hard (Liu et al., 2010). Therefore, most of the research done focuses on finding heuristic methods to solve it, especially in the presence of uncertainty (Arunapuram et al., 2003; Liu et al., 2010; Goel and Irnich, 2017). Zolfagharinia and Haughton (2016; 2017) present surveys with multiple methods to solve this VRPTW. One of the first papers to present stochastic demand for full workload transportation was Powell (1986). He presents a stochastic dynamic load assignment to deal with the demand uncertainty. Powell extends this research in Powell (1987; 1996). Berbeglia et al. (2010b) also solve the pickup and delivery problem with a dynamic routing problem, considering that inputs are revealed or updated during operation. We are unaware of the existence of papers that deal with order cancellation and rescheduling with stochastic programming models. Some authors evaluate rescheduling caused by routing disruption due to vehicle breakdown or traffic conditions (Li et al., 2009a;b; Mu et al., 2011). However, it is common for a vehicle to arrive to pick up the order, before it is ready. In this situation, the vehicle needs to wait or it cancels this order to be able to continue its route and pick up other demand. Since this canceled order needs to be attended, another vehicle should be rescheduled to pick it up. We present a new stochastic programming model for VRPTW, with load cancellation and reschedule, considering multi-depots.

5.3 Mixed integer stochastic programming model

In this section we present a simplified version of the middle mile mixed integer stochastic programming model. Since it assumes full workload trucks, there is no capacity constraint. The main constraints considered are related to time windows and total route time.

In this problem, each truck utilized has to return to its start node, and different trucks can start from different nodes. This represents a realistic situation where different starting positions can determine different trucking companies. We assume that the start node of a truck is the node where it picked up its first order. Therefore, we create two dummy nodes o and f, that have distances equal to zero from all the other nodes. These two dummy nodes are considered to be the initial and the final nodes for each route. In all scenarios, the truck must start its route at the same node. Therefore, if truck v_1 starts its route at center 1 at scenario θ_1, at scenario θ_2, it will also start its route at center 1. The time to start a route is defined as t^{θ}_{ojv}, where o denotes the dummy starting node, j is a node in N, θ is the scenario and v the truck. The value of t^{θ}_{ojv} is zero. The time to finish a route is defined as t^{θ}_{fov}.

For each scenario θ, there is a probability p of order cancellation. Each order is defined as a combination of an initial node, a terminal node, the earliest arrival time, the latest finish time, and the probability of cancellation. An order is not canceled in advance. We assume that it happens when the truck arrives to pick it up and the order is not ready to be transported. This is a common problem for middle mile routing. For those orders, we create a dummy order that is a loop in its pick up node I_k. In addition, we create a new order, similar to the original, but with time windows postponed two hours. As a result, the truck may wait for this order at the node, or another truck will pick it up later. Table 5.1 gives the sets, parameters, and variables used in the mathematical model.

The objective function in equation (5.1) minimizes the expected cost of the route, considering the travel time and cost of trucks per scenario multiplied by its respective probability. In addition, the time to visit a node is also considered, multiplied by 0.01. The reason for adding the time is to force the times for unvisited nodes and unused trucks to equal 0.

$$\min \sum_{i\in N}\sum_{j\in N}\sum_{v\in V}\sum_{\theta\in\Theta} p^{\theta}(w_{ij}x^{\theta}_{ijv} + d^{\theta}_{v} + 0.5x^{\theta}_{iiv} + 0.01t^{\theta}_{ijv}) \tag{5.1}$$

The constraints of the model follow:

$$\sum_{j\in N} x^{\theta}_{ojv} \leq 1 \quad \forall v \in V, \theta \in \Theta \tag{5.2}$$

$$\sum_{j\in N\backslash\{o\}} x^{\theta}_{ojv} = \sum_{i\in N\backslash\{f\}} x^{\theta}_{ifv} \quad \forall v \in V, \theta \in \Theta \tag{5.3}$$

$$\sum_{j\in N} x^{\theta}_{ijv} - \sum_{j\in N} x^{\theta}_{jiv} = 0 \quad \forall i \in N\backslash\{o,f\}, v \in V, \theta \in \Theta \tag{5.4}$$

$$x^{\theta_1}_{ojv} = x^{\theta_2}_{ojv} \quad \forall j \in N, v \in V, \theta_1 \in \Theta\backslash\theta_2, \theta_2 \in \Theta\backslash\theta_1 \tag{5.5}$$

$$x^{\theta}_{ijv} \leq \sum_{j\in N\backslash\{o\}} x^{\theta}_{ojv} \quad \forall i \in N\backslash\{o,f\}, v \in V, \theta \in \Theta \tag{5.6}$$

$$\sum_{v\in V} x^{\theta}_{I_kF_kv} \geq 1 \quad \forall k \in K^{\theta}, \theta \in \Theta \tag{5.7}$$

$$t^{\theta}_{ojv} = 0 \quad \forall j \in N\backslash\{o\}, \forall v \in V, \theta \in \Theta \tag{5.8}$$

Table 5.1: Model notation - sets, parameters, and variables

Sets	
Nodes – N	$i \in N$, o is the supply node and f the terminal node
Trucks – V	$v \in V$
Probabilities for scenario θ – P	$p^\theta \in P$
Order for scenario θ – K^θ	$k^\theta \in K^\theta$
Scenarios – Θ	$\theta \in \Theta$
Decision Variables for each scenario θ	
x^θ_{ijv}	binary variable: equals 1 if truck v goes from i to j; and equals 0 otherwise
s^θ_{ijv}	binary variable: equals 1 if truck v goes from i to terminal node f and from the initial node o to node j; and equals 0 otherwise
t^θ_{iv}	time that truck v arrives at node i
d^θ_v	distance from the last delivery to the initial node of truck v
Parameters	
I_k	initial node of order k^θ
F_k	final node of order k^θ
EST_k	earliest start time of order k^θ
LFT_k	latest final time of order k^θ
u_v	maximum route time for truck v
w_{ij}	average transit time between i and j
p^θ	probability of scenario θ
M	big number

$$t^{\theta}_{fov} \geq t^{\theta}_{ijv} \quad \forall i \in N\backslash\{o\}, j \in N\backslash\{f\}, v \in V, \theta \in \Theta \tag{5.9}$$

$$t^{\theta}_{ijv} + w_{ij} \leq t^{\theta}_{jlv} + M(2 - x^{\theta}_{ijv} - x^{\theta}_{jlv}) \quad \forall i, j \in N, l \in N, v \in V, \theta \in \Theta \tag{5.10}$$

$$t^{\theta}_{iiv} + 0.5 \leq t^{\theta}_{ijv} + M(2 - x^{\theta}_{iiv} - x^{\theta}_{ijv}) \quad \forall i, j \in N, v \in V, \theta \in \Theta \tag{5.11}$$

$$t^{\theta}_{iiv} \geq t^{\theta}_{jiv} + M(2 - x^{\theta}_{iiv} - x^{\theta}_{jiv}) \quad \forall i, j \in N, v \in V, \theta \in \Theta \tag{5.12}$$

$$t_{I_k v} + M(1 - x_{I_k F_k v}) \geq EST_k \quad \forall k \in K^{\theta}, v \in V \tag{5.13}$$

$$t^{\theta}_{F_k v} - M(1 - x^{\theta}_{I_k F_k v}) \leq LFT_k \quad \forall k \in K^{\theta}, v \in V, \theta \in \Theta \tag{5.14}$$

$$u_v \geq \sum_{i \in N} \sum_{j \in N} w_{ij} x^{\theta}_{ijv} + d_v \quad \forall v \in V \theta \in \Theta \tag{5.15}$$

$$x^{\theta}_{ojv} + x^{\theta}_{ifv} - 1 = s^{\theta}_{ijv} \quad \forall i \in N\backslash\{f\}, j \in N\backslash\{o\} v \in V, \theta \in \Theta \tag{5.16}$$

$$d^{\theta}_{v} \leq \sum_{i \in N\backslash\{f\}} \sum_{j \in N\backslash\{o\}} (w_{ij} s^{\theta}_{ijv}) \quad \forall v \in V, \theta \in \Theta \tag{5.17}$$

$$x^{\theta}_{iiv} \leq \sum_{j \in N\backslash\{i\}} \sum_{l \in N\backslash\{i\}} x^{\theta}_{ijv} + x^{\theta}_{liv} \quad \forall i, j \in N, v \in V, \theta \in \Theta \tag{5.18}$$

$$x^{\theta}_{ofv} = 0 \quad \forall i, j \in N, v \in V, \theta \in \Theta \tag{5.19}$$

$$x^{\theta}_{ijv} \in \{0, 1\}, \quad \forall i, j \in N, v \in V, \theta \in \Theta \tag{5.20}$$

$$s^{\theta}_{ijv} \in \{0,1\}, \quad \forall i,j \in N, v \in V, \theta \in \Theta \tag{5.21}$$

$$t^{\theta}_{iv} \geq 0, \quad \forall i \in N, v \in V, \theta \in \Theta \tag{5.22}$$

$$d^{\theta}_{v} \geq 0, \quad \forall v \in V, \theta \in \Theta \tag{5.23}$$

The objective function (5.1) minimizes the estimated total distance traveled including the return to the original node. It also considers the time multiplied by 0.01 to force unvisited nodes to have time equal 0.

Constraints (5.2) guarantee that each truck v will have at most one route for each scenario θ. Constraints (5.3) guarantee that if the truck v leaves the original dummy node o, it will finish at the final dummy node f. Constraints (5.4) ensure that if a truck enters a node, it has to leave it. Constraints (5.5) ensure that the truck will start at the same node in all scenarios. Different trucks can start in different nodes. Constraints (5.6) guarantee that the truck will start the route at the original dummy node. Constraints (5.7) ensure that at least one truck will attend the order k. Constraints (5.8) define the initial time for each truck to be equal zero, and constraints (5.9) ensure that the time at the final dummy node will be greater than the times in all the other nodes during the route. According to constraints (5.10), the time sequence between two sequential nodes must be obeyed. This means that if node j follows i, the time that truck v arrives at j has to be greater than the time that it passed by i plus the transit time between the nodes. Constraints (5.11) and (5.12) ensure that time for the canceled order k is considered before leaving the center, and the truck will need to wait 0.5 h at the center, before leaving it. The canceled order is represented by a loop in the order initial node I_k. Constraint (5.13) and (5.14) guarantee that the time window to deliver the order will be respected. Constraints (5.15) ensure that the truck will respect the maximum time of the route, considering the return to the initial node. Constraints (5.16) define the variable s_{ijv} that records the first and last center visited by truck v. Constraints (5.17) calculate the return transit time for truck v. Constraints (5.18) guarantee that the truck will arrive in the node of a canceled order and leave it after it. Constraints (5.19) guarantee that there is no route from initial dummy node to the final dummy node. Constraints (5.20) to (5.23) define the domains of the binary variables x^{θ}_{ijv} and s^{θ}_{ijv}, and non-negative variables t^{θ}_{iv} and d^{θ}_{v}.

5.4 Evaluating and planning load cancellation

Figure 5.1 represents the dataset used to evaluate the model (Gronalt et al., 2003). In our problem we consider the possibility of using four trucks.

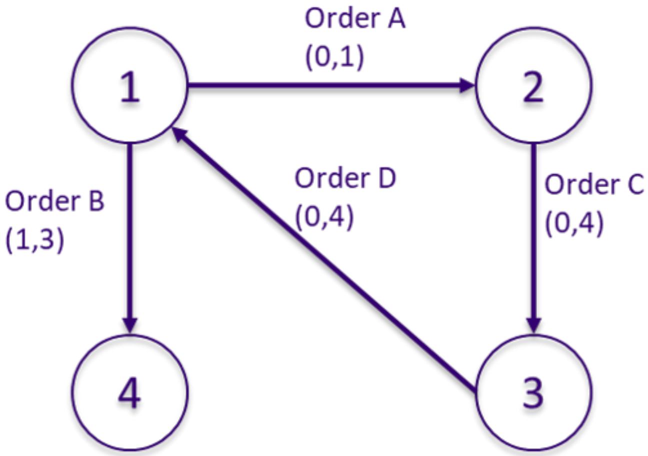


Figure 5.1: Small problem with four centers and four orders. The time window is represented in parenthesis for each order (EST,LFT)

Besides a scenario without order cancellations, we evaluate ten other scenarios given in Table 5.2. The stochastic programming model defines routes and times for each scenario that considers different cancellation probabilities.

Figure 5.2 exemplifies the solution for three scenarios. Scenario 0 does not have canceled orders, Scenario 2 has order B canceled and postponed by two hours, and Scenario 7 has orders A and D canceled and postponed by two hours. Each canceled order is represented by a loop in its initial node. As shown in Figure 5.2, due to the cancellations, Scenarios 2 and 7 needed to assign an extra vehicle for the routes. We assume that each vehicle has a cost of five, as in Table 5.2.

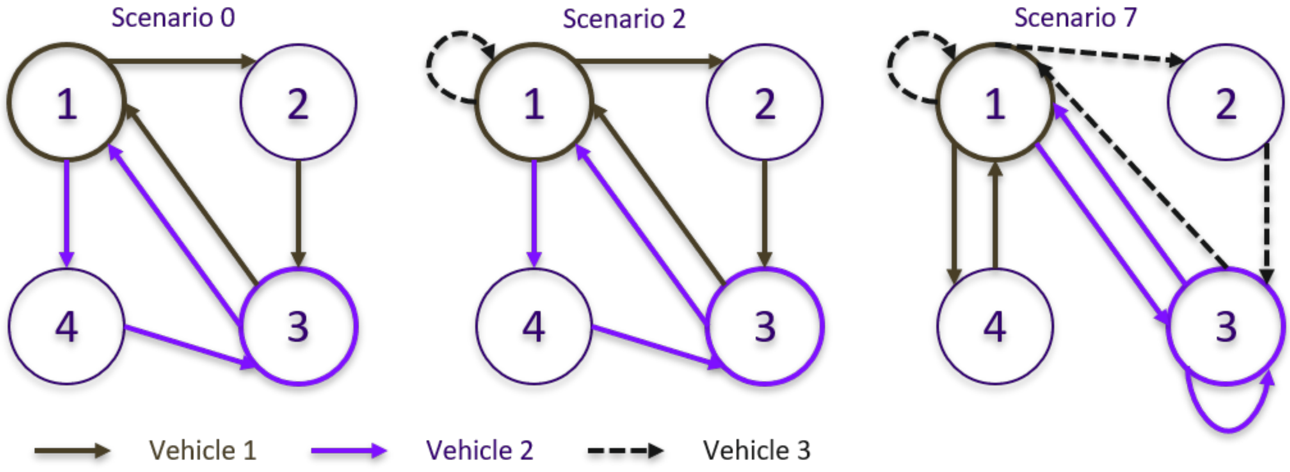


Figure 5.2: Three solutions of the cancellation scenarios. Loops represent canceled orders

One way of calculating the cost of cancellations would be to calculate the cost difference between the expected cost of the routes with cancellation and the cost of the routes without cancellation. In our example, the cost of cancellations would be $22.15 - 15.25 = 6.90$.

5.5 *Conclusion and next steps*

This chapter presented a stochastic mixed-integer programming model that solves middle mile routing problems in which uncertainty of demand is due to last minute cancellations and reschedules are considered. This model considers order cancellation, reschedule, time windows, and multi-depot. The objective function is to minimize the total cost of the route, considering distance and cost of trucks for different cancellation scenarios. It gives routes for different scenarios, each of them with a different probability of order cancellation.

In this chapter, we also suggest a method of evaluating the cost of cancellation based on the calculated routes and vehicles used, comparing the costs without any cancellation and the expected costs of the scenarios with cancellations.

As a next step, we suggest adding to the model multiple days and driver schedule (e.g., sleep and rest time), try different stochastic models, and solve this problem using a version of VeRSA, as discussed in Chapter 6.

Table 5.2: Example of calculating expected penalty. Vehicle cost = 5.00. Cost without cancellation = 15.25. Expected cost for scenarios with cancellation = 22.15. Cost difference = 6.90

Scenario	Confirmed orders	Canceled orders	Total distance	Routes	Cost(distance+ cancellation+ trucks)	Used trucks	Prob- ability
0	A-B-C-D	0	5.25	1-2-3-1 3-1-4-3	5.25+0+10 15.25	2	1
1	B-C-D	A	5.25	3-1-4-3 1-1-2-3-1	5.25+0.5+10 15.75	2	0.2
2	A-C-D	B	5.25	1-2-3-1 3-1-4-3 1-1	5.25+0.5+15 20.75	3	0.07
3	A-B-D	C	5.25	3-1-4-3 1-2-2-3-1	5.25+0.5+10 15.75	2	0.1
4	A-B-C	D	6.78	1-2-3-1 3-3-1-3 1-4-1	6.78+0.5+15 22.28	3	0.5
5	C-D	A-B	5.25	3-1-4-3 1-1-2-3-1	5.25+1.0+10 16.25	2	0.014
6	B-D	A-C	5.25	3-1-4-3 1-1-2-2-3-1	5.25+1.0+10 16.25	2	0.02
7	B-C	A-D	6.78	1-4-1 3-3-1-3 1-1-2-3-1	6.78+1.0+15 22.78	3	0.1
8	A-D	B-C	5.25	3-1-4-3 1-1-2-2-3-1	5.25+1.0+10 16.25	2	0.007
9	A-C	B-D	5.25	1-1 3-3-1-4-3 1-2-3-1	5.25+1.0+15 21.25	3	0.035
10	A-B	C-D	6.78	1-4-1 3-3-1-3 1-2-2-3-1	6.78+1.0+15 22.78	3	0.05

Chapter 6

GENERALIZED VARIATIONS OF VERSA FOR VEHICLE ROUTING

This chapter presents an *optimization methodology and solution technique* to solve last mile routing problems. We discuss a generalization of the Vehicle Routing and Scheduling Algorithm (VeRSA) (Zabinsky et al., 2019a;b) for vehicle routing problems with time windows. A version of this algorithm was used to create the tool presented in Chapter 3. The differences of the algorithm in this chapter and the one from Chapter 3 are the objective functions, index and feasibility check. Those are easy to adapt to any particular problem. The most important part of the algorithm is its tree structure and how to search the tree. Most of the chapter is in a paper in progress by a master student Yi Chu, Prof. Zelda B. Zabinsky and I. It will be submitted in June. My contribution to this work was on designing the algorithm collaboratively, discussing variations and their relationships to the literature, defining the indexing, and running comparisons to other models. The variation of VeRSA used in Chapter 3 was based on my design and indexing.

6.1 Introduction

Vehicle routing problems (VRP) are known NP-Hard problems (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012), and if time windows are considered, the difficulty of solving the problem increases (Liu et al., 2010). Therefore, solving VRP or vehicle routing problems with time windows (VRPTW) for large instances of 50 customers or more using mathematical model is difficult. Consequently, many authors focus their efforts on developing heuristics or meta-heuristics algorithms that give good solutions, near-optimal, in a reasonable time (Cordeau et al., 2002; Vidal et al., 2013a). One of the main problem of these methods is the lack of optimality gap or guarantee that it will reach the optimal solution in some moment in time. As an alternative to those methods, we present VeRSA.

VeRSA embeds a heuristic index inspired by scheduling methods into a branch-and-bound tree. Its efficiency depends on obtaining a good incumbent solution quickly using the index and coupling it with a theoretical lower bound for pruning and providing an optimality gap.

VeRSA can be used as a framework to solve different VRP problems and variations. The branch-and-bound algorithm with the indexing rule shows results similar to heuristic and exact method approaches, and it gives an optimality gap, eventually proving optimality.

6.2 Literature review

Due to the complexity of solving a vehicle routing problem (Laporte and Louveaux, 2002; Kumar and Panneerselvam, 2012), most of the recent papers in the field are focused on heuristic and metaheuristics approaches that give near-optimal solutions in a feasible time. Cordeau et al. (2002) and Vidal et al. (2013a) present an extensive review of different heuristic methods to solve VRPs.

Considering VRPTW, the current best heuristics for VRPTW are evolutionary strategies, large neighborhood search, iterated local search, and multi-start local searches (Nagata et al., 2010). Nagata et al. (2010) solve the problem using a penalty-based memetic algorithm. This algorithm combines an evolutionary algorithm with a local search. Hashimoto and Yagiura (2008) present a path-relinking algorithm that creates vehicle routes. Those routes are improved by local search. Infeasible routes are penalized by the use of penalties and the neighborhood for the local search is also chosen by a heuristic approach. Vidal et al. (2013b) present a hybrid genetic search with advanced diversity control, applying "time constraint relaxations during the search to benefit from infeasible solutions in the search space."

All those heuristic methods present good results and are state-of-art in heuristics for VRPTW. However, they do not guarantee optimal solutions. Therefore, another branch of research combines heuristics and mathematical models to reach the optimal solution or to present an optimality gap, although they do not present results as quickly as heuristics.

Prescott-Gagnon et al. (2009) present a large neighborhood search algorithm with branch-and-price that can reach an optimal solution to some problems if given enough time. Jepsen et al. (2006) present a non-robust branch-and-cut-and-price algorithm for VRPTW. Their algorithm solved to optimality some of the Solomon (1987) instances, and gave an optimality gap to others.

Our VeRSA method is based on Zabinsky et al. (2019a), and combines a heuristic indexing method to find feasible good solutions in a timely manner with an exact branch-and-bound method, that offers optimality gaps and optimal solution with time. It calculates the lower bound using the minimum weight spanning tree method. Other ways of calculating lower bounds are relaxed linear programming and Lagrangian multiplier (Liu et al., 2010; Gronalt et al., 2003; Adulyasak et al., 2015; Kolen et al., 1987).

6.3 Mixed integer programming model

In this chapter, we evaluate the use of VeRSA on VRPTW problems. In this section we present the mathematical model to this type of problem. Table 6.1 gives the sets, parameters and variables used by the model.

The objective function is to minimize the total travel time, as shown in (6.1).

$$\min \qquad \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} t_{ijk} x_{ijk} \tag{6.1}$$

This objective is subject to the constraints given in (6.2)-(6.10).

$$\sum_{k \in K} \sum_{j \in N \setminus \{i\}} x_{ijk} = 1, \quad \forall i \in N \setminus \{0\} \tag{6.2}$$

$$\sum_{j \in N \setminus \{0\},} x_{0jk} \leq 1, \quad \forall k \in K \tag{6.3}$$

$$\sum_{j \in N \setminus \{i\}} x_{jik} - \sum_{j \in N \setminus \{i\}} x_{ijk} = 0, \quad \forall i \in N, \forall k \in K \tag{6.4}$$

$$\sum_{i \in N \setminus \{0\}} x_{i0k} \leq 1, \quad \forall k \in K \tag{6.5}$$

Table 6.1: Model notation - sets, parameters, and variables

Variable	Description
Sets	
N	customer set that contains all customers, $N = \{0, 1, ..., n\}$
A	arc set that contains arcs between customer pairs, $A = \{(i, j) : i, j \in N\}$
K	vehicle set that contains all vehicles, $K = \{1, ..., k\}$
Parameters	
d_i	demand at customer i
Q_k	maximum capacity of vehicle k
a_i	earliest arrival time to customer i
b_i	latest departure time from customer i
s_{ik}	service time at customer i by vehicle k
t_{ijk}	travel time on arc (i, j) by vehicle k
Decision Variable	
x_{ijk}	binary variable, 1 indicates vehicle k travels along arc (i, j), 0 otherwise
y_{ik}	time that vehicle k arrives at customer i
z_{ik}	remaining capacity of vehicle k at node i.

$$\sum_{i \in N} d_i \sum_{j \in N \setminus \{i\}} x_{ijk} \leq Q_k, \quad \forall k \in K \tag{6.6}$$

$$y_{ik} + t_{ijk} + M(1 - x_{ijk}) \geq a_j, \quad \forall (i, j) \in A, \forall k \in K \tag{6.7}$$

$$y_{ik} + t_{ijk} - M(1 - x_{ijk}) \leq b_j, \quad \forall (i, j) \in A, \forall k \in K \tag{6.8}$$

$$y_{ik} + t_{ijk} + s_{jk} - M(1 - x_{ijk}) \leq y_{jk}, \quad \forall (i, j) \in A, \forall k \in K \tag{6.9}$$

$$x_{ijk} \in \{0, 1\}, \quad y_{ik} \geq 0, \quad \forall k \in K, \forall (i, j) \in A \tag{6.10}$$

Constraints in (6.2) limit each customer to be visited by only one vehicle. Constraints in (6.3) guarantee that each vehicle will start from the depot. Constraints in (6.4) enforce that if the vehicle enters a customer, it will have to leave it. Constraints in (6.5) guarantee that each vehicle will finish the route at the depot. Constraints in (6.6) define that the total demand of a route has to respect the vehicle capacity. Constraints in (6.7) and constraints in (6.8) define the earliest and latest time to visit a customer. Constraints in (6.9) ensure that the time sequence between two sequential nodes must be obeyed. This means that if node j follows i, the time that vehicle k leaves j has to be greater than the time that it left i plus the service time in j and transit time between the nodes. Constraints in (6.10) define the binary variable x_{ijk} and the non-negative variable y_{ik}.

6.4 VeRSA

VeRSA uses a tree structure to solve the VRP. The tree consists of a sequence of decisions that are represented by nodes in the tree. Each node represents a decision being made. The specification of VeRSA requires the tree structure and rules for traversing the tree, an index for constructing a feasible solution, a lower bound and an optimality gap. The decision contains which customer to visit and which vehicle to use. A complete branch of the tree corresponds to a feasible route, and its cost corresponds to the objective function.

The incumbent solution is a complete branch with lowest cost. At each node, a lower bound is calculated using a minimum weighted spanning tree. The indexing rule, explained in Section 6.4.2, is used to determine which branch to explore and to construct a feasible solution. The lower bound and the incumbent solution are used to calculate an optimality gap. If the lower bound at a node exceeds the incumbent value, the rest of the branch is pruned. If a node satisfies the feasibility and optimality checks, then the next node is explored.

The BnB tree is dynamically traversed by exploring branches, updating an incumbent solution, and pruning branches. The user defines the stopping criteria. It could be a threshold on the optimality gap, a predefined time, or confirmation of optimality.

6.4.1 Branch-and-Bound Tree and Feasibility Check

From the root node, there are n branches, each representing a different customer to visit first. To decide which node to visit at each step, we calculate the index of all feasible choices and add to the solution the customer having the largest index. Customers are added until the decision is to return to the depot. After returning to the depot, if there are still unvisited customers, a new route is defined with these unassigned customers. After visiting all customers, the vehicle returns to the depot. In this problem we have a maximum number of identical vehicles, and each of them will be used at most once.

To guarantee that the solution is feasible, we check whether each node respects the constraints of the problem. The constraints are similar to the ones from the mathematical model in Section 6.3. They are mainly related to vehicle capacity and time constraints. If a node fails the feasibility check, the rest of the branch is infeasible and we traverse the tree to find other solutions.

6.4.2 Indexing Rule

The indexing rule guides the construction of a complete feasible solution. Our index is created based on the mathematical model's objective function and constraints given in Section 6.3. Each node in the tree has associated state variables that include: current time, accumulated cost, unvisited customers, current vehicle, and capacity left for current vehicle. This information is used to calculate the index for each node that can be added. $I_{i,j}$ is the index for choosing j as the next customer to visit, given the vehicle is currently at customer i. For this version of VeRSA, we assumed all vehicles are identical so we dropped the subscript k. Below we give the index for the VRPTW:

$$I_{i,j} = e^{-(b_j - y_j)/\bar{b}} \times e^{-t_{i,j}/\bar{t}} \times e^{-(z_j - d_j)/Q}$$

where i and j represent the current node and potential next node, respectively, b_j is the latest departure time for customer j, t_{ij} is the transit time on arc (i, j), d_j is the demand at customer j, Q is the maximum capacity of the vehicle, z_j is the remaining capacity of the vehicle at node j, $z_j = Q - (\sum_{i \in N} d_i \sum_{j \in N \setminus \{i\}} x_{ij})$, the current y_j is the time that vehicle arrives at customer j, $\bar{b} = \frac{1}{n} \sum_{j=1}^{n} b_j$ represents the mean of b_j, and $\bar{t} = \frac{1}{n} \sum_{j=1}^{n} t_{i,j}$ represents the mean of t_{ij}.

6.4.3 Lower Bound and Optimality Gap

Calculating a lower bound in an algorithm is important to quantify an optimality gap from a current incumbent solution, and to confirm optimality. Common ways of calculating a lower bound include relaxed linear programming, minimum weight spanning tree, and Lagrangian relaxation (Liu et al., 2010; Gronalt et al., 2003; Adulyasak et al., 2015; Kolen et al., 1987). In the VeRSA algorithm, a minimum weight spanning tree algorithm calculates the lower bound. At the beginning of each iteration, after defining an incumbent solution, the algorithm calculates a lower bound at a node. This lower bound at a node is the estimated best solution that a branch can reach, based on a given partial branch. The tree lower bound is calculated at the end of each iteration and gives the optimality gap between the incumbent solution and the best theoretical solution.

For a lower bound at a node, the algorithm calculates a minimum weight spanning tree upon all customers that have not been visited, adding the depot. The minimum weight spanning tree is the "true" lower bound to our partial branch because it relaxed the capacity constraint and assumes that one vehicle can visit all remaining customers in one route. We use local lower bounds for pruning purpose. If our incumbent solution has lower cost than the lower bound on a partial branch, that branch is pruned.

The lower bound takes the minimum bounds calculated among all sub-branches at the end of each iteration. The lower bound gives the optimality gap that is the gap between the incumbent cost and this lower bound.

To define which sub-branch to explore next, an elite set is defined based on the best incumbent solutions calculated. The size of the elite set is equal the number of available vehicles.

The minimum weight spanning tree is computationally light. However, it does not give a tight bound, as shown in our future examples.

Algorithm 1 VeRSA Algorithm for VRPTW

```
initialization: set solution state variables to their initial value, let elite set $\mathcal{E} = \{0 \text{ (root)} : \text{(solution state variables, cost} = \infty\text{, lower bound} = \infty)\}$, visited nodes $\mathcal{P} = \emptyset$, incumbent cost $C = \infty$, overall lower bound $LB = \infty$;
Step 1. **Branching**.
for each node $e$ in $\mathcal{E}$ do
    1-1. Remove $e$ from $\mathcal{E}$;
    1-2. Branch on all unvisited customers that are feasible from $e$ (plus depot if not at depot), create a new node $p$ each time branching is done;
    1-3. Update node $p$ (label, current customer, current vehicle, current time, remaining vehicle capacity, unvisited customers);
    1-4. Add $p$ to $\mathcal{P}$;
end
Step 2. **Calculating Lower Bounds**.
for each new node $p$ in $\mathcal{P}$ do
    2-1. Calculate lower bound associated with $p$, $LB_p$. Update $LB = \min_{\widetilde{p} \in P} LB_{\widetilde{p}}$;
end
Step 3. **Construct Feasible Solutions**.
for each new node $p$ in $\mathcal{P}$ do
    3-1. Construct a feasible solution guided by indexing rule, or claim infeasible. Update the cost associated with $p$ if feasible;
    3-2. If new solution's cost is lower than cost of current incumbent solution, update incumbent solution and incumbent cost $C$;
end
Step 4. **Compute Optimality Gap and Check Stopping Criterion**.
for each new incumbent solution do
    4-1. Compute optimality gap, $GAP = C - LB$. If optimality gap is less than a desired threshold, or if $\mathcal{P} = \emptyset$, stop, output the incumbent solution with its cost, the lower bound $LB$, and optimality gap, $GAP$. Otherwise, proceed to step 5;
end
```

```
Step 5. Decide Elite Set.
for all nodes $p$ in $\mathcal{P}$ do
    5-1. Delete node $p$ if $LB_p \geq C$;
    5-2. Move $m$ nodes with lowest cost in $\mathcal{P}$ to $\mathcal{E}$, $m$ being the total number of vehicles;
end
Go back to Step 1;
```

6.4.4 Traversing the Branch and Bound Tree

VeRSA uses dynamic route construction, which does not require maximum number of routes. The algorithm does a top down traversing: first explores n sub-branches, n being the total number of customers. In each sub-branch visits a customer $j, j \leq n$ as the first move. At each of these n sub-branches, the index described in Section 6.4.2 guides the selection of nodes to find a feasible solution, that finalizes one iteration. At the end of each iteration, the global lower bound and optimality gap are updated. In addition, the elite set is defined to be explored in the following iteration. The size m of the elite set is equivalent to the number of vehicles available at the depot. Using a fixed number of branches to define the size of the elite set prevents it to grow exponentially. In the following iteration, the algorithm explores all possibilities of second node at the elite branches, resulting in $m \times n$. This quantity also contains the possibility of returning to the depot.

Algorithm 1 gives VeRSA traversing algorithm for VRPTW problems. VeRSA uses a top down traversing method. Node is defined as {label:(current customer, current vehicle, current time, remaining vehicle capacity, unvisited customers, cost incurred, lower bound)}. The label represents consecutive decisions being made, as shown in Algorithm 1.

An example of VRPTW and illustrations on how to traverse the decision tree are shown in Figures 6.1, 6.2, and 6.3. Figure 6.1 gives a VRPTW problem with one depot and three customers. Each customer has a demand and a time window to be visited. In this example, we consider only one vehicle that should visit all customers. Figure 6.2 gives the first iteration of traversing the tree, branching it at the depot. Using Algorithm 1, each branch results in a feasible solution, and an elite set is defined according to the optimality gap, including the best incumbent solutions. In this case, the elite set is the branch containing nodes 0-1, with lower incumbent solution, 29. After the first iteration the incumbent value is 29, the lower bound, 23, and the optimality gap, 6. Figure 6.3 gives the second iteration, in which the branches from the elite set are explored using the Algorithm 1. In this second iteration, the elite set is the branch containing nodes 0-1-3, with an incumbent solution of 25. After the scond iteration the incumbent value is 25, the lower bound continues 23, and the optimality gap, 2.

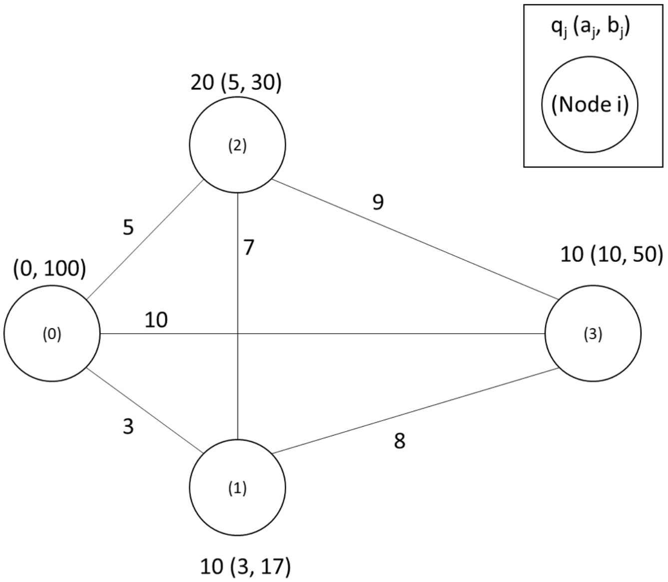


Figure 6.1: Example of VRPTW using one vehicle, the number inside the circle is the node number i (0 is the depot), the capacity q_i and the time window (a_i,b_i) is above the circle, and the numbers near the arcs represent cost, i.e., c_{ij}.

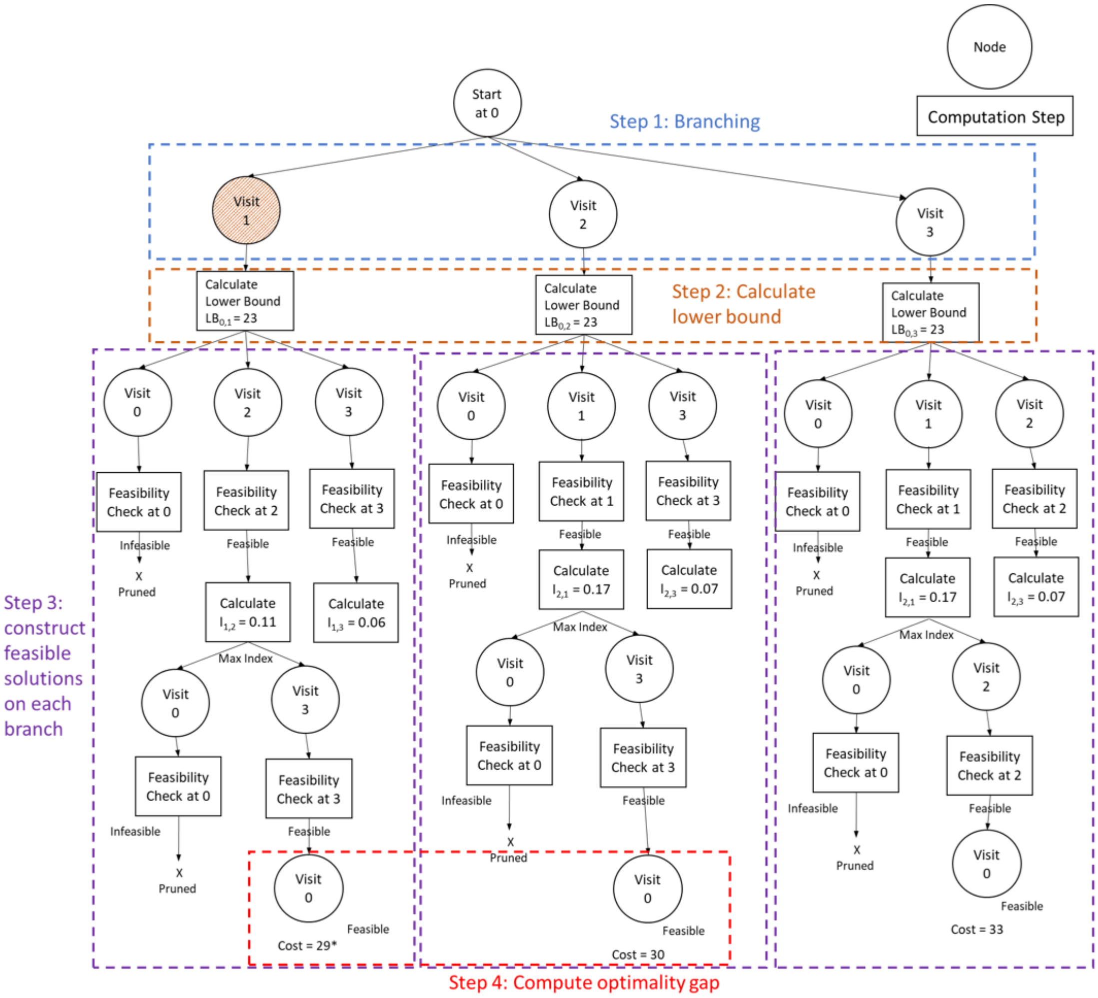


Figure 6.2: First Iteration - Example of traversing the decision tree, circles represent nodes in decision tree, rectangular represents calculations being made in certain step, * labels incumbent solution in each iteration (and hence the elite branch in this particular case).

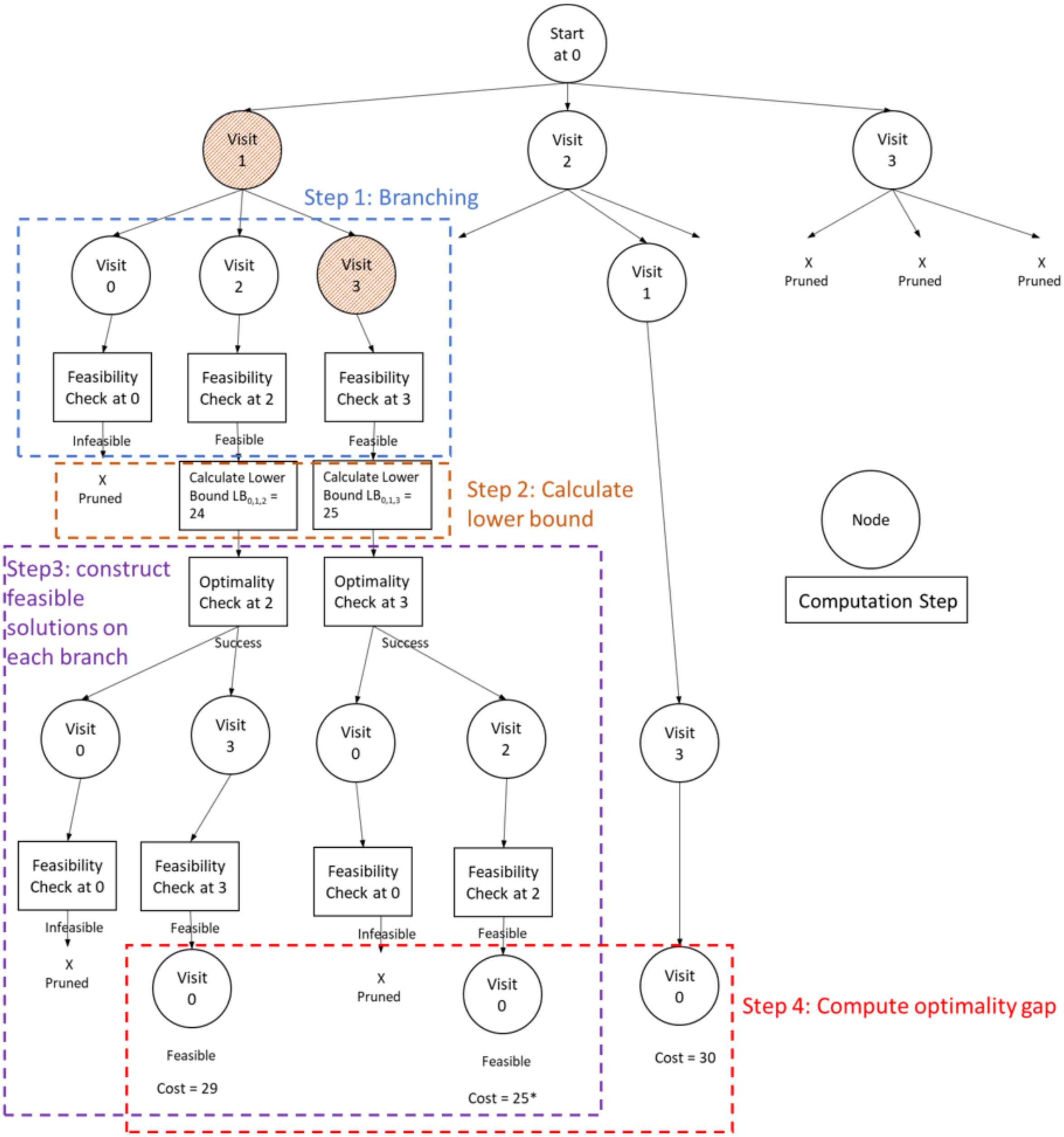


Figure 6.3: Second Iteration - Example of traversing the decision tree, circles represent nodes in decision tree, rectangular represents calculations being made in certain step, * labels incumbent solution in each iteration (and hence the elite branch in this particular case).

6.5 Numerical Results

In instances with 10 customers plus depot and 15 customers plus depot (Solomon, 1987), VeRSA proved optimality within 0.34s and 191.86s, respectively. For the same instances, Gurobi took 0.34s and 11.87s, respectively. However, VeRSA found the optimal solution for both instances in less than 0.03s. VeRSA performance is shown in Tables 6.2 and 6.3.

VeRSA and Gurobi were also compared in larger instances of 25 and 50 customers, including depot. Figures 6.4 and 6.5 represent the numerical result of VeRSA on two Solomon instances (Solomon, 1987). For both instances, VeRSA gives good incumbent solution in a timely manner, faster than Gurobi. However, VeRSA could not prove optimality in 10 minutes, whereas Gurobi proved optimality in 65s and 440s, for 25 and 50 customers, respectively.

Time	Incumbent	Lower Bound
0.02	58.33	35.21
0.08	58.33	51.21
0.19	58.33	52.04
0.29	58.33	52.80
0.34	58.33	55.66

Table 6.2: Result of VeRSA for the first 10 centers from dataset C101 (Solomon, 1987). VeRSA confirmed optimality by pruning at 0.34 seconds.

Instance	Incumbent[1] (one second)	Incumbent[2] (10 minutes = 600 seconds)	Ref.	Time (seconds)	Lower Bound	Upper Bound
C101	879.7	849.3	KDMSS	3	827.3	827.3

C102	1513.9	1180.2	KDMSS	13	827.3	827.3
C103	1622.2	1301.2	KDMSS	34	826.3	826.3
C104	1598.4	1484.8	KDMSS	4113	822.9	822.9
C105	945.4	828.9	KDMSS	5	827.3	827.3
C106	1320.5	963.5	KDMSS	7	827.3	827.3
C107	1102.2	936.9	KDMSS	7	827.3	827.3
C108	1308.6	1104.9	KDMSS	14	827.3	827.3
C109	1361.5	1131.5	KDMSS	21	827.3	827.3
C201	591.6	591.6	CR+KLM	203	589.1	589.1
C202	1094.9	983.8	CR+KLM	3483	589.1	589.1
C203	1169.1	1103.4	KLM	13071	588.7	588.7
C204	1107.3	1067.5	IV	-		588.1
C205	591.6	588.9	CR+KLM	417	586.4	586.4
C206	674.3	652.4	CR+KLM	595	586.0	586.0
C207	689.5	661.6	CR+KLM	1241	585.8	585.8
C208	658.3	645.9	KLM	555	585.8	585.8
R101	2090.9	1771.8	KDMSS	20		
R102	2112.7	1869.9	KDMSS	4	1631.2	1637.7
R103	2004.4	1825.9	CR+L	56	1466.6	1466.6
R104	1890.5	1680.5	IV	-		
R105	1920.9	1680.3	KDMSS	127	1346.2	1355.3
R106	1958.1	1634.7	CR+KLM	511	1227.0	1234.6
R107	1952.0	1721.2	CR+KLM	-		
R108	1858.7	1667.7	JPSP	-		
R109	1791.3	1588.1	CR+KLM	-		
R110	1862.9	1644.7	CR+KLM	-		
R111	1851.6	1712.6	CR+KLM	-		

R112	1777.7	1608.2	KDMSS	-		
R201	1729.9	1589.7	KLM	-		
R202	1497.7	1406.7	JPSP	-		
R203	1502.6	1355.7	JPSP	-		
R204	1324.4	1120.5	-	-		
R205	1533.5	1435.5	-	-		
R206	1373.0	1359.0	-	-		
R207	1345.5	1318.0	-	-		
R208	1195.6	1100.3	-	-		
R209	1474.8	1376.6	JPSP	-		
R210	1511.9	1418.5	-	-		
R211	1376.5	1247.9	-	-		
RC101	2316.6	1985.0	KDMSS	57	1584.1	1619.8
RC102	2461.5	1986.9	CR+KLM	-		
RC103	2509.8	2141.4	CR+KLM	-		
RC104	2585.4	2178.7	IV	-		
RC105	2334.1	2103.6	KDMSS	310	1472.0	1513.7
RC106	2092.8	1948.6	JPSP	-		
RC107	2338.2	1948.6	IV	-		
RC108	2450.6	2026.8	IV	-		
RC201	1928.8	1796.3	KLM	-		
RC202	1962.0	1751.6	IV+C	-		
RC203	1811.1	1586.3	JPSP	34064	922.6	923.7
RC204	1425.2	1311.1	-	-		
RC205	2073.4	1884.0	IV+C	-		
RC206	1761.9	1669.3	JPSP	-		
RC207	1791.5	1687.5	-	-		

RC208	1642.5	1486.5	-	-

Table 6.4: Result of VeRSA for test sets with 100 customers compared to some other algorithms. The first column shows the objective function value of the incumbent solution that VeRSA found in one second, the second column shows the incumbent value at 10 minutes, the third column shows the best method from Jepsen et al. (2006) with time and lower and upper bounds. “A (-) means that the instance was not solved” (Jepsen et al., 2006). The *Ref.* represents the methods compared as following: CR, Cook and Rich (1999): k-path cuts; IV, Irnich and Villeneuve (2006): shortest path problem with Resource Constraints k-cycle elimination (k-cyc-SPPRC); JPSP, Jepsen et al. (2006): branch-and-cut-and-price; KDMSS, Kohl et al. (1999): two-path cuts; KLM, Kallehauge et al. (2000): Lagrangean duality and non-differentiable optimization

We also compared VeRSA with state-of-the art exact algorithms (Jepsen et al., 2006). The result is given in Table 6.4. The first column is the Solomon (1987) dataset tested. The second column, Incumbent^1, is the incumbent solution that VeRSA found within 1 second, and the third column, Incumbent^2, is the incumbent solution that VeRSA found within 10 minutes. The remain columns are the best exact method from Jepsen et al. (2006) with time, and lower and upper bounds. A hyphen indicates that no feasible solution was found.

From the 56 datasets tested, VeRSA gave an incumbent solution to 32 unsolved instances by exact methods in Jepsen et al. (2006), and presented feasible solution with objective value within 1% of the optimal value for three instances, after running for 10 minutes. Table 6.4 shows that VeRSA identifies a feasible solution quickly, within 1 second. However, it takes longer to improve the solution and find the optimal solution. One of the reasons for this performance is a loose lower bound.

However, Pecin et al. (2017) present a branch-price-and-cut algorithm that solved all instances from Solomon (1987). Table 6.5 gives the average solution time for each of the

Time	Incumbent	Lower Bound
0.03	142.14	65.86
0.25	142.14	81.86
0.68	142.14	82.09
1.05	142.14	82.09

Table 6.3: Result of VeRSA for the first 15 centers from dataset C101 (Solomon, 1987). VeRSA confirmed optimality at 191.86 seconds.

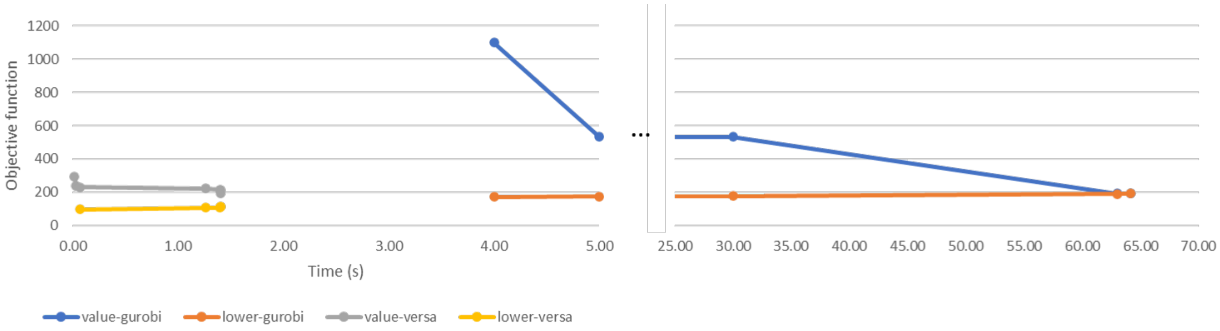


Figure 6.4: Data C101, first 25 customers VRPTW

instance classes given in Solomon (1987).

We also tested VeRSA on larger instances, with 200, 400, 600, 800, and 1000 customers (Gehring and Homberger, 2001), as shown in Table 6.6. For all instances, VeRSA found an incumbent solution in less than one second.

Pecin et al. (2017) ran their branch-price-and-cut algorithm using the instances with 200 customers (Gehring and Homberger, 2001). They found optimality for 51 of the 60 instances, in an average time of 4.2 hours. However, we do not know about exact method that tested the larger instances, greater than 200 customers.

Comparing exact methods to heuristic methods, the second one gives better solutions

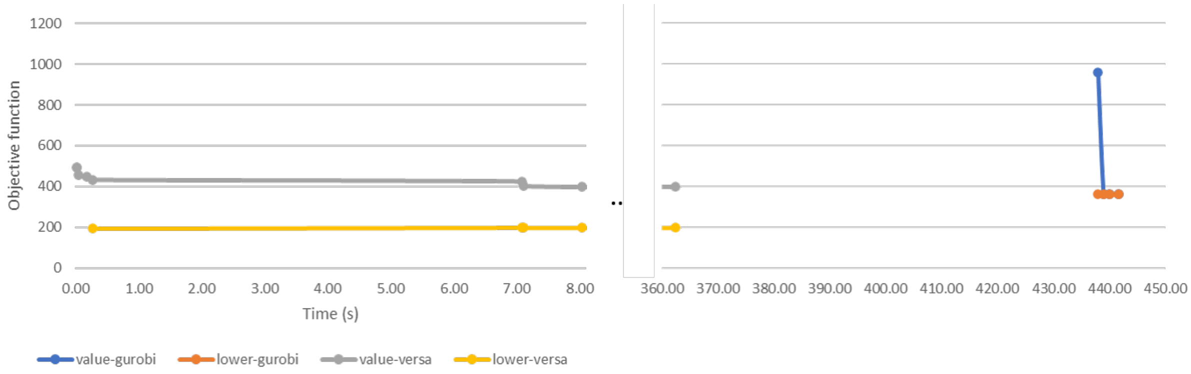


Figure 6.5: Data C101, first 50 customers VRPTW

Class	Time(s)
C1 (C101-C108)	15
C2 (C201-C208)	328
R1 (R101-R108)	31
R2 (R201-R208)	6,432
RC1 (RC101-RC108)	52
RC2 (RC201-RC208)	337

Table 6.5: Summary of solution time for Solomon (1987) using Pecin et al. (2017) algorithm

faster, as shown in Homberger and Gehring (2005), Pureza et al. (2012), and Transportation Optimization Portal (2008). However, heuristic methods do not provide an optimality gap or have a mechanism to confirm optimality.

6.6 Summary and Future Works

VeRSA demonstrated good results combining indexing method with branch-and-bound algorithm. Its incumbent solution gives a good feasible solution in a timely manner in comparison to exact methods, even for larger datasets, with 1,000 customers. However, it still needs to improve on calculating the lower bound. The minimum weight spanning tree gives a loose bound, that results in a long time to prove optimality. In addition, we believe that we could improve the solution faster adding randomness to the elite set and local search while calculating a solution.

Instance	NC	maximum runtime (minutes)	Incumbent^1 (at one second)	LB^1	Incumbent^2 (at maximum runtime)	LB^2
C1_2_1	200	20	4132.62	817.77	3365.98	817.77
C1_2_2	200	20	5326.27	817.77	4594.45	817.77
C1_2_3	200	20	6718.48	817.77	6718.48	817.77
C1_2_4	200	20	6113.89	817.77	5153.80	817.77
C1_2_5	200	20	4422.20	817.77	3613.04	817.77
C1_4_1	400	40	12119.57	1589.63	11007.87	1589.63
C1_4_2	400	40	15580.71	1589.63	14097.31	1589.63
C1_4_3	400	40	16458.95	1589.63	14819.32	1589.63
C1_4_4	400	40	14972.72	1589.63	13968.87	1589.63
C1_4_5	400	40	21954.54	1589.63	11883.71	1589.63
C1_6_1	600	60	24042.04	2675.54	21954.54	2675.54
C1_6_2	600	60	31670.49	2675.54	30326.24	2675.54
C1_6_3	600	60	34792.96	2675.54	32657.52	2675.54
C1_6_4	600	60	32628.62	2675.54	31982.86	2675.54
C1_6_5	600	60	27511.85	2675.54	25412.39	2675.54
C1_8_1	800	80	45169.80	3744.09	42446.40	3744.09
C1_8_2	800	80	61785.88	3744.09	58671.77	3744.09
C1_8_3	800	80	62339.65	3744.09	60959.43	3744.09
C1_8_4	800	80	65577.66	3744.09	64223.62	3744.09
C1_8_5	800	80	50990.82	3744.09	47015.80	3744.09
C1_10_1	1000	100	75995.71	5037.58	71439.60	5037.58
C1_10_2	1000	100	102393.41	5037.58	96092.06	5037.58
C1_10_3	1000	100	111357.17	5037.58	109418.30	5037.58
C1_10_4	1000	100	118921.95	5037.58	114467.85	5037.58
C1_10_5	1000	100	80676.38	5037.58	77296.48	5037.58

Table 6.6: Result of VeRSA on large dataset (Gehring and Homberger, 2001). NC indicates number of customers of the dataset, third column is the maximum runtime for VeRSA, incumbent^1 and LB^1 is the result of incumbent cost and lower bound we get within 1 second for running VeRSA, incumbent^2 and LB^2 are these values after running the maximum time of VeRSA.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This dissertation focuses on formulating optimization models that incorporate realistic constraints and address uncertainty and multiple objectives, and develops optimization methodology and solution technique for larger problems. We detail the application of the models and techniques to a real problem of vaccine distribution in Mozambique. Then, we discuss commercial distribution of perishable goods and commercial middle mile routing.

We consider multiple objectives, such as minimize transit time, penalties for unreliable roads and vehicles, and loss of quality. In addition, we account for uncertainty of transit time and uncertainty of demand.

Chapter 3 *formulates optimization models that incorporate realistic constraints and address multiple objectives, and presents a solution technique* to solve the distribution of vaccines and medical supplies in Mozambique. It gives a model and an indexing method for distributing vaccines and medical supplies in rural areas. The optimization model included two objectives and used penalty parameters to represent the possibility of wasting vaccines due to temperature violation. In order for the optimization approach to be useful for routing planners in Mozambique, it was necessary for the computation time to be very fast. Instead of using classic mixed-integer programming solvers, such as Gurobi, CPLEX, GLPK or CBC, a new version of an indexing-based method was applied to this problem. We presented an Excel spreadsheet-based optimization tool for routing and scheduling to efficiently distribute vaccines and other medical commodities to health centers across Mozambique. It describes the tool and the process used to define the problem and obtain feedback from users during the development. The distribution and routing tool, named Route Optimization Tool (RoOT), uses an indexing algorithm to optimize the routes under constrained resources. This project was jointly developed by the University of Washington and VillageReach.

RoOT gives good solutions in a timely manner. The final users do not have time or resources available to run an optimization model for hours or days to find the optimal solution. They want a good solution in one or two minutes, and RoOT is capable of that, as shown in this paper. Moreover, RoOT obtained good solutions within two minutes on the 50 center datasets. Scalability and speed are important factors for the users.

Considerations for future route optimization versions of RoOT include multiple day routes, with mixed transportation modes (e.g., land vehicles and boats) and island deliveries. This will require discussion on how intermediary storage of vaccines may be handled over multiple days. The indexing algorithm used in RoOT needs to be improved to find better solutions faster and reduce the optimality gap with a tighter lower bound.

In Chapter 4, we *formulate optimization models that incorporate realistic constraints, uncertainties, and address multiple objectives.* We present two models to tackle SVRP with uncertainties of demand, transit time, and loss of quality for perishable products: robust optimization and stochastic program with recourse. We also consider two objectives. The objectives considered were to minimize a combination of transit time and penalties for waste products during transit, and minimize the time to deliver the products considering their freshness. Using a baseline model with expected values for uncertain parameters, we demonstrate the trade-off between transit time plus uncertainties and loss of freshness.

The baseline expected model is good at giving an overall view of the efficient frontier and trade-offs between objectives. The robust model also provides trade-offs between objectives with more detail by scenario, It can also identify conservative situations when a solution is infeasible. However, the robust model cannot provide a way to compensate if conservative solution is not feasible for extreme scenarios. The stochastic program with recourse provides guidance on how to handle extreme scenarios while keeping a stable (first stage) solution. It can be used for strategic planning and operational decisions.

In Chapter 5 we *formulate optimization models that incorporate realistic constraints and* *uncertainties* to solve middle mile routing problem applied to commercial logistics problems. One of the main difficulties of this type of routing is that the trucks have to be allocated without the certainty of demand. This situation can lead to last minute cancelations or ad hoc demands that jeopardize planning. In this case, in addition to uncertain demand of the customers in the last mile, the middle-mile distribution is uncertain. For example, a truck may arrive at a distribution center expecting to pick up an order to transport, only to discover that the order has been cancelled. It is also possible that this cancelled order is rescheduled for a later time, thus the optimization model must consider order cancellations and ad hoc additional orders.

We developed a mixed-integer stochastic programming model of a planning and scheduling problem in middle mile transportation. This model considers order cancellation, reschedule, time windows, and multi-depot. It gives routes for different scenarios, each of them with a different probability of order cancellation.

In Chapter 5, we also suggest a method of evaluating the cost of cancellation based on the calculated routes and vehicles used, comparing the costs without any cancellation and the expected costs of the scenarios with cancellations.

As a next step, we suggest adding as constraint the driver schedule, and solve this problem using the Vehicle and Scheduling Algorithm (VeRSA) (Zabinsky et al., 2019a) for lager problems.

In Chapter 6, we present an *optimization methodology and solution technique* to solve last mile routing problems. While mathematical models are important to describe the problems, the solution technique is also important to convey a meaningful solution. There are exact solution techniques available to solve small instances (e.g., small number of customers) and calculate bounds on the optimal solution. Thus exact methods can confirm optimality and provide an optimality gap on intermediate solutions. We discuss a generalization of the Vehicle Routing and Scheduling Algorithm (VeRSA) for vehicle routing problems with time windows. A version of this algorithm was used to create the tool presented in Chapter 3. The differences of the algorithm in Chapter 6 and the one from Chapter 3 are the objective functions, index and feasibility check. In VeRSA, those are easy to adapt to any particular problem. The most important part of the algorithm is its search structure to solve a problem. VeRSA combines an indexing approach with a quickly calculated lower bound, to provide a hybrid method that embeds a heuristic with an exact lower bound, providing a good feasible solution with an optimality gap quickly.

VeRSA demonstrated good results combining indexing method with branch-and-bound algorithm. Its incumbent solution gives good feasible solution in a timely manner in comparison to exact methods. However, it still needs to improve on calculating the lower bound. The minimum spanning tree gives a loose bound, that results in a long time to prove optimality. In addition, we believe that we could improve the solution faster adding randomness to the elite set and local search while calculating a solution.

BIBLIOGRAPHY

Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

Morteza Ahmadi, Abbas Seifi, and Behnam Tootooni. A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transportation Research Part E: Logistics and Transportation Review*, 75:145–163, 2015.

Vedat Akgün, Amit Parekh, Rajan Batta, and Christopher Rump. Routing of a hazmat truck in presence of weather systems. *Computers & OR*, 34:1351–1373, 05 2007. doi: 10.1016/j.cor.2005.06.005.

Pedro Amorim and Bernardo Almada-Lobo. The impact of food perishability issues in the vehicle routing problem. *Computers & Industrial Engineering*, 67:223–233, 2014.

Sundararajan Arunapuram, Kamlesh Mathur, and Daniel Solow. Vehicle routing and scheduling with full truckloads. *Transportation Science*, 37(2):170–182, 2003.

Hossein Baharmand, T. Comes, and Matthieu Lauras. Managing in-country transportation risks in humanitarian supply chains by logistics service providers: Insights from the 2015 Nepal earthquake. *International Journal of Disaster Risk Reduction*, 07 2017. doi: 10.1016/j.ijdrr.2017.07.007.

André J Balaton, Cinthia B Drachenberg, Cheryl Rucker, Philippe Vaury, and John C Papadimitriou. Satisfactory performance of primary antibodies beyond manufacturers' recommended expiration dates. *Applied Immunohistochemistry & Molecular Morphology*, 7 (3):221, 1999.

Rajan Batta and Samuel S. Chiu. Optimal obnoxious paths on a network: Transportation of hazardous materials. *Operations Research*, 36(1):84–92, 1988.

Aharon Ben-Tal, Byung Do, Supreet Reddy, and Tao Yao. Robust optimization for emergency logistics planning : Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B*, 45:1177–1189, 2011.

Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202:8–15, 04 2010a. doi: 10.1016/j.ejor.2009.04.024.

Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15, 2010b.

John R. Birge and Franois Louveaux. *Introduction to Stochastic Programming.* Springer Publishing Company, Incorporated, 2nd edition, 2011. ISBN 1461402360.

Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.

Margaret Chan, Anthony Lake, Anthony Fauci, Seth Berkley, Joy Phumaphi, Christopher Elias, Pedro Alonso, Ciro Quadros, Nicole Bates, Zulfiqar Bhutta, Lola Dare, Helen Evans, Lee Hall, T. John, Jean-Marie Okwo-Bele, Orin Levine, David Salisbury, Anne Schuchat, Peter Singer, and Sandy Wrobel. Global vaccine action plan. *Vaccine*, 31:B5–B31, 04 2013.

Huey-Kuo Chen, Che-Fu Hsueh, and Mei-Shiang Chang. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5):383–408, 2006.

Huey-Kuo Chen, Che-Fu Hsueh, and Mei-Shiang Chang. Production scheduling and vehicle routing with time windows for perishable food products. *Comput. Oper. Res.*, 36(7): 2311–2319, 2009. ISSN 0305-0548.

Leandro C. Coelho and Gilbert Laporte. Optimal joint replenishment, delivery and inventory management policies for perishable products. *Comput. Oper. Res.*, 47:42–52, July 2014.

Jean-Francois Cordeau, Michel Gendreau, Gilbert Laporte, Jean-Yves Potvin, and Frédéric Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research society*, 53(5):512–522, 2002.

Yves Crama, Mahmood Rezaei, Martin Savelsbergh, and Tom Van Woensel. Stochastic Inventory Routing for Perishable Products. *Transportation Science Publication*, 52:526–546, 2018.

Jacques Desrosiers, Yvan Dumas, Marius M. Solomon, and François Soumis. Time Constraint Routing and Scheduling. *Handbooks in Operations Research and Management Science*, 8: 35–139, 1995.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec 1959.

Moshe Dror and Pierre Trudeau. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research*, 23(2):228–235, February 1986.

John Forrest, Ted Ralphs, Stefan Vigerske, Lou Hafer, Bjarni Kristjansson, J P Fasano, Edwin Straver, Miles Lubin, Haroldo Gambini Santos, Robin Lougee, and Matthew Saltzman. COIN-OR/CBC: Version 2.9.9, 07 2018. URL https://zenodo.org/record/1317566\#.XfbGz-hKg2w.

Song Gao and Ismail Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, 40:93–122, 2006.

Hermann Gehring and Jörg Homberger. A parallel two-phase metaheuristic for routing problems with time-windows. *Asia Pacific Journal of Operational Research*, 18(1):35–48, 2001.

Michel Gendreau, Gilbert Laporte, and Seguin Rene. Stochastic vehicle routing. *European Journal of Operational Research*, 88:3–12, 1996.

Michel Gendreau, François Guertin, Jean-Yves Potvin, and Éric Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33:381–390, 11 1999. doi: 10.1287/trsc.33.4.381.

GNU. GNU linear programming kit solver, 2019. URL https://www.gnu.org/software/glpk/.

Asvin Goel and Stefan Irnich. An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science*, 51(2):737–754, 2017.

Bram L. Gorissen, Ihsan Yanıkoğlu, and Dick den Hertog. A Practical Guide to Robust Optimization. *Omega*, 53:124–137, 2015.

Chrysanthos E Gounaris, Wolfram Wiesemann, and Christodoulos A. Floudas. The Robust Capacitated Vehicle Routing Problem Under Demand Uncertainty. *Operations Research*, 61(3):677–693, 2013.

Alex Grasas, Helena R. Lourenço, Luciana S. Pessoa, M. G. C. Resende, Imma Caballé, and Nuria Barba. On the improvement of blood sample collection at clinical laboratories. *BMC health services research*, 14:12, 01 2014. doi: 10.1186/1472-6963-14-12.

Manfred Gronalt, Richard F Hartl, and Marc Reimann. New savings based algorithms for time constrained pickup and delivery of full truckloads. *European Journal of Operational Research*, 151(3):520–535, 2003.

Mario Guajardo. Six finalists vie for the IFORS Prize for OR in Development 2020. *International Federation of Operational Research Societies*, 03 2020. URL https://www.ifors.org/newsletter/ifors-news-march-2020.pdf.

Ali R Güner, Alper Murat, and Ratna Babu Chinnam. Dynamic routing under recurrent and non-recurrent congestion using real-time its information. *Computers & Operations Research*, 39(2):358–373, 2012.

Gurobi. Mixed-integer programming (MIP), A primer on the basics, 2019. URL https://www.gurobi.com/resource/mip-basics.

Masoud Hamedi, Ali Haghani, and Saini Yang. Reliable Transportation of Humanitarian Supplies in Disaster Response: Model and Heuristic. *Procedia - Social and Behavioral Sciences*, 54:1205–1219, 2012.

Celina M Hanson, Anupa M George, Adama Sawadogo, and Benjamin Schreiber. Is freezing in the vaccine cold chain an ongoing issue ? A literature review q. *Vaccine*, 35:2127–2133, 2017. doi: 10.1016/j.vaccine.2016.09.070.

Hideki Hashimoto and Mutsunori Yagiura. A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 254–265. Springer, 2008.

Jörg Homberger and Hermann Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1): 220–238, 2005.

Maria Camila Hoyos, Ridley S Morales, and Raha Akhavan-tabatabaei. OR models with stochastic components in disaster operations management : A literature survey q. *Computers and Industrial Engineering*, 82:183–197, 2015.

Chaug-Ing Hsu, Sheng-Feng Hung, and Hui-Chieh Li. Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering*, 80:465–475, 05 2007. doi: 10.1016/j.jfoodeng.2006.05.029.

Jonas Ide and Anita Schöbel. Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. *OR Spectrum: Quantitative Approaches in Management*, 38(1):235–271, January 2016.

Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, and David Pisinger. A non-robust branch-and-cut-and-price algorithm for the vehicle routing problem with time windows. *Oper. Res. Forthcoming*, 2006.

Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189:293–309, 2008.

Patricia Kellicker. Drug expiration dates: How accurate are they? *EBSCO Publishing, Ipswich, MA.(http://healthlibrary. epnet. com). Kervinen, L. and Yliruusi, J.(1993). Modeling S-shaped dissolution curves. International Journal of Pharmaceutics*, 92:115–122, 2006.

Niklas Kohl and Oli Madsen. An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation. *Operations Research*, 45:395–406, 1997.

Antoon WJ Kolen, AHG Rinnooy Kan, and Harry WJM Trienekens. Vehicle routing with time windows. *Operations Research*, 35(2):266–273, 1987.

George Kontoravdis and Jonathan Bard. A GRASP for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 7:10–23, 02 1995. doi: 10.1287/ijoc.7.1.10.

Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. 2012.

Gilbert Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

Gilbert Laporte and Francois V Louveaux. An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. *Operations Research*, 50(3): 415–423, 2002.

Gilbert Laporte, François V. Louveaux, and Hélène Mercure. The vehicle routing problem with stochastic travel times. *Transportation Science*, 26:161–170, 1992.

Opeyeolu Timothy Laseinde and Khumbulani Mpofu. Providing solution to last mile challenges in postal operations. *International Journal of Logistics Research and Applications*, 20(5):475–490, 2017a.

Opeyeolu Timothy Laseinde and Khumbulani Mpofu. Providing solution to last mile challenges in postal operations. *International Journal of Logistics Research and Applications*, 20(5):475–490, 2017b.

Bruce Y Lee, Leila A Haidari, Wendy Prosser, Diana L Connor, Ruth Bechtel, Amelia Dipuve, Hidayat Kassim, Balbina Khanlawia, and Shawn T Brown. Re-designing the Mozambique vaccine supply chain to improve access to vaccines. *Vaccine*, 34:4998–5004, 2016.

Chungmok Lee, Kyungsik Lee, and Sungsoo Park. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9):1294–1306, 2012.

Jing-Quan Li, Pitu B Mirchandani, and Denis Borenstein. A lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):419–433, 2009a.

Jing-Quan Li, Pitu B Mirchandani, and Denis Borenstein. Real-time vehicle rerouting problems with time windows. *European Journal of Operational Research*, 194(3):711–727, 2009b.

J J Lin, Chi Y C, and B F Wang. Improved algorithms for the continuous tree edge-partition problems and a note on ratio and sorted matrices searches. *Discrete Applied Mathematics*, 158(8):932–942, 2010.

George F. List, Pitu B. Mirchandani, Mark A. Turnquist, and Konstantinos G. Zografos. Modeling and Analysis for Hazardous Materials Transportation: Risk Analysis, Routing/Scheduling and Facility Location. *Transportation Science*, 25(2):100–114, 1991.

Ran Liu, Zhibin Jiang, Richard YK Fung, Feng Chen, and Xiao Liu. Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Computers & Operations Research*, 37(5):950–959, 2010.

Wapee Manopiniwes and Takashi Irohara. A Review of Relief Supply Chain Optimization. *Industrial Engineering and Management Systems*, 13(1):1–14, 2014.

Kipp Martin. Tutorial: COIN-OR: Software for the OR community. *Interfaces*, 40(6):465–476, 2010. ISSN 00922102, 1526551X.

Olaf Mersmann, Bernd Bischl, Jakob Bossek, Heike Trautmann, Markus Wagner, and Frank Neumann. Local search and the traveling salesman problem: A feature-based characterization of problem hardness. In Youssef Hamadi and Marc Schoenauer, editors, *Learning and Intelligent Optimization*, pages 115–129, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Huseyin Onur Mete and Zelda B. Zabinsky. Stochastic optimization of medical supply location and distribution in disaster management. *International Journal of Production Economics*, 126(1):76–84, 2010.

Qianxin Mu, Zhuo Fu, Jens Lysgaard, and Richard Eglese. Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society*, 62(4):742–749, 2011.

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

Anna Nagurney, Amir H. Masoumi, and Min Yu. Supply chain network operations management of a blood banking system with cost and risk minimization. *Computational Management Science*, 9(2):205–231, 2012.

Steven Nahmias. Perishable inventory theory: A review. *Operations research*, 30:680–708, 08 1982.

Jakob Nielsen and Robert L. Mack, editors. *Usability Inspection Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1994. ISBN 0-471-01877-5.

Pamela C. Nolz, Frédéric Semet, and Karl F. Doerner. Risk approaches for delivering disaster relief supplies. *OR Spectrum*, 33(3):543–569, 2011.

James B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129, Aug 1997.

Ana Osvald and Lidija Stirn. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering*, 85:285–295, 03 2008. doi: 10.1016/j.jfoodeng.2007.07.008.

Jorge Oyola, Halvard Arntzen, and David L Woodruff. The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics*, 7(3):193–221, 2018.

Linet Özdamar and Mustafa Alp Ertem. Models, solutions and enabling technologies in humanitarian logistics. *European Journal of Operational Research*, 244(1):55–65, 2015.

Minnie H. Patel and Alan J. Horowitz. Optimal routing of hazardous materials considering risk of spill. *Transportation Research Part A: Policy and Practice*, 28(2):119–132, 1994.

Diego Pecin, Claudio Contardo, Guy Desaulniers, and Eduardo Uchoa. New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29(3):489–502, 2017.

Larissa P. G. Petroianu. RoOT dataset, 2019a. URL https://github.com/lpetroia/RoOT.

Larissa P. G. Petroianu. RoOT dataset, 2019b. URL https://github.com/lpetroia/RoOT-portugues.

Larissa P. G. Petroianu, Zelda B. Zabinsky, and Mauricio G. C. Resende. Distribution of perishable products under uncertainty. (submitted), 2020a.

Larissa P. G. Petroianu, Zelda B. Zabinsky, Mariam Zameer, Yi Chu, Mamiza M. Muteia, Mauricio G. C. Resende, Aida L. Coelho, Jiarui Wei, Turam Purty, Abel Draiva, and Alvaro Lopes. A light-touch tool for optimal vaccine distribution in Mozambique. (submitted), 2020b.

Larissa P.G. Petroianu, Zelda B. Zabinsky, and Mauricio G.C. Resende. A stochastic program with recourse for distributing perishable medical supplies. In *Proceedings of the 2019 IISE Annual Conference*, 2019.

Warren B Powell. A stochastic model of the dynamic vehicle allocation problem. *Transportation science*, 20(2):117–129, 1986.

Warren B Powell. An operational planning model for the dynamic vehicle allocation problem with uncertain demands. *Transportation Research Part B: Methodological*, 21(3):217–232, 1987.

Warren B Powell. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science*, 30(3):195–219, 1996.

Eric Prescott-Gagnon, Guy Desaulniers, and Louis-Martin Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks: An International Journal*, 54(4):190–204, 2009.

Vitoria Pureza, Reinaldo Morabito, and Marc Reimann. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the vrptw. *European Journal of Operational Research*, 218(3):636–647, 2012.

Katrien Ramaekers and Gerrit Janssens. On the choice of a demand distribution for inventory management models. *European Journal of Industrial Engineering*, 2:479–491, 02 2008. doi: 10.1504/EJIE.2008.018441.

Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231, 2016.

F Frank Saccomanno and AY-W Chan. Economic evaluation of routing strategies for hazardous road shipment. *Transp. Res. Rec.*, 1020:12–18, 1985.

Nikolaos V. Sahinidis. Optimization under uncertainty: State-of-the-art and opportunities. *Computers and Chemical Engineering*, 28:971–983, 2004.

David Sarley, Mustafa Mahmud, Jide Idris, Modele Osunkiyesi, Onome Dibosa-osadolor, Peter Okebukola, and Owens Wiwa. Transforming vaccines supply chains in Nigeria q. *Vaccine*, 35:2167–2174, 2017.

Amanda J Schmitt, Lawrence V Snyder, and Zuo-Jun Max Shen. Inventory systems with stochastic demand and supply: Properties and approximations. *European Journal of Operational Research*, 206(2):313–328, 2010.

Geoffrey De Smet. *OptaPlanner VRP examples: Belgium 2017 dataset.* Red Hat and the community, 2017. URL https://www.optaplanner.org.

Abby Snyder, Larissa P.G. Petroianu, and Zelda B. Zabinsky. Route visualization for efficient vaccine distribution in Mozambique. Poster at the University of Washington Undergraduate Research Symposium, 2020.

Marius M Solomon. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2):254–265, 1987.

Marius M. Solomon and Jacques Desrosiers. Survey Paper–Time window constrained routing and scheduling problems. *Transportation Science*, 22(1):1–13, 1988.

Byung Song and Young Dae Ko. A vehicle routing problem of both refrigerated- and general-type vehicles for perishable food products delivery. *Journal of Food Engineering*, 169:61–71, 01 2016.

Ilgaz Sungur, Fernando Ordóñez, and Maged Dessouky. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40 (5):509–523, 2008.

Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31:170–186, 05 1997. doi: 10.1287/trsc.31.2.170.

Frank A Tillman. The Multiple Terminal Delivery Problem with Probabilistic Demands The Multiple Terminal Delivery Problem with Probabilistic Demands. *Transportation Science*, 3(3):192–204, 1969.

S Tofighi, S Ali Torabi, and S Afshin Mansouri. Humanitarian logistics network design under mixed uncertainty. *European Journal of Operational Research*, 250(1):239–250, 2016.

Transportation Optimization Portal. Gehring and homberger benchmark, 2008. URL https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/.

Luk N Van Wassenhove. Humanitarian aid logistics: supply chain management in high gear. *Journal of the Operational research Society*, 57(5):475–489, 2006.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi, and Walter Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60:611–624, 06 2012a. doi: 10.1287/opre.1120.1048.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231, 01 2012b. doi: 10.1016/j.ejor.2013.02.053.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21, 2013a.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research*, 40(1):475–489, 2013b.

VillageReach. VillageReach website, 2019. URL https://www.villagereach.org/about. Accessed: 2019-09-23.

Begoña Vitoriano, Javier Montero, and Da Ruan. *Decision aid models for disaster management and emergencies*. Springer Science & Business Media, 2013.

Yibing Wang, Wei Dong, Liangqi Zhang, David Chin, Markos Papageorgiou, Geoffrey Rose, and William Young. Speed modeling and travel time estimation based on truncated normal and lognormal distributions. *Transportation Research Record-Series*, 2315:66 – 72, 2012. ISSN 0361-1981. doi: 10.3141/2315-07.

Zheng Wang, Ying Li, and Xiangpei Hu. A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. *Computers and Industrial Engineering*, 89, 11 2014. doi: 10.1016/j.cie.2014.11.004.

Wen-Huei Yang, Kamlesh Mathur, and Ronald Ballou. Stochastic vehicle routing problem with restocking. *Transportation Science*, 34:99–112, 02 2000. doi: 10.1287/trsc.34.1.99.12278.

Zelda B. Zabinsky, Pattamon Dulyakupt, Shabnam Zangeneh-Khamooshi, Cao Xiao, Pengbo Zhang, Seksan Kiatsupaibul, and Joseph A. Heim. Optimal collection of medical specimens and delivery to central laboratory. *Annals of Operations Research*, pages 1–28, 2019a.

Zelda B Zabinsky, Ting-Yu Ho, and Hao Huang. Integrating heuristics and approximations into a branch and bound framework. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 774–779. IEEE, 2019b.

Junlong Zhang, William H K Lam, and Bi Yu. On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows. *European Journal of Operational Research*, 249:144–154, 2016.

Lin Zhou, Roberto Baldacci, Daniele Vigo, and Xu Wang. A Multi-Depot Two-Echelon Vehicle Routing Problem with Delivery Options Arising in the Last Mile Distribution. *European Journal of Operational Research*, 265:765–778, 2018.

Hossein Zolfagharinia and Michael Haughton. Effective truckload dispatch decision methods with incomplete advance load information. *European Journal of Operational Research*, 252 (1):103–121, 2016.

Hossein Zolfagharinia and Michael A Haughton. Operational flexibility in the truckload trucking industry. *Transportation Research Part B: Methodological*, 104:437–460, 2017.