COMPUTATIONAL METHODS TO OPTIMIZE HIGH-CONSEQUENCE VARIANTS OF THE VEHICLE ROUTING PROBLEM FOR RELIEF NETWORKS IN HUMANITARIAN LOGISTICS

Joshua Charles Urbanovsky

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

August 2018

APPROVED:

Armin R. Mikler, Major Professor
Robert Renka, Committee Member
Chetan Tiwari, Committee Member
Bill Buckles, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Yan Huang, Interim Dean of the College of Engineering
Victor Prybutok, Dean of the Toulouse Graduate School

Urbanovsky, Joshua Charles. *Computational Methods to Optimize High-Consequence Variants of the Vehicle Routing Problem for Relief Networks in Humanitarian Logistics*. Doctor of Philosophy (Computer Science and Engineering), August 2018, 128 pp., 9 tables, 21 figures, 62 numbered references.

Optimization of relief networks in humanitarian logistics often exemplifies the need for solutions that are feasible given a hard constraint on time. For instance, the distribution of medical countermeasures immediately following a biological disaster event must be completed within a short time-frame. When these supplies are not distributed within the maximum time allowed, the severity of the disaster is quickly exacerbated. Therefore emergency response plans that fail to facilitate the transportation of these supplies in the time allowed are simply not acceptable. As a result, all optimization solutions that fail to satisfy this criterion would be deemed infeasible. This creates a conflict with the priority optimization objective in most variants of the generic vehicle routing problem (VRP). Instead of efficiently maximizing usage of vehicle resources available to construct a feasible solution, these variants ordinarily prioritize the construction of a minimum cost set of vehicle routes.

Research presented in this dissertation focuses on the design and analysis of efficient computational methods for optimizing high-consequence variants of the VRP for relief networks. The conflict between prioritizing the minimization of the number of vehicles required or the minimization of total travel time is demonstrated. The optimization of the time and capacity constraints in the context of minimizing the required vehicles are independently examined. An efficient meta-heuristic algorithm based on a continuous spatial partitioning scheme is presented for constructing a minimized set of vehicle routes in practical instances of the VRP that include critically high-cost penalties. Multiple optimization priority strategies that extend this algorithm are examined and compared in a large-scale bio-emergency case study. The algorithms designed from this research are implemented and integrated into an existing computational framework that

is currently used by public health officials. These computational tools enhance an emergency response planner's ability to derive a set of vehicle routes specifically optimized for the delivery of resources to dispensing facilities in the event of a bio-emergency.

ACKNOWLEDGMENTS

This dissertation was in part supported by the contributions and collaborations of the Texas Department of State Health Services (DSHS) and the Tarrant County Public Health Department (TCPHD). The research described in this dissertation was additionally supported by Grant Number 1R01LM011647-01 and Grant Number 1R15LM010804-01 from the National Institutes of Health (NIH).

I would like to thank my parents, Joe and Connie Urbanovsky, for their endless love and unconditional support. No matter my ambitions they have always done everything possible to provide me with an opportunity to succeed. Without my parents, none of my achievements would have been possible. I am very grateful for the daily encouragement and loving support I received from Cassie. With her by my side, I have never been afraid to fail. She constantly motivates me to settle for nothing less than my greatest hopes and dreams. I am also appreciative of my little brothers, Luke and Nathan, who have helped me remember to not sweat the small things and to instead appreciate what is really important in life. I also want to thank my two puppies, Neiko and Loki, who make me laugh every single day, especially when I need it the most.

I would like to thank the members of my Ph.D. committee, who have provided valuable feedback on my research. In particular, I would like to thank my advisor, Dr. Armin Mikler, for all of his support, guidance, and encouragement throughout my entire pursuit of a doctoral degree. I am very grateful for the countless research opportunities he has provided. I am also thankful to Dr. Marty O'Neill II and Dr. Saratchandra Indrakanti for the knowledge and guidance they provided me. The experience and lessons they shared with me from their experience in the Ph.D. program were more valuable than I could ever express. I would also like to recognize all the Center for Computational Epidemiology and Response Analysis members at UNT whom I have had the amazing opportunity to work with.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A wide span of applications in logistics and distribution management are modeled by variations of the general vehicle routing problem (VRP). The VRP is simply described as defining the optimal delivery routes denoted by an ordered list of customers or locations the fleet of vehicles must visit. This extent of practical applications has resulted in significant progress of the exploration and optimization in countless aspects of this problem. Accordingly, theses variants of the VRP define the optimization objectives for a specific application. The *Open* VRP (OVRP) is a notable variant that has gained interest in recent years. Contrary to the VRP where vehicle tours are required to begin and end at a specified location (depot), the OVRP allows tours to end after the last stop (customer) in its tour is served.

A core focus in this research is the logistical optimization of practical applications that involve critically high-cost penalties. Instances associated with these type penalties are often constrained by a strict time allowed for vehicles to complete a tour within. For this reason, variants of the OVRP are used to model these type of instances. Instead of reducing the sum total cost of completing all tours for a known vehicle fleet size, the primary objective is to find the minimum number of vehicles required to serve all customers without a single route surpassing its allowed time constraint. Accordingly, we denote this variation as the Open Vehicle Routing Problem with *Uniform Time* constraints (OVRP-UT) for the purpose of distinguishing it from existing related variants (e.g., the OVRP with Time-Windows (OVRP-TW)). Unlike the OVRP-TW, the time constraint defined in the OVRP-UT is unvarying across the entire scenario such that for any solution to be feasible, all tours must be completed within the time permitted.

Optimization of relief networks in humanitarian logistics often exemplify the need for solutions that are feasible given a hard constraint on time. The distribution of medical countermeasures immediately following a disaster represents one instance for which adherence to this constraint is critically important. Additionally this classification of logistics are often further constrained by a limited number of resources available, which are often distributed over large regions. These constraints are clearly emphasized in an epidemic disaster. For instance, if the response following the release of a biological agent such as Antrax, antimicrobials (i.e., ciprofloxacin, doxycycline, amoxicillin, penicillin, etc...) must be dispensed to the affected population within 48 hours to treat and prevent illness effectively [1]. Optimizing these logistical relief networks are only of use when the resulting solution is deemed feasible. Of course, the properties required for a solution to demonstrate feasibility must be explicitly defined by the scenario as they are otherwise ambiguous. Accordingly, these properties must be expressed through constraints modeled by the optimization objectives when applying VRP algorithms within this domain.

Practical instances of the VRP corresponding to these high-consequence constraints create challenging issues that uniquely define the main optimization objective. To the best of our knowledge however, even the most powerful existing heuristics for both exact and approximation algorithms are not specifically designed with these objectives in mind. As a result, they will often fall short in optimizing variants of OVRP-UT. Furthermore performance evaluations for heuristics are traditionally evaluated based on solving standardized sets of problems (i.e., benchmarks). Consequently when heuristics are adjusted to fit the individual instance or the unique difficulties defined by a specific problem set, it can decrease the robustness of the algorithm. This conflict is exemplified in Section 1.1, substantiating the design of heuristics prioritizing the objective of minimizing the number of vehicles requires.

With this in mind, we analyze the attributes that define the OVRP-UT and the variants representing practical instances such as in a bio-emergency. Computational methods driven from these insights are then presented for the optimization of these high-consequence variants of the VRP in relief networks for humanitarian logistics.

Section 1.2 states the overall organization of this dissertation. In particular the objective for each chapter is identified, including its associated to the chapters that follow. The specific contributions resulting from this dissertation are described in Section 1.3. Section 1.2.

This section presents an independent and simplistic example demonstrating the conflicting core objectives between most variants of the OVRP and the OVRP-UT. In Chapter 3 we introduce a mathematical formulation based on the objective of minimizing the fleet size. Using a generalization of the OVRP-UT we further present observations of the search space in deriving an optimal solution and motivate a partitioning approach to reduce the computational complexity.

When heuristics are developed for the VRP they are often described when solving a generalized variant or usually the classical VRP itself. This practice is common as it allows a single algorithm to be utilized for many variants of the problem for a specific application. Furthermore it reduces the complexity of implementation and reproduction of the method originally introduced. Even when more constraints or specific bounds and properties are introduced, the overall objective is the same: *to minimize the total travel cost to visit all customers with a known number of vehicles*. However, we argue that the change to minimize the required vehicles for the OVRP-UT, instead of minimizing the total travel time, are not as related as they might appear.

To demonstrate the conflict in solutions between the two objects: (i) Minimize total cost, and (ii) Minimize fleet size; Figure 1.1 represents a simple 5 node graph (1 depot, 4 customers) constructed that highlights this conflict. In this construction for all VRP variants, we assume that a feasible solution must have a set of vehicle routes that start at the depot, $\{n_0\}$, and visit (in any order) each of the the customers, $\{n_1, n_2, n_3, n_4\}$, once at some point within the resulting set of routes. Additionally we assume that all vehicle tours (VT) in each solution must be completed within some known maximum time constraint, where the edge cost in the graph represent the symmetric cost of travel by time between the connected nodes.

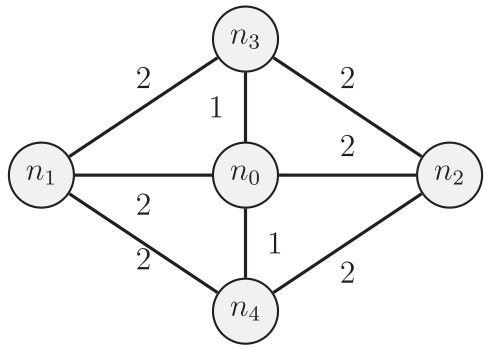


FIGURE 1.1. Simple network with conflicting solutions based on the optimization goal. The node $\{n_0\}$ represents the starting location (depot) for all tours, and the nodes $\{n_1, n_2, n_3, n_4\}$ are the customers required to be visited for any feasible solution.

In Table 1.1 the optimal solutions related to the graph shown in Figure 1.1 are listed for both objectives. This demonstrates a clear conflict between what their formulation has determined to be *optimal*. For both objectives included in this table, the solutions allow the tour to end without returning to the depot (i.e., OVRP, OVRP-UT). Additionally every tour within each solution is one that is feasible. Namely, the cost of each individual tour within each of the solutions listed are under the max time constraint, 7. The formulation of the OVRP usually requires the number of vehicles to be known. Therefore the optimal solutions listed for this variant under the *Minimize total cost* objective can be viewed as if an exact algorithm was iteratively executed with an arbitrarily large fleet size initially but was reduced at each iteration. Then the solutions listed represent the parameters and resulting tour(s) to the iteration produced the minimum sum total cost of tours that were all under the provided time constraint.

Objective:	**Minimizing sum total cost**	**Minimizing number of vehicles**
VRP Variant*:	OVRP	OVRP-UT
Vehicle Tours (VT):	VT_01: $[n_0, n_1]$ VT_02: $[n_0, n_2]$ VT_03: $[n_0, n_3]$ VT_04: $[n_0, n_4]$	VT_01: $[n_0, n_3, n_1, n_4, n_2]$
Fleet Size:	4	1
Total Cost Of Tour(s):	6	7

**All tours are restricted to a maximum cost of 7*

Table 1.1: The resulting optimal tours of conflicting optimization objectives: (i) minimizing sum total cost, and (ii) minimizing number of vehicles

The difference between the two objectives is clearly visible by comparing their respective optimal solutions from this example. The OVRP formulation produces a solution that is optimal for the objective of minimizing the total cost. In particular it is observed that the sum total costs of the tours for the OVRP variant is lower than the total costs for the OVRP-UT. Conversely, the OVRP-UT variant produces an optimal solution for the objective of minimizing the fleet size. From this example, it is evident that heuristics designed for the objective of minimizing total cost can produce solutions that are not optimal for the objective of minimizing the vehicle fleet size.

1.2. Overview of the Dissertation

The focus of this dissertation is the design of efficient computational methods for optimizing high-consequence variants of the VRP for relief networks in humanitarian logistics. As illustrated by this introductory chapter, problem instances classified by logistics of this type often introduce optimization objectives who differ from those of existing variants of the OVRP. Chapter 2 provides the necessary background information to the logistics in emergency response planning. Specifically this is centered around the context of the distribution of medical countermeasures immediately following a biological disaster event. The chapter further reviews the existing literature for the VRP problem and its current exact and approximation algorithms. Additionally, the existing VRP research in emergency response is reviewed.

With the intention to integrate the resulting computation methods from this research into an accessible software framework for emergency response planners, the design of highly efficient algorithms are of interest. To that end, the design of a meta-heuristic algorithm based on a continuous spatial partitioning scheme is presented in Chapter 6. This algorithm constructs a minimized set of vehicle routes for variants of the VRP that are defined by the high-consequence constraints in a bio-emergency scenario. Notably, the hard constraints for the maximum tour duration and the identical vehicle capacity are prioritized in this optimization.

Due to the conflict between the objectives of the minimization of total vehicles and minimizing the total travel time, the optimization of the time and capacity constraints in the context of minimizing vehicles are independently examined. The design of this algorithm is thus preceded by the independent study of these constraints. Furthermore, the context of these studies are structured around continuous partitioning heuristics to allow a consistent association in the design of the resulting algorithm. Accordingly, the structure of this dissertation is as follows: Chapter 3 introduces an abstract strategy for determining the minimum number of vehicles that feasibly solve the OVRP-UT (i.e., only the time constraint is considered while vehicle capacity is disregarded). As consequence of the complexities quantified by this chapter, methods to estimate the lower bound of vehicles required without an explicitly defined solution is introduced in Chapter 4. Chapter 5 similarly introduces a bounding procedure for the minimum fleet size, but instead is based around the OVRP-UC (i.e., only the vehicle capacity is considered while the time constraints are disregarded). Chapter 6 then formally defines an optimization problem for the high-consequence VRP variant that models a bio-emergency response. Additionally included in this chapter is the design and theoretical verification of the aforementioned approximation algorithm constructed to solve this newly defined problem. Chapter 7 illustrate the direct application of this two-phase algorithm in the context of bio-emergency response. Namely, a case study is constructed from a large region in the state of Texas to exemplify the performance and outcomes of the algorithm. Multiple priority optimization strategies are also introduced in this chapter and their associated performance is compared and examined accordingly using the regional case-study. Chapter 8 concludes this dissertation by identifying the limitations of the work presented. Future work for improvements and expansions of this research are additionally identified.

1.3. Research Contributions

The investigation and results from this research make the following key scientific contributions:

(i) Formulation of mathematical models that represent the VRP variants associated with high-consequence constraints in optimizing relief networks for humanitarian logistics

(ii) Identify and analyze the conflicting objectives between minimizing the required vehicles and minimizing the total travel time

(iii) Investigate a strategy based on an continuous partitioning scheme to determine the minimum number of required vehicles for the OVRP-UT

(iv) Formally identify the complexities in finding the minimum number of required vehicles for any instance of the OVRP-UT; design techniques for estimating the lower bound without explicitly defining the solution

(v) Design and implement an efficient and flexible meta-heuristic algorithm to solve variants of the OVRP-UT that are defined by multiple high-consequence constraints; in particular for distributing medical countermeasures to mitigate an epidemic disaster

(vi) Design a upper bounding procedure for the OVRP-UC based on maximizing an identical capacity usage; prove its maximum error a worst-case scenario

Furthermore, the algorithms resulting from this research have been implemented and integrated into an existing computational framework designed for public health officials [51]. These computational tools enhance an emergency response planner's ability to devise a set of vehicle routes specifically optimized for the delivery of resources to dispensing facilities in the event of a bio-emergency.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

2.1. Background

History has repeatedly demonstrated that throughout the entire world, disasters will not only occur but can enforce some of the most deadly consequences humanity has and will ever face. The frequency of occurrence has brought awareness to the significant impact that arise from many types disasters and crisis situations, but most prominently it has shown that there is little uncertainty in predicting if another will occur but instead is only a question of when.

The complexity of responding to and mitigating a disaster varies drastically depending upon the type of disaster and the populations it affects. Often some of the most consequential factors that increase this complexity are when the affected population is large and sparsely distributed geographically. Different types of disasters can include this symptom of large populations being affected such as natural disasters. As shown in Table 2.1, natural disasters represent a grouping of disasters that span areas of geophysical, meteorological, hydrological, climatological, biological, and extraterrestrial disasters.

Disaster Group	Disaster Subgroup	Definition	Disaster Main Type
Natural	Geophysical	A hazard originating from solid earth. This term is used interchangeably with the term geological hazard.	Earthquake
			Mass Movement
			Volcanic Activity
	Meteorological	A hazard caused by short-lived, micro- to meso-scale extreme weather and atmospheric conditions that last from minutes to days.	Extreme Temperature
			Fog
			Storm
	Hydrological	A hazard caused by the occurrence, movement, and distribution of surface and subsurface freshwater and saltwater.	Flood
			Landslide
			Wave Action
	Climatological	A hazard caused by long-lived, meso- to macro-scale atmospheric processes ranging from intra-seasonal to multi-decadal climate variability.	Drought
			Glacial Lake Outburst
			Wildfire
	Biological	A hazard caused by the exposure to living organisms and their toxic substances (e.g. venom, mold) or vector-borne diseases that they may carry. Examples are venomous wildlife and insects, poisonous plants, and mosquitoes carrying disease-causing agents.	Epidemic
			Insect Infestation
			Animal Accident
	Extraterrestrial	A hazard caused by asteroids, meteoroids, and comets as they pass near-earth, enter the Earths atmosphere, and/or strike the Earth, and by changes in interplanetary conditions that effect the Earths magnetosphere, ionosphere, and thermosphere.	Impact
			Space Weather

Table 2.1: Natural disasters categorized by the Emergency Events Database (EM-DAT) [29]

Each of the main types of disasters (e.g., storm, earthquake, epidemic, landslide) defined in Table 2.1 will introduce unique complexities into responding to the disaster, including the challenges imposed by funding the appropriate resources needed. For example, in 2004 New Orleans participated in a simulated exercise where a hypothetical category 3 hurricane named Pam would cause significant damage, allowing knowledge to be gained from what it would take to adequately respond this type of disaster. However just a short time later in 2005, the category 5 hurricane Katrina arrived on the Gulf coast and quickly became one of the most deadly storms in the history of the United States. Although the earlier hurricane simulation of Pam had accurately predicted the impact that such an event might have, poor allocation of funds hindered the mitigation process. Further, even with the knowledge that Katrina was approaching, difficulties added by the flooding and mass evacuations left emergency responders without many critical resources needed [49]. Still, even with the unique attributes of each main type of natural disaster, there exists similarities between the complexities that are introduced into the emergency response, such as the large populations and geographic areas that need to be treated or delivered resources.

Year	**Disaster type**	**Total deaths**	**Total affected**	**Total damage**
2005	Storm	1,852	830,000	157,530,000
2011	Storm	590	18,593	27,000,000
2002	Epidemic	214	3,624	N/A
2006	Extreme temperature	188	N/A	N/A
2007	Storm	167	1,377	4,600,000
2008	Storm	156	2,300,400	39,540,000

Table 2.2: Top 5 natural disasters by deaths from 2000-2015 in the USA. Units for the Total Damage are given in US $ (in thousands) in the value of the year of occurrence. [29]

Natural disasters have continued to draw attention to their severity in consequences they bring. According to the results from a query of the Emergency Events Database (EM-DAT), from 2000 to 2015 in the United States, natural disasters alone caused 6,287 deaths and over $567 billion dollars in estimated damage. The most deadly of these natural disasters were caused by either storms, an epidemic, or extreme temperatures, as shown in Table 2.2.

The statistics and frequency from these previous events clearly show that emergency response planning can be an integral part to mitigation efforts. This notion was further highlighted in the White House report following the devastation caused by hurricane Katrina in 2005, requesting that the capabilities of the Department of Health and Human Services (HHS) be significantly strengthened for supporting and coordinating services during a crisis situation. Following this report, in December of 2006 the Pandemic and All-Hazards Preparedness Act (PAHPA) (Public Law No. 109-417) was passed by the U.S Congress to improve the nations abilities to prepare and respond to these disaster scenarios [7].

Disaster preparedness specifically highlights volatile environments that can easily amplify detrimental consequences that result from reducing the ability to respond to an emergency situation when the data and assumption initially incorporated changed or did not accuracy reflect the situation. Emergency response plans designed by public officials to mitigate negative outcomes to a situation such as a bio-emergency, will immediately be put into action with little to no flexibility for further adjustment. Therefore the stability and reliability of any emergency response plan under these types of environments must incorporate and model these characteristics into their configuration of an optimal plan. The following section provides a brief overview of the process and attributes of preparing and responding to emergencies.

2.1.1. Emergency Response Planning

Effectively mitigating a disaster is often the result of exhaustive emergency response planning by public officials. The importance of this planning process and the activities involved are authorized from the national level by the PAHPA in 2006 and then improved upon in the reauthorization of this law as the Pandemic and All-Hazards Preparedness Reauthorization Act of 2013 [44]. The key items in this act are as follows:

(i) Strengthening national preparedness and response for public health emergencies

(ii) Optimizing state and local all-hazards preparedness and response

(iii) Enhancing medical countermeasure review

(iv) Accelerating medical countermeasure advanced research and development

While the items listed are only defining a broad set of goals to achieve, the act further describes detailed activities (e.g., simulating drills and exercises to ensure medical surge capacity for events without notice) that enhance the abilities to respond to a crisis situation [44]. Therefore in the event of a natural disaster occurring both naturally or deliberately (e.g., hurricanes, disease outbreaks, the release of a biological agent), a feasible response plan becomes crucial to minimizing the severity of the impact it has on the public.

Emergency preparedness plans essentially provide a road-map to treat or serve the affected population within the time and resource constraints given. Usually preparedness plans involve a strategic placement of multiple points of dispensing (PODs) facilities that act as the central hub for their respective service area to distribute or dispense mitigation resources, such as life sustaining supplies or vaccinations [1]. The success of mitigation efforts that results from these plans are often dependent on utilizing sparse resources in the least cost manner. Therefore much attention is placed on each stage of distribution of resources, in particular distributing supplies to the PODs, followed by dispensing these supplies to the population.

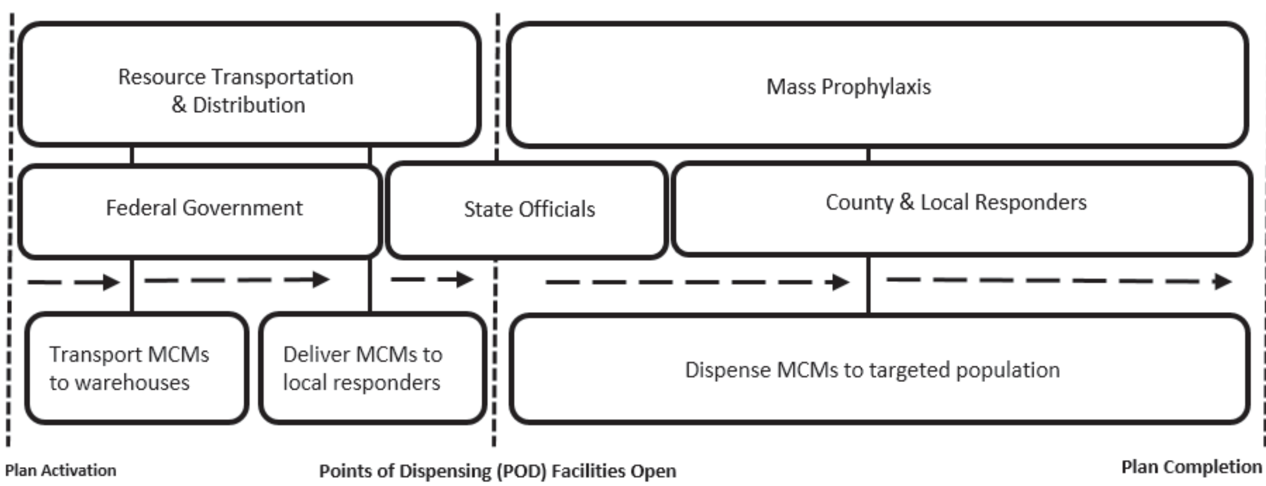


Figure 2.1. Phases of a bio-emergency response plan

In the event of a epidemic disaster, such as the release of the biological agent Antrax, public officials will rely on an emergency response plan that must result in the dispensing of antimicrobials (i.e., ciprofloxacin, doxycycline, amoxicillin, penicillin, etc...) to the population affected or at risk within 48 hours to treat/prevent illness from the threat. As shown in Figure 2.1, the commencement of this bio-emergency response plan will follow through two main sequential phases: (i) Resource Dispensing and Distribution, and (ii) Mass prophylaxis. The first phase the Federal government will transport the Medical Countermeasures (MCMs) from the Strategic National Stockpile (SNS) to the local state government's Receiving, Storing and Staging (RSS) warehouses. From the RSS warehouses, the MCMs must then be distributed to the affected region's strategically placed POD facilities within the affected region. This final stage is achieved through dispensing the MCMs to treat the population [1]. The MCMs described in these plans are not restricted to just pharmaceutical interventions (e.g., vaccines, antimicrobials). According to the Receiving, Distributing, and Dispensing Strategic National Stockpile Assets: A Guide to Preparedness, the Public Health Emergency Medical Countermeasures Enterprise (PHEMCE) defines MCMs as the resources that may be used to "prevent, mitigate, or treat adverse health effects from an intentional, accidental, or naturally occurring public health emergency" [1]. With MCMs also including non-pharmaceutical interventions (e.g., ventilators, personal protective equipment), these emergency response plans can cover a large variety crisis situations.

2.2. Literature Review

2.2.1. Overview of the Vehicle Routing Problem (VRP)

A wide span of applications in logistics and distribution management are modeled by variations of the general vehicle routing problem (VRP). The classical Vehicle Routing Problem (VRP) can generally be described as the development of optimal vehicle routes that start a single depot, visit a provided set of geographically located customers, and return to the depot in the least cost manner. These VRP variants then define the objective functions and constraints that represent this cost accordingly.

The VRP is a well known problem with an extensive body of research and litterature dating back to 1959 where it was first introduced by [23] who described it as the Truck Dispatching Problem. Initially the VRP was considered a generalization of the Traveling Salesman Problem (TSP) were additional conditions and constraints are imposed, such as multiple vehicles and capacity constraints. Both the TSP or the VRP are computationally difficult problems when they represent an instance that is non-trivial, which in actuality represents most applications of a realistic size. Through the generalization of the TSP it has been shown that the VPR cannot be solved in polynomial time because of its computational complexity and therefore is an NP-hard problem for these non-trivial cases [45]. Moreover, certain multi-objective optimization goals and constraints can restrict the a VRP solution from improving unless at the detriment of another objective. Therefore the solution to any specific instance and variant of the VRP that is determined to be the best solution will coincide with the Pareto optimality of the problem. This state of optimality then describes the solution to the VRP in which the objectives can no longer be improved from any reallocation of resources without introducing deterioration to at least one of the objectives [32].

One of the reasons the VRP is so well studied is due to its broad spanning application to logistics and distribution management. Modeling these real world logistical problems (e.g., United States Postal Service mailing routes) have further lead to many natural extensions of the VRP resulting in countless variations that define optimization goals adhering to the unique constraints representing the characteristics defining the application. This heterogeneous nature of VRP variants has led to the inability to globally define this problem in a singular accepted formulation. The most common of these varations include conditions restricting capacity and total time of any route, bounding the number of stops a route may include, and routing restrictions designating certain stops that must be either selected first as a priority or delivered within a given time window [42]. The surveys and literature reviews in [40][17][20][41][62] include further comprehensive lists of the variations of the VRP and their application in practice.

Several algorithmic approaches have been developed for solving both the connical VRP and its variants, yet they each introduce trade-offs such as performance, quality, and accuracy of the solution itself or its derivation. Moreover the usability of these approaches and their respective solutions can be influenced by factors such as the number of variables and constraints described by the multi-objective optimization formulation, computing infrastructure capabilities or limitations, and even the solution resilience to uncertain or stochastic variables. As a result the flexibility of a algorithm to adapt to different environments that define variations of the VRP has become increasingly important. In fact, the benefit of designing algorithms that model the canonical form of the VRP with the ability to adapt to more complex and practical conditions has been identified [43]. This instead would be an alternative to the extension of the existing variants of VRP.

Existing literature for solving the VRP are commonly classified into one of the two classifications: (i) exact optimization algorithms that solve for the best possible (i.e., optimal) solution [18][50] and (ii) approximation algorithms [40][54] that rely upon heuristics to reach an acceptable solution [27][17]. The acceptability or quality of a solution can be influenced by a number of factors such as the number of variables and constraints described by the multi-objective optimization formulation, computing infrastructure capabilities and limitations, and even the solution resilience to uncertain or stochastic variables. Within these two broad classifications, algorithm approaches can be further divided into different levels of additional sub-classifications that describe their basic strategy. Figure 2.2 provides a road map of algorithms by their classification of approach and strategies that often be easily identified in previous works. Each of these classifications are capable in solving all manners of the VRP and its variants with trade-offs such as performance, quality, and accuracy of the solution. Furthermore, algorithms are not restricted to only one of the classifications listed but instead can utilize different strategies of the overall approach. In fact, certain stages within a given strategy are often represented as smaller sub-problems that adhere to less computationally complex problems to solve, such as the TSP which can employ different sophisticated algorithms for solving it. For example, it is common for many algorithms classified within the approximation algorithms to incorporate post-optimization strategies for re-ordering customers to visit within individual routes [16]. The following sub-sections will provide an insight to the strategies and formulations that are core to these algorithm classifications.

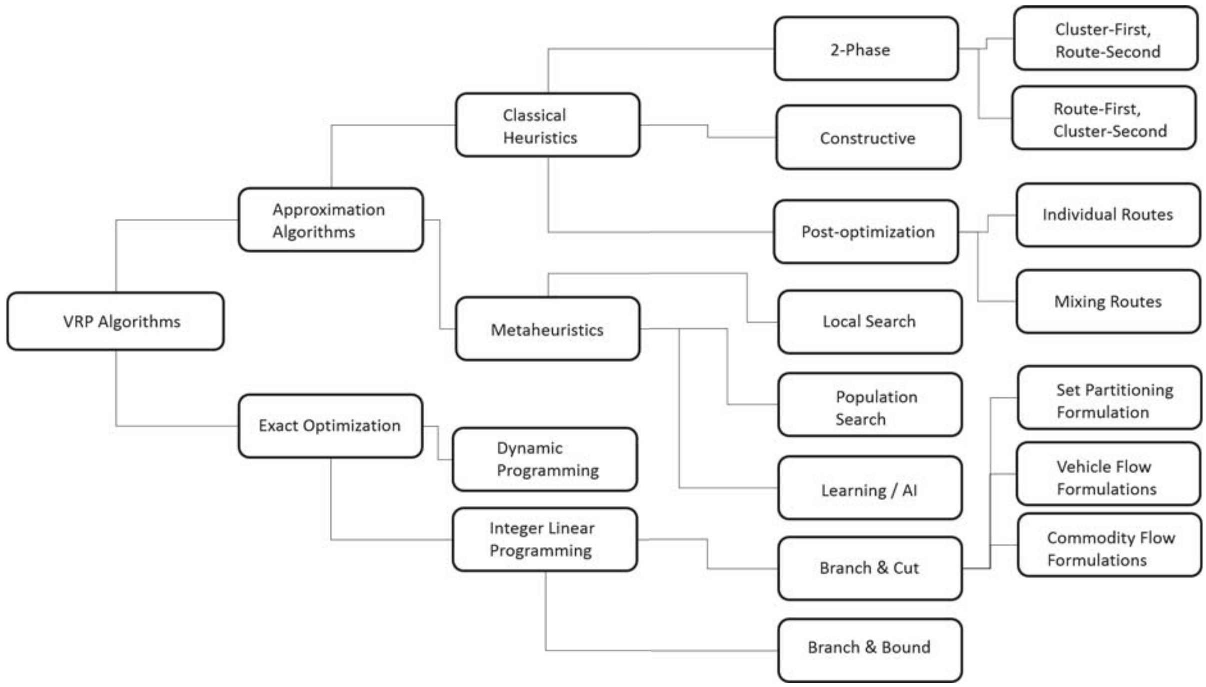


Figure 2.2. Algorithm classification for the Vehicle Routing Problem

Exact Optimization Algorithms

Previous research for solving the VRP with exact optimization algorithms have demonstrated numerous approaches such as Integer Linear Programming (ILP) and Dynamic Programming (DP), which are commonly utilized in solving a wide span of combinatorial optimization problems. Solving the VRP through DP involves recursively searching subsets of the customers to be delivered by a fixed number of vehicles to find the minimum cost of vehicle routes. This approach is formulated in [42] and is also noted by the authors that the excessive amount of computations performed by this DP algorithm require significant reductions in the search space through relaxation procedures of the constraints. These conditions have restricted the DP algorithms to only be viable in small instances of the problem, and as a result are not commonly used in practice.

The most successful strategies of exact optimization algorithms are based on solving ILP formulations through the use of two classes of algorithms: Branch & Bound (BnB), and Branch & Cut (BnC); both of which rely on combinatorial relaxations. Exact algorithms for solving the VRP through BnB have been based on various relaxations from elementary relaxations to recently more sophisticated bounds. [20] describes these relaxations of BnB algorithms based on the classic *Assignment Problem*, state-space relaxations, and degree-constrained shortest spanning tree, along with the more recently defined bounds relying on Lagrangian relaxations and additive bounding procedures. However, the applicability of BnB methods are similarly constrained by the DP algorithms. Conversely, solving the VRP using BnC algorithms are capable of solving much larger instances of the problem, although these instance are still very limited compared to approximation algorithm approaches. BnC algorithms are used to solve the VPR modeled by one of the following ILP formulations: Set Partitioning (SP), Vehicle Flow (VF), and Commodity Flow (CF) formulations. [47] describes the interrelationships between these ILP formulations. SP formulations were first introduced by [55] representing the VRP through a cluster-first, route second approach, but due to a large number of variables required these formulations are not practical in obtaining an optimal solutions for most cases. VF formulations are the most commonly used formulation as it is an extension of the TSP assignment based formulation. The VF formulation can further be categorized into two formulations (Three-Index and Two-Index VF formulations) as described in [50]. Similarly to the VF formulation, the CF formulation is also an extension to the TSP to a lesser degree. CF formulations combine constraints modeling the movement of goods (i.e., load or capacity) with the movement of vehicles (i.e., route) together. These formulations are described in detail in [47][50].

Existing litterature supports a signficant amount of algorithms that have been developed to solve these ILP formulations through BnC for modeling different variations of the VRP. [43] includes a survey of exact algorithms that specifically solve the Two-Index VF, Two-Index Two-Commodity Flow, and SP formulations of the VRP by using a BnC approach. Other BnC algorithms for these formulations are additionally surveyed in [50][42][20]. Beyond the BnC approach, [18] reviews exact algorithm approaches for solving the VRP that are based on spanning tree and shortest path relaxations, along with presenting computational bounds and results in comparison between algorithms. However, due to the complexity of the VRP, even the most sophisticated of these exact optimization algorithms will only have varying success where the instances of the VPR are limited. For example, [46] demonstrates a general BnC algorithm for the Capacitated Vehicle Routing Problem (CVRP) and [9] presents a comparable BnC algorithm based on the SP formulation with additional cuts using a Langruarian relaxation for the CVRP, yet in both cases the algorithms appear to be limited at instances with around 100 customers. These two algorithms vary in their performance with respect to solution quality, accuracy, and execution time of their individual problem instances and benchmarks. These limitations hold for even the more recent formulations and extended variations of the VRP, such as the Dial-A-Ride Problem as shown in [21].

Unlike using BnC algorithms for deriving optimal solutions of the TSP for significantly larger instances than that of the VRP [61], it can be clearly observed that practical instances of the VRP quickly surpass the limitations of these exact optimization algorithm approaches. Additionally, the complexity of implementation and the computational resource limitations and requirements further reduce the practicality of these approaches and motivate the use of approximation optimization algorithms instead.

Approximation Optimization Algorithms

Approximation optimization algorithms for the VRP are similar those used for most combinatorial optimization problems as they relay upon heruistics to reach an acceptable solution within a reasonable time. As motivated by the upper bounds of the exact algorithms mentioned previously due to the VRP being an NP-Hard problem, heuristic approaches to the VRP account for the majority of the research focus because of the solutions produced and flexibility to more easily adapt to variants of the VRP than exact algorithms are. Coinciding to that of exact optimization algorithms for the VRP, existing literature of approximation algorithms can be grouped into two comprehensive categories of strategies that define their approach or procedures: (i) Classical Heuristics and (ii) Metaheuristics. These algorithms are not mutually exclusive to these categories as many approaches fall into a number of these in the groups, even as pre- or post-processing steps. Additionally, approximation algorithms can be used to determine the upper bounds of the VRP for advancement or comparison of exact algorithms [20].

The continuation of new VRP variant introductions into this area of logistics research has resulted in a substantial amount of new algorithms whose procedures are usually an improvement upon an originally introduced algorithm as the foundation. For example, the Clarke and Write Savings Algorithm [19] developed in the early 1960s is a constructive heuristic approach to the VRP that has maintained its popularity due to its simplicity and quick execution. This algorithm consists of constructing an initial set of feasible routes, followed by an iterative process of merging routes when there is a benefit computed by a savings function. Using this strategy as the foundation, it has continued to be improved upon [3][53][35]. The Sweep Algorithm by Gillett et al. [28] is also an important algorithm for the VRP by solving the ILP Set Partitioning formulation through a rotation of a half line in intervals around the depot to gradually add customers onto the route. The 2-phase algorithm by Fisher et al. [25] is another novel approach by first solving the generalized assignment problem (GAP) to cluster the customers, followed by solving the TSP individually within each cluster.

Classical heuristic deficiencies such as returning solutions that represent local optimums from limiting the search space by ignoring solutions that don't immediately make an improvement, have resulted in a majority focus on metaheuristics. Applying metaheuristics to the VRP has expanded this search of the solution space and demonstrated significant improvements to VRP solutions for over 20 years [27]. [16] provides some history of this advancement and provides an overview of metaheursitic algorithms. General purpose metaheursitic algorithms have demonstrated some of the best results and can be associated with the schemes: (i) local search (e.g., simulated or deterministic annealing, tabu search), (ii) population search (e.g., genetic algorithms, adaptive memory procedures), and (iii) learning mechanisms (e.g., neural networks, ant colony optimization). Furthermore, many of the approaches will incorporate a number of these schemes and are even used as a hybrid approach with the classical heuristics and exact optimization algorithms. A review of metaheuristic algorithms and a survey of these respective existing approaches are discussed in [43][20][42].

Tabu search heuristics have typically demonstrated to be some of the most promising approaches. This heuristic has even posted some of the best known solutions to popular benchmarks. However Tabu search has a tendency to be slower in determining a solution. It also can require a significant amount of variables to be included, all of which usually must be tuned. Nevertheless, it could still potentially offer only slight improvements over more simplistic and efficient algorithms. Hence advantages can be gained from alternative heuristic algorithms. For example, [8] developed an extension to the generalized assignment heuristic [25] as an efficient and computationally fast way to obtain results within just a few percentage points of optimal VRP benchmarks without some of the complexities of Tabu search.

2.2.2. The VRP in Emergency Response

The unpredictable nature of a disaster situation requiring immediate emergency response, in addition to the disparity between scenarios, can quickly add complexity to the mitigation efforts. These are described by the varied spatial and temporal aspects of infrastructural components (e.g., road network, treatment facility capabilities) or limiting conditions of the scenario (e.g., resource availability, disaster longevity). As a result, logistical operations in such scenarios necessitate the incorporation of the high-consequence constraints when formulating the VRP. Furthermore, the evaluation of circumstantial information availability and ambiguity presents unique challenges from the classical VRP formulation when modeling this for emergency response. [60] presents a model for the facilitation of dynamic resource demand allocation and distribution under large-scale disasters. This approach includes the fusion of multiple data sources to forecast demand and then will prioritize affected areas using multi-criteria fuzzy clustering. Although this methodology is beneficial for optimizing the relief network, the original problem of optimally delivering the supplies defined by the VRP still exists.

Unlike the classical VRP, logistics in emergency response plans require the evaluation of when information is received and the ambiguity of the information. Following the evaluation of the information quality and evolution, emergency response planners must consider the strategies to respond and the capabilities required. Allahviranloo et al. [2] classify this type of response into two general frameworks for handling this information uncertainty developing VRP solutions either with or without recourse. Essentially, if the situation allows for solutions to adjust after the start of the response plan execution then formulating the problem as a the Dynamic VRP (DVRP) is an appropriate strategy for incorporating variable uncertainty. Variants of the Dynamic VRP (DVRP) provide adequate formulations in modeling logistical operations in emergency response under the assumption that iterative optimization can be made after the initiation of a response plan. Moreover, the incorporation of variable uncertainty from the emergency situation is a benefit of this formulation. [54] describes the DVRP formulation with the relation to the information quality and its evolution while further surveying applications and the corresponding applicable approaches. Specifically, the DVRP for relief logistics in natural disasters is presented in [17]. A holistic model formulating a nonlinear dynamic logistics model for disaster response under uncertainty is additionally presented in [38]. However in contrast with the aforementioned scenarios, many emergency disaster situations depend on optimal emergency response plans and do not rely on the re-optimization after the plan is initiated. In these situations, small perturbations to the plan can result in large consequences. For example, the effectiveness of mitigating a bio-emergency can be completely compromised unless the allowed time window for treating the affected population is met. Therefore the initial optimization must at least meet these constraints and cannot rely on further optimization for this after the fact. Further logistical and policy issues such as the security requirements for accompanying vehicles transferring resources often prevent the dynamic optimization solutions from being used in these type of situations and instead will rely on a solution without changes after the initiation. [2] presents three Selective VRP (SVRP) formulations for optimizing the selection of facilities to be used and delivered when information is uncertain and recourse is not allowed. A robust optimization algorithm for emergency vehicle scheduling is presented in [56] and includes the comparison against a particle swarm optimization algorithm from a purposed case study. Although this research is beneficial for modeling the uncertainty in these scenarios that can be reflected in the optimal network of facilities to be delivered to, these formulations are still reliant on existing heuristic algorithms such as the genetic algorithmic approaches presented by the authors. Additionally these formulations must be flexible to reflect the optimization priorities and high-consequence constraints of the situation it is modeling.

CHAPTER 3

MINIMIZING THE VEHICLES REQUIRED FOR THE OVRP-UT

3.1. OVRP-UT Mathematical Formulation

We generally describe the geometric OVRP-UT as the combinatorial optimization problem of determining the minimum number of vehicles required to visit a given set of customers located within a region. In an effort to maintain some consistency with existing research of related problems, a simple mathematical programming formulation of the OVRP-UT is provided based on the *two-index vehicle flow formulation* in [43].

We define this problem on the undirected graph $G = (V, E)$, where the vertex set $V = \{0, 1, ..., n\}$, and the edge set $E = \{[i, j] : i, j \in V, \forall i < j\}$. We denote the depot (i.e., the location where all vehicle routes must start) by the index $0 \in V$, and each $i \in V \setminus \{0\}$ is a customer required to be visited by exactly 1 route. This implies that G is constructed as a complete graph. Therefore any customer can directly reach the depot and all other customers for a cost defined by the edge of the adjacent vertex. Accordingly, no permutation of customers are restricted. The cost matrix, c_{ij}, is defined on the edge set E, and corresponds to the cost of travel between the customers i and j. If $i = 0$, c_{ij} represents the cost between the depot and customer j. We assume that the cost of travel is symmetric and these costs satisfy triangle inequality, where $c_{ij} = c_{ji}$ and $c_{ij} \leq c_{ik} + c_{kj}$, $\forall i, j, k \in V$ respectively.

Initially, we abstractly use the term *cost* in the formulation, because in practice cost is relative to the geometric region that defines the problem (travel time, length, etc...). For the remainder of this chapter unless specifically stated, these terms are used interchangeably to represent cost. It is additionally assumed that any units of measure used for the cost are all identical.

Next, the binary variables, x_{ij} and y_{ij}, are introduced to represent, respectively, the existence of the edge $[i, j]$ in the optimal solution and the assignment of vertex j to route i. Customers and depots are both denoted as a vertex accordingly. Then

$$x_{ij} = \begin{cases} 0 & \text{if the edge } [i,j] \text{ does not appear in the optimal solution} \\ 1 & \text{if the edge } [i,j] \text{ does appear in the optimal solution} \end{cases}$$

and

$$y_{ij} = \begin{cases} 0 & \text{if vertex } j \text{ is not assigned to route } i \\ 1 & \text{if vertex } j \text{ is assigned to route } i \end{cases}$$

The number of vehicles available, denoted as m, usually represents a fixed known value. Conversely in the OVRP-UT the value of m is unknown, instead representing the vehicles used to serve all n customers in $V \setminus \{0\}$. Moreover each route i for $i = 1, ..., m$, must have an associated total cost no greater than Z, where Z denotes a constant integer corresponding to the maximum allowed total travel time per route. Accordingly the value of Z is fixed, thus it is an unvarying constraint for all routes. We assume that the value of Z is known in the problem definition.

The OVRP-UT formulation follows:

$$Minimize\ m \quad = \quad \sum_{j=1}^{n} x_{0j} \tag{1}$$

subject to:

$$\sum_{j=1}^{n} \sum_{k=j+1}^{n} y_{ij} x_{jk} c_{jk} + \sum_{j=1}^{n} y_{ij} x_{0j} c_{0j} \leq Z, \quad i = 1, ..., m \tag{2}$$

$$x_{0j} c_{0j} \leq Z, \quad j = 1, ..., n \tag{3}$$

$$\sum_{i=1}^{m} y_{ij} = 1, \quad j = 1, ..., n \tag{4}$$

The primary objective function in this formulation is to minimize the total number of vehicles to serve all customers within the required time Z. Objective (1) represents this minimization using the total number edges $[0, j]$ between the depot and any customer in the optimal solution, due to the fact that all routes must start from the depot. Constraint (2) define the core constraint of this variant, restricting the total cost of any route from exceeding the maximum total cost allowed per route Z. The first and second summation in this constraint represent the cost of route i starting from the first customer visited through the last customer in the route. The third summation in the constraint represents the cost from the depot to the first customer of the route. Constraints (3) define the assumption that the cost from the depot to any customer is no more than Z. Constraint (4) ensures that all customers are visited by exactly 1 route.

For clarity, we define the conditions required for a solution to be considered *optimal* and/or *feasible* based on this formulation. For a feasible classification, the set of routes must satisfy all of the criteria defined in Constraints (2),(3) and (4). For a solution to be optimal, it must at least be feasible. At the point of feasibility, the number of required vehicles m, is similarly considered a feasible solution. Additionally when Objective (1) is satisfied, the solution is then considered optimal. For this purpose we denote m^* as the absolute lower bound of the number of vehicles required, representing the optimal solution to the problem.

Once m^* is determined, slight adjustments of the formulation can be made to generalized this problem as a classical OVRP, where m^* denotes the fixed number of vehicles available. Without loss of generality, Constraint (2) would therefore become the secondary optimization objective by instead representing it as a minimization objective. However, this would require defining a state of Pareto optimality to denote an optimal solution. In this case, the priority objective is associated with maintaining the value of m^* such that all routes have a total cost no more than Z. The secondary objective of minimizing the sum total cost of all routes can be improved accordingly, as long as the primary objective is not negatively impacted, i.e., the value of m^* does not increase. In other words, the original primary objective to the classical OVRP essentially can be applied to represent a post-processing step for the OVRP-UT once m^* is determined. Nonetheless our study focuses solely on solving the OVRP-UT as defined.

3.2. Structuring a Heuristic from an Optimal TSP

In this section we extend upon the OVRP-UT formulation by establishing the geometric context of the problem. With this in mind, the current notation is broadened to include new variables and terms to be used for the remainder of this chapter. Additionally the terms *locations*, *cities*, and *points* are used interchangeably. The term *cities* is particularly used to reference a set containing a depot and customers.

For the *geometric* context of the OVRP-UT, we assume that the vertex set V representing the depot and customers are re-defined as the complete set of cities, S, within the region X, where $S = \{s_0, s_1, ..., s_n\}$. We similarly let s_0 represent the depot location within the set of cities S. The total number of cities in a given set or region is denoted as $N()$, such as $N(X)$ for the region X. Hence $N(X) = n + 1$ since X is the containing region of all the cities in S (i.e., 1 depot, plus n customers). The cities in S exist on a Euclidean 2-dimensional plane such that each is associated with an (x, y)-coordinate in $\mathbb{R}^2$. Thus the edge set E and its corresponding cost matrix c_{ij}, represent the Euclidean distance between the points $i, j \in S$. For example in c_{ij}, where the points $i = (x_i, y_i)$ and $j = (x_j, y_j)$, the cost is simply calculated as

$$c_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \tag{5}$$

The partitioning algorithm described in Section 3.4, and the corresponding analysis presented in the following sections all rely on the quantification of the least cost individual TSP tours, which we denote generally as

$$T_S = \{s_{\phi(0)}, s_{\phi(1)}, ..., s_{\phi(n)}, s_{\phi(0)}\}. \tag{6}$$

In other words, T_S denotes the *optimal* TSP solution of the cities in S whose tour is represented by the least cost permutation of the cities, where each $s_{\phi(i)}$ is the city $s \in S$ whose subscript, $\phi(i)$, maps the ith position in the tour and maps to a specific city within the set. Unless explicitly stated we assume that each tour begins and ends at the depot (i.e., a *closed* tour) and is optimal for the provided set of cities. In the case of an *open* tour such as in the previously defined OVRP-UT, the superscript *'O'* is included to signify that the tour only starts the depot and excludes the return trip back to the depot. Thus T_S^O would represent the optimal open tour through the cities in S which only begins at the depot.

The cost of the associated tour constructed from the cities in S is denoted as $|T_S|$. If we let i and j reference the customers mapped at the i- and j-index in the permutation of customers for the tour T_S, then the total cost associated with traversing $N(T_S) = n + 1$ total customers is calculated as

$$|T_S| = c_{n0} + \sum_{i=0,j=1}^{i=n-1,j=n} c_{ij} \tag{7}$$

Furthermore, an open tour can simply be constructed from an closed tour by just removing the depot from either end of the open tour. Accordingly, the resulting open tour would maintain the same permutation of the closed tour it was constructed from, excluding the return to the depot. Or conversely if the depot is removed from the start of the tour, the open tour would represent the reversed permutation of the closed tour. Thus, the cost of the open tour T_S^O constructed from T_S is calculated as

$$|T_S^O| = |T_S| - max\{c_{n0}, c_{01}\} \tag{8}$$

In that case, constructing T_S^O from T_S will always result in $|T_S^O| < |T_S|$.

As previously mentioned, the heuristic presented in this chapter assumes that the optimal TSP solution for any given set of cities is either known or can be derived. This is admittedly not always achievable for instances where the number of cities is significantly high. Under those circumstances, sub-optimal TSP solutions can be used in place of the optimal TSP solution when this cannot be assumed. Therefore if the optimal cannot be assumed than the underlying error relative to the selected approximation TSP algorithm will just decrease the overall performance bounds presented this study accordingly. This concept of propagating the error to results based around optimal TSP tours is similarly presented in the conclusion of [37] for extending the work. To that end, [30] presents a theorem that states the length of a tour created using any insertion heuristic (i.e., inserting all points in a set one by one in some defined order) gives a length at most $\lceil \log_2(n) \rceil + 1$ times the the optimal TSP. Although the optimal TSP continues to be assumed for the heuristic algorithm presented in Section 3.4, extending these results to exclude this assumption is relatively straightforward. For example, [13] describes many fast algorithms for approximating the geometric TSP and includes a comparison of their performance relative to various distributions of the cities. The euclidean distance as shown Equation 5 is particularly useful for geometric TSP approximations. Approximation algorithms for the geometric TSP are often highly efficient even for a large number of cities. Accordingly, [6] presents multiple polynomial time approximation schemes for the euclidean TSP.

3.3. Fundamentals of the Heuristic Scheme

Determining the optimal value, m^*, often requires specific knowledge of the problem instance's defining properties (number of cities and their distribution, cost constraints, etc...). Nonetheless, a loose lower bound, denoted at $\omega(m)$, is easily obtained without this information for $\omega(m) \geq m^*$. The lower bound $\omega(m)$ of vehicles required in the OVRP-UT can intuitively be constructed from a generalization of determining the number of vehicles required in the Capacitated-VRP (CVRP) literature. For example, [43] determines this value for the CVRP by dividing the total sum of all the customers' demand, divided by the capacity of a vehicle. Since the solution to the OVRP-UT is determined by the set of individual open traveling salesman tours (OTSP), a similar bound can be constructed as

$$\omega(m) = \frac{max(c_{ij}) * n}{Z - max(c_{0i})} \quad i > 0, i \neq j, \quad \forall i, j \in S, \tag{9}$$

where $max(c_{ij})$ and $max(c_{0i})$ denote, respectively, the max distance between any two customers and the max distance between the depot and any customer. However equation (9) is clearly a very loose lower bound of m^*, as it is assuming individual OTSP routes at a very high cost. It is well-known that the individual optimal TSP tour grows at a cost of $\sqrt{n}$

when n is large enough, where $n = N(S \setminus \{s_0\})$[14]. Thus we conjecture that a tighter lower bound, denoted as $\Omega(m)$, of required vehicles is possible when partitioning an optimal- or near optimal-TSP for the OVRP-UT.

Axiom 3.1. *For all positive integer values of m denoting the number of vehicles tours in any independent solution, $m^* \leq m$ when a solution of m tours satisfies the OVRP-UT criteria.*

From Axiom 3.1, it is evident that demonstrating the existence of a set of m vehicle routes satisfying all constraints of the OVRP-UT formulation only proves that the value of m^* is at most equal to m. Even more so, no measure of *closeness* relative to $(m^* - m)$ is discernible and subsequently prevents an evaluation of m in comparison to the absolute minimum value. Conversely if the same instance of m total vehicle routes can additionally show that that no feasible solution exists with $(m - 1)$ vehicles tours then m^* must equal to m.

Nonetheless the challenges are quite apparent in proving the feasibility of specific values of m. This is particularly the case for the smaller values of m relative to the specific problem instance. Even more, any specific set of m constraint-satisfying vehicle routes can potentially have no resemblance or correlation to the set of $(m - 1)$ feasible routes when the inequality $m^* \leq m$ is valid (or assumed). On the other hand, explicitly constructing a set of m feasible routes proves without any uncertainty that m^* is at most equal to m. For these reasons an alternative approach is defined.

Instead of selecting an arbitrary value of m that is feasible and then iteratively proving this feasibility for values less than m, the feasibility of m is first verified through a single tour through all cities such that $(m = 1)$. The value of m is then incrementally increased until it is deemed feasible and thus represents the value of $\Omega(m)$ relative to the optimal value m^*. Determining $\Omega(m)$ through a partitioning approach is motivated by the aim of structuring a heuristic whose underlying notion extends from the concept of Axiom 3.1.

3.4. A Recursive Partitioning Heuristic

In this section we present a recursive partitioning heuristic to construct the $\Omega(m)$ bound that satisfies the constraints defined by the OVRP-UT previously defined in Section 3.1. Algorithm 1 formally describes this heuristic and consists of two fundamental procedures: (i) splitting a single tour into two open TSP tours at a similar cost, and (ii) constructing a third tour from a set of cities removed from two infeasible tours. The objective of the first procedure is to try and construct two feasible tours satisfying all constraints. Similarly, the second procedure's objective attempts to modify these two tours by introducing a third tour in a manner such that they are all feasible. These procedures are defined in the following sections accordingly by the SPLIT and the EXTRACTCITIES procedures, which reflect the aforementioned strategy for deriving $\Omega(m)$ by incrementally increasing the value of m, initially set as 1, until a feasible value is reached. Specifically, Algorithm 1 extends this strategy by recursively partitioning a set and independently verifying its feasibility in three distinct *phases* at each recursive step. Table 3.1 illustrates the objective of these phases for the initial execution of the algorithm. If $\Omega(m) > 3$ is verified in the third phase, the algorithm is recursively called independently for subsets of the original set S. For this reason, Algorithm 1 denotes the parameter T_A in a general manner. Accordingly, A denotes either the initial set S in the first pass, but otherwise denotes $A \subset S$, as a subset of S.

Given a set of cities, A, Algorithm 1 constructs a lower bound on the total number of vehicles required for the OVRP-UT constraints to be satisfied. As illustrated in the initial call of Algorithm 1, the optimal TSP tour, T_A, will represent the complete set of cities defined by the problem instance. For the first phase, the feasibility of a single tour (i.e., $m = 1$) is examined. If ($m = 1$) is shown to not be feasible, the second phase attempts to construct a solution to examine the feasibility of two tours (i.e., $m = 2$). However, if ($m = 2$) is similarly demonstrated to not be a feasible solution for the set of cities in A, three tours are constructed in the third phase. For most problem instances it is to be expected that $m^* > 3$. Assuming $m^* > 3$ for A, the value of the lower bound $\Omega(m)$ is at least verified following the

Bound Verified	Fleet Size	Algorithm Phase Description
$\Omega(m) \geq 1$	$m = 1$	**[Phase 1]** Determine the optimal tour, T_S, through all cities in S, starting and ending at the depot. If $\lvert T_S^O \rvert \leq Z$, then $\Omega(m) = 1$. Otherwise we have illustrated that $\Omega(m)$ must be greater than 1 and continue to Phase 2.
$\Omega(m) \geq 2$	$m = 2$	**[Phase 2]** Partitioning the cities from the set $\{S \setminus \{s_0\}\}$ into two independent sets ,A and B, such that an optimal open tour of these sets, T_A^O and T_B^O, are approximately equal, $\lvert T_A^O \rvert \cong \lvert T_B^O \rvert$, after adding the depot back in for each. If $max\{\lvert T_A^O \rvert, \lvert T_B^O \rvert\} \leq Z$, then $\Omega(m) = 2$. Otherwise we have illustrated that $\Omega(m)$ must be greater than 2 and continue to Phase 3.
$\Omega(m) \geq 3$	$m = 3$	**[Phase 3]** Extract an optimal set of cities $Y \subseteq A \bigcup B$ from the two routes, T_A^O and T_B^O, currently exceeding the cost of Z such that $A_Y \leftarrow A \setminus Y$, and $B_Y \leftarrow B \setminus Y$. Determine the optimal tours independently for each of the resulting sets as $T_{A_Y}^O$, $T_{B_Y}^O$ and T_Y^O. If none of the three routes have a cost that exceeds Z, then $\Omega(m) = 3$. Otherwise we have illustrated that $\Omega(m)$ must be greater than 3 and we will repeat Phase 1 for each set, A and B, independently.

Table 3.1: Description of the three phases defining the procedures applied in each recursive step of the partitioning heuristic. This table provides a general outline of objectives in Algorithm 1or the initial execution. The first column represents the lower bound that is verified at the beginning of the phase. The second column denotes the fleet size to illustrate the feasibility of in this phase.

third phase. With this in mind, disjoint subsets of A are then independently passed into this algorithm recursively. Hence, each recursive step of this algorithm will consist of verifying the feasibility of a 1, 2 or 3 tour solution for the specific set passed in.

Table 3.1 generally describes the objective of the objective of the Split and ExtractCities procedures in Algorithm 1 for the second and third phases, respectively. This abstractly illustrates the structure of the the heuristic for verifying the feasibility associated with incremental values of m. The individual optimization statements, however, must each be addressed. The aforementioned fundamental procedures of the algorithm are presented to derive an optimal partition of an existing tour, as well as extracting an optimal subset cities to remove from a tour. As previously stated, it is assumed that an optimal tour adhering to the classical TSP can be achieved for any given set of cities.

3.4.1. Verifying the Correctness of the Partitioning Heuristic

Before explicitly defining the two additional procedures called in Algorithm 1, the *correctness* of this algorithm is verified in this section. Namely, the veracity of Algorithm 1 in representing the bounding procedure for $\Omega(m)$ is formally proven. In other words, we illustrate that the number of required vehicles derived from this lower bounding procedure

Algorithm 1 Recursive Partitioning Heuristic

procedure PARTITION(T_A, Z)
    if $||T_A| \leq Z$ **then** ▷ Phase 1
        return 1
    end if
    $\{T_B, T_C\} \leftarrow$ SPLIT(T_A) ▷ Phase 2
    if $|T_B| \leq Z$ **and** $|T_C| \leq Z$ **then**
        return 2
    end if
    if $|T_B| > Z$ **and** $|T_C| > Z$ **then**
        $\{T_{Y_B}, T_{Y_C}, T_Y\} \leftarrow$ EXTRACTCITIES (T_B, T_C) ▷ Phase 3
        if $|T_{Y_B}| \leq Z$ **and** $|T_{Y_C}| \leq Z$ **and** $|T_Y| \leq Z$ **then**
            return 3
        else
            return (PARTITION(T_B, Z)+PARTITION(T_C, Z))
        end if
    end if
    if $|T_B| > Z$ **then**
        return (PARTITION(T_B, Z) + 1)
    else
        return (PARTITION(T_C, Z) + 1)
    end if
end procedure

is in fact achievable (i.e., satisfies the OVRP-UT criteria) for the problem instance provided. This, however, does not explicitly illustrate the worst-case performance ratio of the resulting $\Omega(m)$ bound that is associated with the optimal lower bound m^*. Nonetheless, it does confirm that the structure of this heuristic is aligned with deriving the minimum number of vehicles required for any problem instance of the OVRP-UT.

For the sake of clarity in this section, the optimal solution, m^*, referencing the minimum required vehicles for a specific set of cities is denoted with a subscript of the set. For example, $m^*_{A_1}$ and $m^*_{A_2}$ denote respectively, the optimal solution for the set A_1 and A_2 independently. Moreover, unless explicitly stated, any set referenced in this manner is assumed to contain exactly one depot and at least 1 customer. From these assumptions, the following lemmas are presented for the verifying the correctness of Algorithm 1's partitioning scheme.

LEMMA 3.2. *$m^*_{A_1} \leq m^*_S$ for the subset $A_1 \subseteq S$.*

PROOF. Let $N(S) = n$ and $N(A_1) = l$. Since $A_1 \subseteq S$, $l \leq n$ accordingly. If $l = n$, then clearly $m^*_{A_1} = m^*_S$. Otherwise $m^*_{A_1} < m^*_S$ for when $l < n$. $\square$

LEMMA 3.3. *$m^*_S \leq m^*_{A_1} + m^*_{A_2}$ for the subsets A_1 and A_2 where $A_1 \subseteq S$, $A_2 \subseteq S$, $S = A_1 \bigcup A_2$, and A_1 intersects with A_2 by the depot only.*

PROOF. Given that $N(A_1) + N(A_2) - 1 = N(S)$ as a consequence of the subsets' properties, then for m^*_S in the worst case, the customers in A_1 will require $m^*_{A_1}$ tours, while the customers in A_2 will require $m^*_{A_2}$ tours. In other words $m^*_{A_1} + m^*_{A_2}$ are required for the set S in the worst case. Thus, $m^*_S \leq m^*_{A_1} + m^*_{A_2}$. $\square$

LEMMA 3.4. *$m^*_S \leq m^*_{\bar{S}} + m^*_{A_1}$, where $A_1 \subseteq S$, and $\bar{S} = S \setminus A_1$*

PROOF. Lemma 3.3 $\square$

Lemma 3.4 is an extension of the property demonstrated in Lemma 3.3. Accordingly, Lemma 3.4 illustrates that regardless of the customers included in the subset, $A_1 \subseteq S$, the minimum number of tours required for S is never more than $m^*_{A_1} + m^*_{\bar{S}}$, where $\bar{S}$ always denotes the customers not included in A_1.

LEMMA 3.5. *$m^*_S \leq m^*_{A_1} + m^*_{A_2} + ... + m^*_{A_n}$, where $A_1, A_2, ..., A_n$ are all disjoint (excluding the depot) such that $A_i \subset A, \quad i = \{1, 2, ..., n\}$*

PROOF. Lemma 3.4 $\square$

To illustrate the correctness of Algorithm 1, we first assume that set of cities, A, are partitioned into n subsets, denoted as $A_1, A_2, ..., A_n$, such that $A_1 \bigcap A_2 \bigcap ... \bigcap A_n = \{\text{depot}\}$, and $A_1 \bigcup A_2 \bigcup ... \bigcup A_n = S$. Next assume a feasible solution for each subset is derived. Applying Lemma 3.5 accordingly, the minimum number of vehicle tours required for X is guaranteed to be at most equal to the sum of tours used in each of the n subsets. Hence, any partitioning scheme adhering to this structure, including Algorithm 1, will result in a total number of vehicles that satisfying the OVRP-UT constraints. For this reason, both the SPLIT and EXTRACTCITIES procedures included in this algorithm are designed such that their resulting sets only intersect with the depot and their union will equal the union of the input sets. Furthermore, since each recursive call of this algorithm will always just result in further partitioning of the original set, these constraints will continue to be satisfied. For example, if the original set X is partitioned into the disjoint subsets B and C, a recursive call of the algorithm on B will never construct a subset that intersects with C beyond the depot. And similarly for C instead of B, accordingly. Therefore the $\Omega(m)$ bound derived by Algorithm 1 for a given problem instance will be achievable, and thus verifying that $m^* \leq \omega(m)$.

3.4.2. SPLIT: Verifying $\Omega(m) > 2$ for the Set A

In the initial execution of Algorithm 1, parameters of the PARTITION procedure include the optimal tour T_S, where S defines the set of all cities contained in the original region of interest. Additionally it takes the value of m initially set as 1 (corresponding to the initial route T_S), and the constant value Z defined by the problem. Proving $m^* > 1$ is easily demonstrated by validating $|T_S^O| > Z$, i.e., if the cost of the tour exceeds the allowed time for the entire region. This is expected to be true for large instances in practice for this initial pass. At some depth in the recursion however, the set of cities passed into this algorithm, denoted as A, phase will be small enough such that the construction of either 1, 2, or 3 independent routes through all cities in A will not exceed Z.

At the beginning of the second phase during the algorithm's initial execution, $m^* > 1$ will have been verified from PHASE:1. Thus the feasibility of $(m = 2)$ needs to be determined. Without loss of generality, it can be assumed that A..either denotes the complete set of all cities, S, or a subset of S associated with a recursive call of the algorithm. Regardless, the feasibility of $(m = 2)$ would still need to be determined for the set. For this reason, we reference A, as the set of cities to verify the feasibility of $m = 1, 2$, or 3 without explicitly mentioning the level of recursion it is associated with.

As previously stated, it is assumed that the optimal tour for any set of customers can be determined. Nonetheless, determining a single partition for A that assigns each of its customers to one of the two newly constructed tours must still be addressed. Moreover, this assignment must be performed such that the two resulting routes allow us to prove that

$m = 2$ is not feasible for A if we cannot derive an assignment to show otherwise. Under these circumstances two tours are constructed from a procedure based on logic introduced in Conjecture 3.6.

CONJECTURE 3.6. *Let* T_B^O *and* T_C^O *be any two open tours such that* $B \subset A$, $C \subset A$, $B \bigcup C = A$, *and* $B \bigcap C = n_0$; *where* n_0 *denotes the depot in the set of cities* A. *Additionally, let* T_A *denote the optimal TSP for the entire set* A. *Assume* l *and* k *are positive integers denoting the value of* $N(B)$ *and* $N(C)$, *respectively, where* $l > 2$ *and* $k > 2$. *Furthermore,* $l + k - 1 = n$ *where* $n = N(A)$ *since every set contains the depot. Accordingly, let* P_B *and* P_C *denote the permutations* $\{b_{\phi(0)}, b_{\phi(1)}, ..., b_{\phi(l)}\}$ *and* $\{c_{\phi(0)}, c_{\phi(1)}, ..., c_{\phi(k)}\}$ *for the tours* T_B^O *and* T_C^O, *respectively. Assuming* $\overleftarrow{P_C}$ *represents the reverse order of the permutation* P_C, *let* $T_{B\overleftarrow{C}} = \{P_B, \overleftarrow{P_C}\}$ *represent the resulting tour from appending the reverse order of the tour* T_C^O *to the end of* T_B^O. *Thus* $T_{B\overleftarrow{C}} = \{b_{\phi(0)}, b_{\phi(1)}, ..., c_{\phi(l)}, c_{\phi(k)}, c_{\phi(k-1)}, ..., c_{\phi(0)}\}$, *where* $b_{\phi(0)} = c_{\phi(0)} = n_0$. *Then,* $m^* > 2$ *for the set* A *if either* $T_B^O > Z$ *or* $T_C^O > Z$ *when the* $LCS(T_{B\overleftarrow{C}},\ T_A) \approx n$, *where LCS represents the value of the Longest Common Subsequence between the permuation of the tours.*[1]

Although we cannot prove or disprove this conjecture, the following supporting arguments are presented on why it is natural to assume to assume its validity. Consider a problem where a set of n total customers and single depot are given. From these cities a single optimal open tour, denoted as T_1, must be constructed by selecting a subset of these cities. The maximum number of cities that can be chosen to include in T_1 must be at least one and less than n. Following the construction of this route, the remaining cities that were not included in T_1 are then automatically assigned to a second open tour T_2. It is also assumed that T_2 will be a least cost tour corresponding to an optimal permutation of the cities it includes. Both of these tours will begin at the same depot. The objective of this problem is to minimize the total penalty. The penalty is determined by a function that grows linearly to the value of $|T_1| + |T_2|$. However, there is an additional penalty that is calculated by a function that grows exponentially with the absolute value of $|T_1| - |T_2|$.

[1]Information for the LCS is presented in [52]

From these constraints a decision heuristic can then be designed for selecting the cities to assign to T_1. Because the total penalty must be minimized, each decision will have a distinct penalty and an ambiguous penalty related to the choice. The known penalty for selecting a city is proportional to the tour cost of inserting the city into T_1. By selecting this city however, it is no longer available to include in T_2 at the end. This motivates a heuristic that iteratively selects the city that would (i) minimally increases T_1's total cost, that would otherwise (ii) maximally increase T_2's total cost if not assigned to T_1. Furthermore to reduce the difference in cost between the final tours, the heuristic would alternate the assignment of a cities to the tour with the lowest current tour cost. Assuming this decision heuristic will only make optimal decisions towards minimizing the total penalty previously described, we conjecture that the combined permutation of both resulting tours will closely reflect the optimal permutation of a single *closed* tour through all of the cities. Additionally the final cost of T_1 and T_2 should be minimized such that removing any city from one route to include in another would result in a cost increase, along with further unbalancing the routes' cost.

From the concepts of Conjecture 3.6, the SPLIT procedure defined in Procedure 2 is designed to construct two feasible tours from the optimal tour whose current cost is exceeding Z. As shown in Algorithm 1 (line 5), T_A is the parameter of the SPLIT procedure. This procedure starts at the beginning of T_A and traverses the tour while maintaining the cumulative cost at each additional step. During the traversal the index order of the current customer being visited is kept. When the traversal reaches a point in the tour at some index j, such that traveling to the next customer at index $j+1$ would result in a cumulative cost exceeding $|T_A|/2$, the traversal terminates.[2] This value of j that marked the index of the last customer traversed in T_S, becomes the splitting point to construct two new routes T_B^O and T_C^O. T_B^O is assigned the customers from T_A starting at the depot and through customer j, maintaining the same permutation. The assignment of T_C^O is similarly constructed with the remaining customers from index $j+1$ through $n-1$ in T_A. However, because T_A starts and ends at the depot, the permutation of T_C^O instead will begin at the depot, but then travels

[2]The indexes reference the customers corresponding to their position in the T_A tour.

to the customer at the $n-1$ in T_A. Accordingly, the tour for T_C^O will start at the depot, followed by the $n-1^{th}$ customer, then $n-2$, and so on until ending at the customer at the $j+1$ index in T_A.

Procedure 2 Splitting A Single Route

---

procedure SPLIT(T_A)
  $n = N(A)$
  $i = 0, j = 0$
  repeat
    $i = i + c_{a_{\phi(j))}a_{\phi(j+1)}}$
    $j = j + 1$
  until $(i + c_{a_{\phi(j))}a_{\phi(j+1)}}) > \frac{|T_A|}{2}$
  $T_B \leftarrow \{a_{\phi(0)}, a_{\phi(1)}, a_{\phi(2)}, ..., a_{\phi(j)}\}$
  $T_C \leftarrow \{a_{\phi(0)}, a_{\phi(n)}, a_{\phi(n-1)}, ..., a_{\phi(j+1)}\}$
  return $\{T_B, T_C\}$
end procedure

The final step of this phase is to verify $\Omega(m) > 2$ for the set A. This is again easily demonstrated by verifying if either $|T_B^O| > Z$ or $|T_C^O| > Z$. If even one of the two tours exceed the allowed time than we know that the addition of at minimum one more tour is required for the set of tours to be feasible. For most instances if $|T_S| > 2Z$, then both T_B^O and T_C^O will also be expected to exceed Z. This expectation is a consequence of the partitioning from the SPLIT procedure that produces two tours both at a cost approximately equal to $|T_A|/2$. Specifically Lemma 3.7 proves both $|T_B^O| \leq |T_A|/2$ and $|T_C^O| \leq |T_A|/2$, demonstrating that both tours will always have a cost less than half of the original.

LEMMA 3.7. *For an optimal closed tour* $T_A = \{a_{\phi(0)}, a_{\phi(1)}, ..., a_{\phi(n)}, a_{\phi(0)}\}$*, there exists an index,* i*, such that the sub-tours,* $\{a_{\phi(0)}, a_{\phi(1)}, ..., a_{\phi(i-1)}\}$ *and* $\{a_{\phi(i)}, a_{\phi(i+1)}, ..., a_{\phi(0)}\}$*, each have a total cost that is at most* $|T_A|/2$*. Additionally if one sub-tour equals* $|T_A|/2$*, the other must be less than* $|T_A|/2$*.*

PROOF. Let T_A be an optimal closed tour visiting $n+1$ the customers[3] of A. Since the tour T_A is closed, the depot will be visited twice (at the start & end of the tour) and the remaining n customers will be visited only once. Now let 0 index any customer in the tour

[3]Triangle inequality is still assumed for the associated cost between the customers

such that 1 will index the customer next in the tour according to the permutation of T_A, then similarly for 2 and so forth. Following the ordering of the permutation of the tour, beginning at customer 0 and sequentially moving through the remaining customers will represent the tours total cost $|T_A| = h$, where h is a positive constant value representing the total tour cost. Observe the sub-tour of T_A that starts at customer 0 and ends at customer i, denoted as T_{0i}. Moreover assume that the customer at index i represents the first customer in T_A such that the sub-tour $|T_{0i}|$ exceeds $h/2$. Therefore because $|T_{0i}| > h/2$ then $|T_{i0}| < h/2$ must also be true. Additionally for $j = i-1$, $|T_{0j}| \leq h/2$ must be true since as a consequence of $|T_{0i}| > h/2$. From these cases it is clear that $|T_{0j}| \leq h$ and $|T_{i0}| < h$ must both be true; hence proving Lemma 3.7 $\square$

Additionally by extending Lemma 3.7 we also know that the cost of the two resulting tours are approximately equal, $|T_B^O| \approxeq |T_C^O|$. Lemma 3.8 verifies this approximation by bounding the proximity to equality between these two tours' associated cost.

LEMMA 3.8. $\left|\left(|T_B^O| - |T_C^O|\right)\right| < c_{ij}$ *, where c_{ij} denotes the cost of the edge removed to split an optimal tour T_A between its i^{th} and j^{th} customers such both tours' associated cost do not exceed $|T_A|/2$ (as shown in Lemma 3.7).*

PROOF. To bound this approximation we first assume $T_B^O \leq T_C^O$ (i.e., the cost of the tour listed first is at most equal to the second tour's cost), otherwise the terms are simply switched and the following equations still hold. In a manner similar to the proof of Lemma 3.7, we continue to let h represent the cost of the original route to be partitioned, $|T_A| = h$. Accordingly it is then known that $|T_B^O| + |T_C^O| + c_{ij} = h$, and with a simple substitution the difference can be alternatively observed as

$$\begin{aligned}
|T_B^O| - |T_C^O| &= |T_B^O| - (h - |T_B^O| - c_{ij}) \\
&= |T_B^O| - h + |T_B^O| + c_{ij} \qquad (10)\\
&= 2|T_B^O| - h + c_{ij}
\end{aligned}$$

To demonstrate the bound of the equality approximation we recall that Lemma 3.7 proves that the addition of c_{ij} to either of these tours would result in a cost that exceeds $h/2$. Therefore both of these tours' cost must be within c_{ij} of $h/2$, and hence $|T_B^O| > h/2 - c_{ij}$ and $|T_C^O| \geq h/2 - c_{ij}$. Furthermore, Lemma 3.7 proves that $|T_B^O| < h/2$ and $|T_C^O| \leq h/2$. Thus,

$$\begin{aligned} \big||T_B^O| - |T_C^O|\big| &= 2|T_B^O| - h + c_{ij} \\ &< 2(h/2) - h + c_{ij} \\ &< c_{ij} \end{aligned} \tag{11}$$

□

Once the cost of the tours T_B^O and T_C^O are shown to exceed Z, we attempt to prove the feasibility of $(m = 3)$ in the following section. However, if $\Omega(m) > 3$ is verified to be true (as shown in Section 3.4.3), the routes derived from this section, T_B^O and T_C^O resulting from the SPLIT procedure will be used as the parameters for the next depth of the recursive algorithm.

3.4.3. EXTRACTCITIES: Verifying $\Omega(m) > 3$ for the Set A

Following the initial pass through the first two phases, the third phase (PHASE:3) begins with verification that that $\Omega(m) > 2$ for a set of optimal tours visiting the cities in A. To demonstrate that $(m = 3)$ is feasible, three tours must be constructed such that their associated cost does not exceed Z. Otherwise more vehicles are required for a feasible solution and thus $\Omega(m) > 3$.

The tours constructed from PHASE:2 provide significant value as a starting point for this phase. Recall that the objective in the OVRP-UT is to minimize the vehicles required. The previously constructed tours, T_B^O and T_C^O, similarly reflect this minimization objective in their assignment of cities between the tours. Accordingly, the $\Omega(m) > 2$ lower bound is validated specifically from these routes. Assuming $\Omega(m) > 2$, the cost of T_B^O and T_C^O can be used to quantify the distance between these two routes and a feasible solution with $m = 3$ routes.

Procedure 3 *abstractly* outlines the objective of defining three tours by: (i) removing cities from the two existing infeasible tours, and (ii) constructing a new tour from the cities removed. This resembles a generalized version of a *local optimization* TSP heuristic known as the *inter-tour improvement* [13], which improves an existing tour by moving a customer's current placement within the tour into different spots that result in a lower cost. However, in the ExtractCities procedure, cities are swapped from one of the two tours and into the newly constructed tour. Moreover, the objective is to improve the overall solution by *reducing* the two existing tours while *minimizing* the resulting cost of the tour constructed.

Procedure 3 Determining An Optimal Subset Of Two Tours For Removal

procedure ExtractCities(T_B, T_C)
  Determine optimal Y, where $Y \subset B \bigcup C$
  $B_Y \leftarrow \{B \setminus Y\}$
  $C_Y \leftarrow \{C \setminus Y\}$
  return $\{T_{B_Y}, T_{C_Y}, T_Y\}$
end procedure

Called recursively in Algorithm 1 (line 10), this procedure (ExtractCities) takes as input the two tours constructed from Phase:2 whose cost exceeded Z for a set of cities. Returning from this procedure are three new tours used to determine the feasibility of $(m = 3)$. The aim is to derive an optimal set Y, where $Y \subset \{B \bigcup C\}$, such that by removing the cities in Y from B and C, the resulting three sets will each have an optimal tour at a cost lower than Z.

Formally we denote the cities removed from B and C as $Y_B = \{B \bigcup Y\}$ and $Y_C = \{C \bigcup Y\}$ respectively, and thus $Y = \{Y_B \bigcup Y_C\}$. Additionally, the optimal tour through the resulting sets $\{B \setminus Y_B\}$ and $\{C \setminus Y_C\}$ (i.e., the cities remaining in B and C) are referenced, respectively, as $T^O_{B_Y}$ and $T^O_{C_Y}$, where $B_Y = \{B \setminus Y_B\}$, and $C_Y = \{BC \setminus Y_C\}$. Similarly the tour T^O_Y will denote the tour defined by the cities $\{Y_B \bigcup Y_C\}$ removed from the tours T^O_B and T^O_C. Due to the restriction that all tours must begin at the depot, it is assumed that the depot is excluded from both subsets of cities removed and is initially placed within Y.

When selecting the cities Y_B and Y_C to extract from T^O_B and T^O_C respectively, the resulting tour through the remaining cities must have a reduced cost such that $|T^O_{B_Y}| \leq Z$ and $|T^O_{C_Y}| \leq Z$. Accordingly, let $\triangle T_{i_Y}$ for $i \in \{B, C\}$ represent this reduction in cost from the tour T_i^O following the removal of the cities Y_i where

$$\triangle T_i = |T_i^O| - |T_{i_Y}^O|, \quad \forall i \in \{B, C\}. \tag{12}$$

Calculating this change in cost from Y_i when removing a city is relatively straightforward. This is calculated by: (i) subtracting the cost of the two edges between the city to remove and its adjacent cities in the tour, and (ii) add the edge cost to connect those same two adjacent cities.

We recall that T_B^O and T_C^O are the tours constructed from partitioning the optimal TSP tour T_A (i.e., a parameter of Algorithm 1). Hence from Lemma 3.7 and Lemma 3.8, $|T_A|$ can be used to illustrate the minimum value of $\triangle T_i$ required for $T_{i_Y}^O \leq Z$, $\forall i \in \{B, C\}$. Accordingly, the cost of T_S that exceeds Z is quantified as $|T_S| = \alpha Z$. Hence, the coefficient α is equal to $|T_S|/Z$, representing the number of times $|T_S|$ exceeds Z. Since it has been verified from the second phase that $m > 2$ for the set A, a division by 0 is not possible.

Lemma 3.9. *When $\alpha > 1$, $(m = 3)$ will not feasible if*

$$\sum_{i=A,B} \triangle T_i \; < \; Z(\alpha - 1) - [|T_A| - (|T_B^O| + |T_C^O|)]. \tag{13}$$

Proof. The cost of at least one tour will always exceed Z when this inequality (i.e., Equation 13) is true; hence proving this lemma. □

If the inequality in Equation 13 is not true, however, it does not necessarily mean that $(m = 3)$ is feasible for the set A. Instead it is only one of the conditions required to verify this feasibility accordingly.

With the intention to verify the feasibility of $(m = 3)$ at this phase, the increased cost in $|T_Y^O|$ that results from the insertion of the cities removed from B and C must also be considered. We denote this change of increased cost in $|T_Y^O|$ from inserting the cities removed for B and C as

$$\triangle T_Y = \sum_{i=B,C} \triangle T_{Y_i}, \tag{14}$$

where

$$\triangle T_{Y_i} = |T_Y^O + Y_i| - |T_Y^O|, \quad \forall i \in \{B, C\}. \tag{15}$$

Calculating $\triangle T_{Y_i}$, the increased cost of the tour T_Y^O, is not as straightforward as computing $\triangle T_i$, i.e., the cost reduced from the tour T_i when removing cities. The dependence of each procedure (inserting cities, removing cities) on the other demonstrates this. Specifically for the reduction of cost from removing any cities, the change $\triangle T_i$ is both known and unchanging at the same point that the cities in Y_i are defined, where i represents either B or C.

The insertion procedures, however, diverge from the independence maintained between removing the cities Y from T_B^O and T_C^O. In other words, for any set Y, the cost reduced from T_B^O is independent of the cost reduced from T_C^O. Conversely, the construction of Y depends on the cities removed from both B and C. Accordingly, any city from from either B or C will have a relative insertion cost associated with the completed tour T_Y^O constructed. Moreover, if each city is inserted into T_Y^O in the order it is removed from either B or C, the increase in cost will only be relative to the previously inserted cities. Hence, unless order of insertion corresponds directly with the exact order the cities are visited in for the optimal tour through Y, the cost will differ.

Selecting the cities to remove will have an associated cost that can be measured immediately, and a cost that is known only with the complete definition of both Y_B and Y_C. From observing that the inclusion of any prospective city in Y will have an analogous effect on the resulting tours, it is indisputable that the selection of Y must be made towards the following optimization objectives:

$$\begin{aligned} &Maximize \quad \sum_{i=B,C} \triangle T_i \\ &Minimize \quad \triangle T_Y \end{aligned} \tag{16}$$

subject to:

$$
\begin{aligned}
&|T^O_{i_Y}| \leq Z, \quad \forall i \in \{B, C\} \\
&|T^O_Y| \leq Z.
\end{aligned} \tag{17}
$$

The two objectives in Equation 16 establish, respectively, that Y should be selected such that (i) the total independent reduction from the tours, T^O_B and T^O_C, is maximized, and (ii) the increased cost of the tour, T_Y, is minimized.

To quantify the reduction in cost, $\triangle T_i$, and the cost increased, $\triangle T_{Y_i}$, resulting from the selection of cities, Y_i, their proportionality relationship is observed. In particular, consider the scenario where these rates of change are directly proportional such that

$$\triangle T_i = (\lambda_1) \triangle T_{Y_i}, \quad \forall i \in \{B, C\} \tag{18}$$

where the coefficient $\lambda_1 \in \mathbb{R}^+$, describes the proportionality constant.

Axiom 3.10. *If $\lambda_1 \leq 1$, then $|T_Y| \leq Z$ if*

$$\sum_{i=A,B} \triangle T_i \ \leq \ Z \tag{19}$$

It is apparent from Axiom 3.10 that when the total reduction in cost to remove the cities, Y, from the tours, T^O_B and T^O_C, is *at most* equal to the cost of inserting these cities into T_Y, the difficulty in constructing the optimal selection of Y is drastically reduced. Similarly for the case when $\lambda_1 > 1$, or at least is assumed for some upper bound, deriving Y will correspond directly to the cost of T^O_B and T^O_C exceeding Z. As the value of λ_1 is increased, the optimal value, m^*, would similarly be expected to increase. Nonetheless, quantifying the change of cost increased $\triangle T_{Y_i}$, is only possible when both Y_B and Y_C are defined. As a result λ_1 cannot solely define the proportionality constant between $\triangle T_i$ and $\triangle T_{Y_i}$, where i represents either B or C. Instead this relationship must be formalized such that the construction of each set, Y_B and Y_C, independently maximizes the reduction of the tour that cities are extracted from. Furthermore, the resulting set, Y, must satisfy the constraints throughout the construction of the two sets (i.e., Y_B and Y_C). Theorem 3.11 is subsequently defined from these observations and lemmas.

THEOREM 3.11. $(m = 3)$ *is feasible when* Y_i *is selected such that* $|T_{i_Y}^O| \leq Z$ *and* $\triangle T_{Y_i} = (\lambda_2)\triangle T_i$ *where* $\lambda_2 = (\triangle T_i\ (m-1))^{-1}$, $\forall\ i \in \{B, C\}$.

PROOF. If removing Y_i from T_i^O results in a tour cost that is at most equal to Z, then an additional tour through all the cities in Y that has a cost no more than Z will prove that $(m = 3)$ is feasible. Therefore the average increase in cost to T_Y resulting from adding Y_i should be equal to $Z(m-1)^{-1}$. The value of λ_2 can be determined by setting $\triangle T_i(\lambda_2) = (m-1)^{-1}$, accordingly, and thus $\lambda_2 = (\triangle T_i\ (m-1))^{-1}$. Under these circumstances, the cost of T_Y will be no more than Z, proving this theorem. □

This theorem defines the maximum value of λ_2 for $(m = 3)$ to be feasible. Recall that λ_1 was previously used to represent the proportionality constant between the increase to T_Y^O resulting from adding the cities Y_i and the cost reduced from T_i^O with the removal of these cities. λ_2 similarly defines this relation but instead the coefficient is defining the increased cost to a new tour as a result of the reduced cost for the cities defined by Y_i for all $i \in \{B, C\}$.

COROLLARY 3.12. *For any three tours that verify the feasibility of* $(m = 3)$ *when* $\alpha > 2$*, the cost to insert the cities in* Y_i *to construct* T_Y^O*, is at most* $(\alpha - (m-1))^{-1}$ *times the reduction of cost from removing* Y_i *from* T_i^O*,* $\forall i \in \{B, C\}$.

PROOF. From Lemma 3.7 and Lemma 3.8 we know that $|T_i^O| \approxeq (m-1)^{-1}|T_A|$, where T_i^O represents each tour T_B^O and T_C^O independently that were derived by splitting T_A. These lemmas demonstrate the maximum cost of the tour as $|T_i^O| \leq \alpha(m-1)^{-1}Z$, given that $|T_A| = \alpha Z$. At minimum, Y_i must be selected such that its removal from T_i^O results in a cost that no longer exceeds Z. Otherwise when $|T_{i_Y}^O| > Z$, the selection of cities, Y_i, will not verify that $(m = 3)$ is feasible. Consequently the minimum value of $\triangle T_i$ can be calculated from these constraints. Hence the minimum required reduction required by any selection of cities to define Y_i is defined as

$$\begin{aligned}\triangle T_i &= Z\left(\frac{\alpha}{m-1}\right) - Z \\ &= Z\left(\frac{\alpha - m + 1}{m-1}\right)\end{aligned} \tag{20}$$

Furthermore $\triangle T_{Y_i} \leq (m-1)^{-1}Z$ for the tour T_Y^O to be feasible. Otherwise $|T_{i_Y}^O| > Z$ when the average cost from inserting Y_i is more than $(m-1)^{-1}$ times the total allowed cost. Similar to Theorem 3.11, we define the proportionality constant between the variables denoting the change in cost. Substituting for the constraints defined in Theorem 3.11's proof results in the following:

$$\triangle T_{Y_i} = (\lambda_2)\triangle T_i \quad \Rightarrow \quad \frac{Z}{(m-1)} = (\lambda_2)\left(\frac{\alpha - m + 1}{m-1}\right) Z \tag{21}$$

Following Equation 21, λ_2 is simplified as

$$\begin{aligned}\lambda_2 &= \frac{(m-1)^{-1}Z}{\left[(\alpha - m + 1)(m-1)^{-1}\right] Z} \\ &= \frac{1}{\alpha - m + 1} \\ &= (\alpha - (m-1))^{-1}\end{aligned} \tag{22}$$

Therefore if $(m = 3)$ is assumed to be feasible, the maximum insertion cost $\triangle T_{Y_i}$ of Y_i such that $|T_Y^O| \leq Z$ is at most $(\alpha - (m-1))^{-1}$ times $\triangle T_i$ (the cost to remove the set Y_i) when $\alpha > 2$. □

Following Corollary 3.12, Axiom 3.13 states a significant property resulting from the proportionality constraints associated with selecting a set of cities Y from T_B^O and T_C^O.

Axiom 3.13. *For $(m = 3)$ to be feasible when $\alpha > 3$, the increase in cost resulting from the insertion of the cities, Y_i, $\forall i \in \{B, C\}$, into the new tour T_Y^O must always be less than the cost reduced by its removal from the existing tour T_i^O.*

This axiom identifies one of the more complex properties in solving this problem.

This is particularly the case when the cost of the initial optimal tour through X exceeds the allowed time by more than a factor of 3.

Verifying the feasibility of $(m = 3)$ with certainty in this scenario presents an additional combinatorial problem. Namely, even by assuming a procedure can optimally construct the set Y, a resulting tour, T_Y^O, that exceeds the allowed time (i.e., Z) would not prove with certainty that $m = 3$ tours are not feasible. This would require a more thorough investigation of alternative approaches to deriving a set of three disjoint TSP tours. Approaches independent of the single optimal tour constructed through the entire encapsulating region would be particularly of interest. Nonetheless, the properties illustrated in this section can still provide the characterizations of a heuristic structure for determining a tight bound on the minimum required vehicles for the OVRP-UT.

In the following section we present a probabilistic approach for estimating the value of m^* without explicitly defining the third tour, T_Y^O.

CHAPTER 4

PROBABILISTIC ESTIMATION OF FLEET SIZE FEASIBILITY

Algorithm 1 defines a partitioning heuristic presented in Chapter 3 for calculating the lower bound, $\Omega(m)$, of the OVRP-UT where m denotes the minimum number of vehicle tours required to satisfy the OVRP-UT criteria. Moreover, $m^* \leq \Omega(m)$, where m^*, is used to denote the optimal solution to the OVRP-UT. The partitioning heuristic attempts to verify if either $(m = 1)$, $(m = 2)$, or $(m = 3)$ independent tours will satisfy the OVRP-UT criteria. If $\Omega(m) > 3$ for the set A, Algorithm 1 is recursively called on two disjoint subsets of A.[1]

The feasibility of the $(m = 1)$ and $(m = 2)$ solutions are each verified through the explicit definition of 1 and then 2 optimal tours, respectively. In other words, each of the two solutions, $(m = 1)$ and $(m = 2)$, are verified by constructing their respective tour(s) in a specific manner, such that if a tour exceeds Z (i.e., the maximum time allowed) it verifies that the solution is not feasible. Algorithm 1 additionally defines an abstract heuristic strategy (i.e., the ExtractCities procedure) to verify the feasibility of $(m = 3)$ total tours through A for when $\Omega(m) > 2$ is assumed. This strategy defines a set of cities, Y, to extract from the two disjoint subsets, $B \subset A$ and $C \subset A$, where the cities in each of the three resulting subsets all define a feasible tour. Furthermore, constraints are defined in Chapter 3 that restrict how the set Y of extracted cities must be selected. If Y satisfies these constraints accordingly, the resulting subsets will verify the feasibility of $(m = 3)$ tours. In other words, $\Omega(m) > 3$ is assumed when the resulting three tours do not satisfy the OVRP-UT criteria.

Due to the complexity of the aforementioned constraints, no viable procedure was presented in the previous chapter for explicitly defining three tours in these circumstances. With this in mind, a strategy is presented in this chapter to verify the feasibility of $(m = 3)$ tours by estimating each tour's associated cost as the number of cities in A tends towards infinity. Accordingly, the feasibility of the solution can be determined without explicitly defining: (i) the extracted cities that define the set Y; or (ii) the three resulting tours that

[1] The subsets are disjoint with respect to the customers in A and are assumed to both include the depot.

verify the solution's feasibility. The cost associated with a tour is instead quantified based on the expected cost of an optimal TSP tour through all cities within a region. The term *region* is used to defined a set of cities according to their geographic context.

In the following sections, the procedures that verify the feasibility of ($m = 3$) tours for the corresponding region are formally defined. Section 4.1 initially introduces the required fundamental concepts for the remainder of this chapter. Namely, the *probabilistic analysis of partitioning algorithms* presented by Karp [37] is the focus of the section. Section 4.2 demonstrates how we have extended the research presented by Karp [37] to quantify regions of unequal sizes according to the expected cost of the optimal TSP tour resulting from the partitioning heuristic presented in Chapter 3. Section 4.3 employs these cost estimation strategies to determine the feasibility of ($m = 3$) for a region without explicitly defining the vehicle routes. The chapter is concluded by discussing the advantages and drawbacks of this approach in Section 4.4.

4.1. Approximating Tour Cost for Random Points in the Plane

The techniques presented in this chapter is an extension to the research presented in *Probabilistic Analysis of Partitioning Algorithms for the TSP in the Plane*, by Karp [37]. In particular, some of the characteristics presented by Karp are decoupled to be utilized in a different logistical context. For this reason, the work presented by Karp [37] is briefly outlined in this section. However, the probabilistic analysis of Karp's algorithm is based on the the Beardwood-Halton-Hammersley Theorem (BHH)[10].

The BHH Theorem. The BHH theorem generally states that the cost of an optimal TSP tour through n points tends to grow at the rate $\sqrt{n}$ when n is large enough. Moreover, with a probability of one, the length of the shortest tour through n points, as $n \to \infty$, will be asympototic to $\beta\sqrt{n}$, where β is a positive constant which does not depend on the distribution of n.

The BHH theorem is commonly used in determining the shortest path through any independent identically distributed (iid) random variables [30]. This constant, β, in the BHH theorem has been extensively studied in the domain of probability theory and combinatorial optimization. Arlotto and Steele [5] state that "sophisticated numerical computations", such as is presented in [4], has demonstrated $\beta \approx 0.714$, although only $0.62499 \leq \beta \leq 0.91996$ is known with certainty [26]. This constant is used to calculate the expected performance of the partitioning algorithm presented by Karp [37] for the TSP through n random points in the plane.

Until this point, all references to a TSP tour were defined by the optimal permutation of all cities within some given set. The set was consequently a description of the cities contained within it, but otherwise provided no further information about the cities. Namely, the distribution of the cities. With this in mind, let $T(X)$ denote the optimal TSP tour for all cities located in region X. Thus, instead of defining a set solely by the cities it contains, we define a region of cities according to some distribution. The notation $N(X)$ will continue to denote the number of cities within the region X as previously defined.

Following the notation used in [37], the region X is explicitly defined as a rectangular region $a \times b$. It is assumed that a and b are fixed and denote the dimensions as the width and length respectively. Additionally the length, b, is assumed to always be positioned along the longer edge of the rectangle. The basic measurements of perimeter and area are denoted, respectively, as $per(X)$ and $v(X)$. It is further assumed that the cities model a random distribution of points in a defined region on the plane. Accordingly, let $\Pi_n(X)$ denote n cities that are placed in the region X, according to a two-dimensional Poisson distribution. Moreover, let γ denote the rate parameter (i.e., the average number of cities per area) of this two-dimensional Poisson process in the plane. By definition, $\gamma = E[N(X)] = Var(N(X))$, where $E[N(X)]$ and $Var(N(X))$, denote respectively, the expected number (i.e., the mean) of cities in X and the associated variance value. Hence, for some region $A \subseteq X$ where $X \in \mathbb{R}^2$, the probably of $N(A)$ being exactly equal to i total cities is

$$Pr[N(A) = i] = \mathrm{e}^{\gamma}\frac{\gamma^i}{i!}, \text{ where } \gamma = nv(A), \text{ for } i = 0, 1, 2, ... \tag{23}$$

where Equation 23 is the probability mass function of the Poisson distribution. Furthermore, given that $N(X) = n$, the expected number of cities in region A will be

$$E[N(A)] = n\left(\frac{v(A)}{v(X)}\right). \tag{24}$$

From this distribution of cities, [37] introduces a partitioning algorithm that divides the region X into 2^k equally sized sub-regions such that the expected number of cities in each is equal to t. The variable $k = \lceil \log_2(n-1)/(t-1) \rceil$, where t denotes the maximum number of cities that can be optimally solved for a TSP tour. Moreover it is assumed that k is even. These sub-tours are then stitched together to form a walk, denoted as W_n, resulting in a tour through all n cities in X. It is shown that the expected cost of this walk is as follows

$$E(|W_n|) = \sqrt{n}\Big(\beta_X(t) + O\Big(t^{-7/6}\Big)\Big) \quad [37], \tag{25}$$

where the term $\beta_X(t)$ is similar to the positive constant shown in the BHH Theorem but that instead is dependent on X. This length is comprised of two components: (i) the sum of the shortest tour's length within each sub-region; and (ii) the back and forth cost of the arcs connecting each tour to form the walk. For the purposes of this research, we will only be interested in the first contributing factor of this expected cost. To demonstrate this first part we denote the expected cost of these equally sized sub-regions as $E[|Y_i|]$ where $Y_i \in \{Y_1, Y_2, ..., Y_{2^k}\}$. Karp shows that the expected cost of each sub-region is $E[|Y_i|] = \beta_X(t)\sqrt{t}$. Further the study shows that the expected cost of all of the sub-regions combined is equal to

$$\sum_{i=1}^{2^k} E[|Y_i|] = 2^k\Big(\beta_X(t)\sqrt{t}\Big)2^{-k/2} \quad [37], \tag{26}$$

since each of the 2^k regions are like X but scaled down by a factor of $2^{-k/2}$. Simplifying this equation algebraically as shown in Equation 27 demonstrates the performance of Karp's partitioning method.

$$\begin{aligned} 2^k\Big(\beta_X(t)\sqrt{t}\ \Big)2^{-k/2} &= 2^{k/2}\Big(\beta_X(t)\Big)\sqrt{t} \\ &= 2^{k/2}\Big(\beta_X(t)\sqrt{n}\ \Big)\sqrt{2^{-k}} \\ &= \beta_X(t)\sqrt{n} \ \ [37] \end{aligned} \tag{27}$$

In particular Karp proves that for any value of t that $\beta_X(t) - \beta \leq \frac{6(a+b)}{\sqrt{t}}$, but also conjectures that $\beta_X(t) - \beta$ is likely proportional to $t^{-1/2}$ such that

$$\beta_X(t) - \beta = O\big(t^{-1/2}\big) \ \ [37]. \tag{28}$$

This cost relationship between these individual TSP tours within each partition of X and the optimal TSP through X is a crucial component in estimating the minimum required vehicles for the OVRP-TU.

The following subsections presents a strategy based on the partitioning heuristic presented in Chapter 3 to estimate the feasibility of $(m = 3)$ for a region without requiring the solution to be explicitly defined. Instead the total expected cost of these tours will be quantified from their respectively defined regions by leveraging this relationship.

4.2. Expected Tour Cost for Partitioned Regions

Unlike the partitions derived by the approach present by Karp, regions resulting from the partitioning heuristic presented in Chapter 3 are not guaranteed to be equally sized. This introduces a few complications that violate constraints required to quantify the expected cost of an optimal TSP tour traversing the cities within each independent region.

4.2.1. Extracting the Scale Factor

Deriving the total expected cost from an odd number number of regions (e.g., $m = 3$) is directly associated with the requirements defined by our partitioning approach. With this in mind, we rearrange Equation 25 to eliminate the variable k.

$$
\begin{aligned}
\sum_{i=1}^{2^k} E[|Y_i|] &= 2^k\left(\beta_X(t)\sqrt{t}\right)2^{-k/2} \\
&= \left(2^k\right)\left(\sqrt{t}\ \right)\left(\beta_X(t)\right)\left(2^{-k/2}\right) \\
&= \left(2^k\right)\left(\sqrt{2^{-k}}\ \right)\left(\sqrt{n}\ \right)\left(\beta_X(t)\right)\left(2^{-k/2}\right) \\
&= \left(2^{k/2}\right)\left(\sqrt{n}\ \right)\left(\beta_X(t)\right)\left(2^{-k/2}\right) \\
&= \left(\sqrt{(n)(t^{-1}}\ \right)\left(\sqrt{n}\ \right)\left(\beta_X(t)\right)\left(2^{-k/2}\right) \\
&= \left(\sqrt{n}\ \sqrt{n}\ \right)\left(\sqrt{t^{-1}}\ \right)\left(\beta_X(t)\right)\left(2^{-k/2}\right) \\
&= \left(\frac{n}{\sqrt{t}}\right)\left(\beta_X(t)\right)\ 2^{-k/2} \\
&= \left(\frac{n}{\sqrt{t}}\right)\left(\beta_X(t)\right)\ \left(\frac{per(Y_i)}{per(X)}\right)
\end{aligned}
\tag{29}
$$

Although both equations (25 and 29) are equivalent the latter defines the scale factor by the coefficient of proportionality between the perimeters of the partitioned regions Y_i, and X. Still, all of the resulting regions for either representation must still adhere to the following criteria:

(i) all regions must be the same size,

(ii) no two regions can have any intersecting space,

(iii) the union of all of the regions must equal X exactly.

Because the scale factor is defined by the specific scenario in contrast to the $2^{-k/2}$ term, the rearrangement will apply to both even and odd number of partitions. For example, consider partitioning the rectangular region, $X : \{a \times b\}$, into 4 equally sized sub-regions. For Equation 25, $k = 2$ and thus $2^{-k/2} = 2^{-1}$. In comparison to Equation 29, each region Y_i in this scenario will represent $Y_i : \left\{\frac{a}{2} \times \frac{b}{2}\right\}$. Thus,

$$
\frac{per(Y_i)}{per(X)} = \frac{2[(a/2) + (b/2)]}{2(a+b)} = 2^{-1}, \tag{30}
$$

demonstrating the equality between the two equation's scale factor when the regions are all of equal size and k is even. However, if X is instead partitioned into 3 equally sized regions[2] the resulting regions are defined as $Y_i : \left\{a \times \frac{b}{3}\right\}$. In this scenario only Equation 29 accurately defines the factor for each region to be like X but scaled down.

4.2.2. Directional Cutting Strategies

The dissociation between the scale factor and the number of partitions lets us quantify an odd number of regions. However being able to define the number of partitions must still be addressed. Recall that t denotes the maximum number of cities that can be optimally solved for a TSP tour. In that case, the number of resulting partitions is defined by the value assigned to t. Alternatively for the OVRP-UT, the partitions define the m total independent tours in the solution. In other words, the total number of partitions (i.e., tours) is *directly* associated with the optimal solution, m^*. With this in mind, to define the required total number of regions resulting from the partition, denoted as M, we alternatively let $t = n/M$. Each region will then identically define their number of expected cities, t, from the value assigned to M accordingly.

Defining the number of partitions is an important component in our goal of estimating the feasibility of $(m = 3)$ for a region without explicitly defining the tours in the solution. As a consequence of defining the number of equally sized regions, the strategy for selecting the direction of cuts for the partition differs from that presented by Karp [37]. The partitioning procedure presented in [37] is based on a recursive cutting strategy that align the direction of each cut to be parallel with the longest edge of the region in that phase. By continually cutting the longer edge at each step the total perimeter for the resulting regions will be minimized. Consequently this "cutting game" strategy demonstrates an optimal approach in minimizing the total combined cost of the TSP in each region. Conversely if a cut is made in the opposite direction of the optimal direction (i.e., parallel with the shorter side) the region will obviously result in a higher cost. At the same time, partitioning X with cuts

[2]Two vertical cuts through the longer edge, b, of X is the only possible option resulting in equally sized regions with a minimum total sum of perimeters.

parallel to the shorter side are an unavoidable consequence when an odd number of regions are required. With this in mind we examine a strategy that partitions X using only vertical cuts.

Similar to the previous constraints it is assumed that the regions resulting from only vertical cuts will all be of equal size, while being like X but scaled down. Thus if X is partitioned into M total regions resulting from $M-1$ vertical cuts then each region Y_i is defined as the rectangular region, $Y_i : \left\{a \times \frac{b}{M}\right\}$. Hence we can represent the total expected cost of the combined regions as the following:

$$\begin{aligned}
\sum_{i=1}^{2^k} E[|Y_i|] \;\Rightarrow\; \sum_{i=1}^{M} E[|Y_i|] &= \frac{n}{\sqrt{t}}\,\beta_X(t)\,\frac{per(Y_i)}{per(X)} \\
&= \frac{n}{\sqrt{n/M}}\,\beta_X(t)\,\frac{M[a+(b/M)]}{M(a+b)} \\
&= \sqrt{M}\left(\beta_X(t)\sqrt{n}\right)\left(\frac{Ma+b}{M(a+b)}\right),
\end{aligned} \tag{31}$$

where the expected number of cities in the regions is simply $t = \frac{n}{M}$. Equation 31 is denoted in the same format of Equation 29, to present the cost illustrated by Karp [37] by the coefficient of proportionality between the partitioned regions and X. By comparison, the total expected cost of the regions in X that are subsequently partitioned by $M-1$ cuts only through b will exceed those of the optimal cutting strategy when $a > \frac{b}{M}$. To relate these strategies further, Equation 31 is simplified algebraically (as shown in Equation 32) to appear in a manner similar to that of the optimal cutting strategy.

$$
\begin{aligned}
\sqrt{M}\left(\beta_X(t)\sqrt{n}\right)\left(\frac{Ma+b}{M(a+b)}\right) &= \beta_X(t)\sqrt{n}\sqrt{M}\left(\frac{Ma+b}{M(a+b)}\right) \\
&= \beta_X(t)\sqrt{n}\sqrt{M}\left(\frac{Ma}{M(a+b)}+\frac{b}{M(a+b)}\right) \\
&= \beta_X(t)\sqrt{n}\sqrt{M}\left(\frac{a}{a+b}+\frac{b}{M(a+b)}\right) \\
&= \beta_X(t)\sqrt{n}\left(\frac{Ma+b}{M^{1/2}(a+b)}\right)
\end{aligned} \tag{32}
$$

This rearrangement also clearly demonstrates the factor by which these strategies differ. By substituting M (the total number of regions) with its original notation 2^k,

$$
\begin{aligned}
\left(\frac{Ma+b}{M^{1/2}(a+b)}\right) &\Rightarrow \left(\frac{(2^k)a+b}{\sqrt{2^k}(a+b)}\right) \\
&= \left(\frac{(2^{k/2})(a)+(2^{-k/2})(b)}{a+b}\right)
\end{aligned} \tag{33}
$$

scenarios that produce this difference in cost can be related directly to the dimensions of X. Specifically the expected cost of the 2^k regions partitioned by only vertical cuts will exceed the optimal cutting strategy when $b < a * 2^{k/2}$.

4.2.3. Quantifying the Expected Tour Cost of Unequally Size Partitions

Quantifying the expected cost for regions resulting from our partitioning heuristic still require an additional issue to be addressed. Namely, the regions to quantify will almost always result in varying sizes. Two generalized scenarios are presented to illustrate these issues and their resolutions. Accordingly, $X : a \times b$ will continue to define the rectangular regions with n randomly distributed cities within. Both scenarios will assume that the region X has been partitioned into two regions B and C, such that $B \subset X$ and $C \subset X$.

Additionally since $N(X) = n$ for the cities randomly distributed in X, then

$$E[N(B)] = n\left(\frac{v(B)}{v(X)}\right) \text{, where } v(B) \text{ denotes the area of region } B. \tag{34}$$

In the first scenario let regions B and C be disjoint, $X = \{B \bigcup C\}$, and $per(B) = per(C)$. Therefore both regions are equally represented by the dimensions $\{a \times \frac{b}{2}\}$. Moreover we can easily derive a constant value of t as

$$t \ = \ n\left(\frac{v(B)}{v(X)}\right) \ = \ n\left(\frac{a(b/2)}{ab}\right) \ = \ \left(\frac{n}{2}\right), \tag{35}$$

representing a single vertical cut made directly down the center of X. Thus $E[|T(B)|] = E[|T(C)|]$, given the restriction that both regions be equally sized. As a result, the total combined expected cost is calculated by the following:

$$\begin{aligned} \sum_{i=B,C} E[|T(i)|] \ &= \ \sqrt{2}\ \beta_X(t)\sqrt{n}\ \frac{2a+b}{2(a+b)} \\ &= \ \beta_X(t)\sqrt{n}\left(\frac{2a+b}{2^{1/2}(a+b)}\right) \end{aligned} \tag{36}$$

Alternatively for the second scenario we assume that the regions are no longer required to be of equal size. The remaining constrains are otherwise upheld for this scenario. Accordingly let region B and C's dimensions be defined as $\left\{a \times \left(\frac{2b}{3}\right)\right\}$ and $\left\{a \times \left(\frac{b}{3}\right)\right\}$, respectively, so that B defines two-thirds of X horizontally and C defines the remaining third. Conversely to the first scenario, $E[N(B)] \neq E[N(C)]$, as a result of the area of their respective regions not being equal. However, a single constant value of t is still required quantify the expected tour cost of the regions. Since the area of B is not equal to C, the expected number of cities in each will also not be the same. Yet since $E[N(B)] = \frac{2n}{3}$ and $E[N(C)] = \frac{n}{3}$, splitting X into three equal slices each representing $\frac{1}{3}$ of X allows a value for t to easily be calculated. By representing the total combined expected cost as

$$\beta_X(t)\sqrt{n}\left(\frac{3a+b}{3^{1/2}(a+b)}\right), \tag{37}$$

such that C will represent one of these slices at a third of the cost while B represents twice that of C. As a result quantifying the expected costs can be represented by the following:

$$\begin{aligned} E[|T(X)|] &= E[|T(B)|] + E[|T(C)|] \\ &= \beta_X(t)\sqrt{n}\left(\frac{3a+b}{3(a+b)}\right)\left(\frac{2}{\sqrt{3}}\right) \;+\; \beta_X(t)\sqrt{n}\left(\frac{3a+b}{3(a+b)}\right)\left(\frac{1}{\sqrt{3}}\right) \\ &= \beta_X(t)\sqrt{n}\left(\frac{3a+b}{3^{1/2}(a+b)}\right) \end{aligned} \tag{38}$$

The next subsection describes how these strategies can be applied in the recursive partitioning algorithm presented in Section 3.4. Namely, the feasibility of $(m = 3)$ tours in a region is determined without explicit tour definitions.

4.3. Employing Cost Estimations to Determine Feasibility of Fleet Size

In this section we assume a set of cities are distributed in $A : \{a \times b\}$ according to the two-dimensional Poisson distribution $\Pi_n(A)$. In other words, the cities are distributed in the plane, defined by the rectangular region, A. Furthermore we assume that $n-1$ cities are those to be visited for the OVRP-UT. The remaining city therefore denotes the depot.

Recall from Section 3.4 that T_A is partitioned into two tours (T_B^O, T_C^O) and we conjecture that $m > 2$ if either have an associated cost greater than the allowed time, Z. Then a subset of cities, Y, must be removed from T_B^O and T_C^O. For $(m = 3)$ to be feasible the selection of Y must satisfy the constrains defined in Theorem 3.11. Despite the constraints being known, Axiom 3.13 identifies the complexities in proving this feasibility through explicitly defined solutions. The objective in this section is to determine the feasibility of $(m = 3)$ for the region A without requiring the solution to be explicitly defined. Instead we will quantify the tours based on their expected optimal cost.

In order to quantify the tours based their defined region, we convert the tours T_B^O, T_C^O into regions B, C, respectively. However quantifying $E[|T(A_1)|]$ for some region $A_1 \subseteq A$, is derived directly from the proportional expected cost $E[|T(A)|]$. Although it was previously proven that $|T_B| \approxeq |T_C| \leq |T_A|/2$, the bounds of these regions in the plane might not be disjoint. Additionally, the bounding box for each is not guaranteed to represent the dimensions $\{a \times (b/2)\}$. Thus simply defining the region from a tight bounding box around the cities included in a tour would not be sufficient. Instead we construct the bounding box for the rectangular regions, B and C, for the tours T_B^O and T_C^O, respectively, with the following steps and conditions:

(i) Let $A : \{a \times b\}$ be positioned such that b defines the longest boundary edge and is positioned as the horizontal size

(ii) Let $l(A)$ denote the vertical edge defining the left boundary of A and $r(A)$ similarly for the right boundary of A

(iii) Assume that no two cities have an identical horizontal positional value in A (i.e., no two cities will be the same distance from either the left or right boundary in A)

(iv) Let the tour that includes the city with the closest distance to any point on $l(A)$ denote T_B^O; T_C^O will then denote the remaining tour

(v) Select the city included in T_B^O that is closest to any point on $r(A)$; similarly for T_C^O but instead select the city in its tour that is closest to any point on $l(A)$

(vi) Then let the boundary of B be defined by assigning the left boundary, $l(B)$, to equal $l(A)$; define $r(B)$ as a vertical line parallel with $l(B)$ that passes directly through its city closest to $r(A)$; similarly define C as $r(C) = r(A)$, and $l(C)$ as a vertical line parallel with $r(C)$ that passes directly through its city closest to $l(A)$

(vii) Let the height of $l(B)$, $r(B)$ and $l(C)$, $r(C)$ all be equal to a, the height of A

(viii) Let both regions' top and bottom boundaries stretch from their individual left and right boundaries such that they are horizontally aligned exactly with A's top and bottom edges

This construction of B and C, representing the partitioned tours in A, guarantees that $A = \{B \cup C\}$ since both tours will include the depot location. Therefore the union between these regions will cover the entirety of A. In contrast, if the depot is removed from both tours, T_B^O and T_C^O, before the regions B and C are constructed, it is possible that once constructed the boundary of B does not overlap the boundary of C. To illustrate, assume that both regions, B and C, exist on an (x, y)-coordinate grid. Then let x_1 and x_2 denote the horizontal positions of $r(B)$, and $l(C)$, respectively. If these regions are disjoint, sharing only the depot in common, then $x_1 == x_2$. Alternatively for all other scenarios, regions B and C will position their inside boundary edge (i.e, their only vertical edge not aligned with any edge of A) such that $x_1 > x_2$. Hence, it is possible for the boundaries of each region to overlap with the other.

Computing the expected optimal cost of a TSP tour for regions that overlap in this manner is not effective or feasible. This is due to the fact that the resulting tour for any region is assumed to visit all of the cities within. Hence, regions that overlap would presumably result in tours that all visit the same cities in the intersecting space of the regions. As a result, the cities located within the intersecting space between B and C, would be included in the cost of both tours through the two regions. Furthermore, we cannot derive the value of t from the conditional distribution, when $N(A) = n$ is a result of the assumed multinomial distribution. Nonetheless, the remainder of this section will illustrate how scenarios with overlapping regions actually present favorable conditions when estimating a lower bound of required tours for the OVRP-UT. For this purpose, we continue to denote Y as the set of cities to be removed from B and C such that the feasibility of $(m = 3)$ can be determined.

Initially we set $Y = \{B \ \cup \ C\}$, such that the boundary of Y is defined by the intersection space of the partitioned regions. By removing Y from both B and C, three disjoint regions $(Y, \{B \setminus Y\}, \{C \setminus Y\})$ will remain. Furthermore a union of these resulting regions will equal X exactly. Hence $E[N(Y)]$, $E[N(\{B \setminus Y\})]$, and $E[N(\{C \setminus Y\})]$ (i.e., the expected number of cities within each region) is now known.

Deriving the value of t (i.e., the expected number of cities in a region), is the only step remaining to quantify the regions by their associated expected optimal TSP cost. From the examples previously illustrated, it was assumed that

$$\frac{E[N(B)]}{E[N(C)]} \in \mathbb{N} \ . \tag{39}$$

However we cannot realistically make this assumption, thus additional computation is likely required to derive the value of t. This will indeed be the case unless the two regions with the largest expected number of cities can both be divided by the expected number of cities in the remaining region such that the solution is $\mathbb{N}$. For this reason, each region will be represented by a combination of some number of contiguous disjoint subregions of equal size.

For clarity's sake, each of these aforementioned subregions are individually referenced as a single "*slice*". Every slice is assumed to be the same size. Moreover, the height is assumed to be equal to the height of the rectangular region A. As a consequence of equally sized sices, a constant value denotes the expected number of cities, t, in a single slice. Hence, the value of t will dictate the resulting number of slices included in calculating the total combined expected cost. To guarantee that each region can be represented by a contiguous combination of disjoint slices, each region's expected number of cities must be divisible by t, such that the solution is a positive natural number. The result of each of these divisions will denote, accordingly, the exact number of slices required to represent the region.

To select the value to assign to the variable t, we first observe from Equation 28 (as presented by Karp [37]), that the error is reduced as the value of t is increased. For this reason, we derive t by the Greatest Common Factor (GCF) of each $E[N(Y)]$, $E[N(\{B \setminus Y\})]$, and $E[N(\{C \setminus Y\})]$ region. Accordingly, t is assigned the maximum possible value for the expected number of cities in each slice.

Each region, Y, $\{B \setminus Y\}$, and $\{C \setminus Y\}$), can now independently quantify their expected cost of an optimal TSP by the *(total expected cost of a tour through A)* $\times$ *(*n^{-1}*)* $\times$ *(number expected cities in a sub-region).* We formally define this described cost in Equation 40 for each region as:

$$
E[|T(P)|] = \Big(\beta_X(t)\sqrt{n}\Big)\left(\frac{(n\ t^{-1})\ a+b}{\sqrt{n\ t^{-1}}\ \ (a+b)}\right)\Big(n^{-1}\Big)\Big(E[N(P)]\Big),
\qquad \forall P \in \{Y,\ \{B \setminus Y\},\ \{C \setminus Y\}\}\ , \tag{40}
$$

where $t = GCF\Big\{\ E[N(Y)],\ E[N(\{B \setminus Y\})],\ E[N(\{C \setminus Y\})]\ \Big\}$.

To illustrate this cost let us assume that $n = 76$ and for $Y = \{B \cup C\}$, that $E[N(Y)] = 16$. Further let $E[N(\{B \setminus Y\})] = 36$ and $E[N(\{C \setminus Y\})] = 24$, and thus $t = 4$. Therefore each region can be quantified by their individual expected optimal tour cost as:

$$
\begin{aligned}
E[|T(Y)|] &= \Big(\beta_X(t)\sqrt{n}\,\Big)\left(\frac{19a+b}{19^{1/2}(a+b)}\right)\left(\frac{16}{n}\right)\\
E[|T(\{B \setminus Y\})|] &= \Big(\beta_X(t)\sqrt{n}\,\Big)\left(\frac{19a+b}{19^{1/2}(a+b)}\right)\left(\frac{36}{n}\right)\\
E[|T(\{C \setminus Y\})|] &= \Big(\beta_X(t)\sqrt{n}\,\Big)\left(\frac{19a+b}{19^{1/2}(a+b)}\right)\left(\frac{24}{n}\right)
\end{aligned} \tag{41}
$$

given that $n/t = 19$. The combined total cost of these three regions is then equal to:

$$
\sum_{P \in \{Y,\ \{B \setminus Y\},\ \{C \setminus Y\}\}} E[|T(P)|] = \beta_X(t)\sqrt{n}\left(\frac{19a+b}{19^{1/2}(a+b)}\right), \tag{42}
$$

with the error $t^{-1/2} = 19^{-1/2} = 0.2294157339$ times the optimal expected cost of a tour through the entire region, $E[|T(A)|]$.

The expected cost for a optimal tour through each region Y, $\{B \setminus Y\}$, and $\{C \setminus Y\}$ can now be estimated, and accordingly the feasibility of $(m = 3)$ for X can be determined. Furthermore if either $E[|T(\{B \setminus Y\})|] > Z$ or $E[|T(\{B \setminus Y\})|] > Z$ but $E[|T(Y)|] < Z$, it is possible for Y to iteratively remove additional space from these regions until either: (i) $E[|T(Y)|]$ will exceed Z, or (ii) all three regions have an expected cost at or below Z. Each iteration would accordingly then remove the space equivalent to one slice as previously derived to quantify the regions. Modifications to the cost of the regions (for adding to or removing from) are restricted to be only in increments of the expected cost of a single slice:

$$\left(\beta_X(t)\sqrt{n}\right)\left(\frac{(n\ t^{-1})\ a+b}{\sqrt{n\ t^{-1}}\ \ (a+b)}\right)\left(\frac{t}{n}\right) \tag{43}$$

If it is determined that $(m > 3)$ for region X from the three resulting regions' estimated cost, we repeat the recursive partitioning heuristic (presented in Section 3.4) until a feasible bound is derived for m^*. Accordingly the bounding regions constructed for estimating the partitions cost would be disregarded. Instead the tours T_B^O and T_C^O, originally partitioned from T_A will each define the new region to determine the feasibility in the next recursive phase.

4.3.1. Depot Inclusion without a Known Location

Determining the expected cost of an optimal tour through the cities in a region requires the cities to be placed based on a random distribution. The first two phases of the algorithm that partition the optimal tour are not dependent upon the distribution of the cities. Hence, any distribution can be assumed for these first two phases. Determining the feasibility of three tours, however, was only achieved without explicitly defining a solution. As a result, the reference to the depot's exact positioning is lost as a result of this process. Location specific information in particular for all of the n cities, not just the depot, is not applicable once the expected cost is computed for the regions as it is measured asymptotically. Since the depot is only known to be located somewhere within region Y, the exact cost to connect the depot with tours in the other regions is unknown. Nevertheless, the two contributions representing this cost: (i) inserting the depot into the tour; and (ii) removing an edge from the city adjacent to the depot that is incident to another city, can be estimated accordingly. Further this estimations is calculated for each region $\{P \setminus Y\}$, where $P = A, B$, independently.

The first contribution can be estimated as follows. The cost of inserting the depot consists of connecting any city in $\{P \setminus Y\}$ to the depot in Y. Since the region P is guaranteed to include both the tour within $\{P \setminus Y\}$ and the depot, the distance between any city in

$\{P \setminus Y\}$ and the depot will never exceed the distance of the longest strait line possible in P. Therefore if w and h denote the dimensions for the rectangular region, $P : \{w \times h\}$, then $L = \sqrt{w^2 + h^2}$, and is the upper bound for the cost of the first contribution. Alternatively, a better approximation would be $L/3$, the expected value of distance between two random points on a line that is of the length L [33].

For the second contribution, recall that the the OVRP-UT only requires the tour to begin at the depot. Because the expected cost of the tour in $\{P\setminus Y\}$ is representing a optimal TSP (i.e., a closed tour), an edge breaking this cycle can be removed; thereby reducing the overall tours cost. First we let i denote the position of the city within permutation defining the tour through $\{P \setminus Y\}$ that is adjacent to the depot resulting from the previous contribution. The second contribution is then defined by the removing the edge incident to the cities at position $i+1$ and $i-1$ with the maximum cost. Therefore the additional cost of connecting the depot will be reduced by the $max\{c_{i-1,i}, c_{i,i+1}\}$. However the cities adjacent to i could have an incident edge with a distance anywhere between the smallest and largest distance between any two cities in the region $\{P \setminus Y\}$. As a result, the contribution can be estimated as reducing the cost by $\sqrt{n}$, the rate described in [10] as the cost of an optimal tour through n points tends to grow at when $n \to \infty$.

4.4. Advantages and Drawbacks of the Partitioning Heuristic

Quantifying tours based on the expected optimal cost of the partitioned regions allows us to derive a lower bound to the number of required vehicles for the OVRP-UT. Proving the tightness of the bound, however, is constrained by a few drawbacks in this approach. Describing these drawbacks and the overall advantages of our approach is the objective of this section.

In Section 3.4 we define constraints (theorem 3.11) for a partitioning outcome to satisfy in order to prove that a solution for the OVRP-UT of a given region is feasible with only three total vehicle tours. Yet determining that an explicitly derived solution is *the* optimal solution for all problem instance is not feasible. In contrast this axiom does assert a critical heuristic regarding the absolute cost in removing a city from a tour to insert it into another. In other words, when structuring an optimal partition, the cost of removing cities from a tour should be at a rate greater than the cost to insert the cities into another tour. Accordingly, this heuristic is reflected in our partitioning approach by defining the intersecting space of regions B and C as the initial set of cities to be remove from. As a result the expected cost of a tour through this overlapping space, $E[|T(Y)|]$, will tend to equal half the sum total difference in the expected cost being reduced from B and C by removing Y. This consequence is observed from the theorem presented by Karp [37], stating that "*For all t,* $\beta_X(t) - \beta \leq 6(a+b)/t$ ", where β is the constant in the Beardwood-Halton-Hammersley Theorem [10]. Hence the length of $T_t(X)$, the tour through t cities in X, will tend to equal $T_n(X)$ where $t < n$. To that end it can be determined that the expected cost of the cities in $\{Y \setminus C\}$ (or conversely for $\{Y \setminus B\}$) will tend to be the same as all of the cities in Y.

Furthermore this heuristic assumes that the single optimal tour through all cities in a region is associated with three optimal open tours that cover all of the cities in the region without overlap (i.e., each city is visited only once). The proof to this is not included with this study, although we do conjectured in Section 3.4 that it is likely the case.

Another potential drawback of this approach is the effect that an initial region's defining dimensions can have on the error of the expected cost for an optimal tour through the region. This is due to the vertical-only cutting strategy employed to be able to apply the research presented in [37] to our problem. Although as shown in Section 4.2.2, increasing the error is not exclusively the only outcome. For instance, if applied on a region where one dimension significantly exceeds the other, the error can actually be improved. Namely, for a rectangular region, $A : \{a \times b\}$, the error is improved when constructing 2^k regions using the vertical-only cutting strategy, when $b > a * 2^{k/2}$.

CHAPTER 5

PARTITIONING HEURISTICS FOR THE CAPACITY CONSTRAINT

As first introduced in Chapter 1, the open vehicle routing problem with uniform time constraints (OVRP-UT) presented an anomalous priority on the core objective of minimizing the number of vehicles required. Instead of minimizing the total cost of tours, as most variants of the VRP most commonly represent as the core objective, the OVRP-UT only requires the tours to be completed within the allowed time. This variant provides an adequate model of practical instances of the VRP that correspond to high-consequence constraints. As previously mentioned, optimization of relief networks in relation to the mitigation of an epidemic disaster is one such motivating instance. In this scenario, mitigation efforts include supplying the effected population with the associated medical countermeasures (MCMs) from a set of point of dispensing centers (PODs) within an limited time frame. Defining the set of vehicle tours that optimally transport the MCMs from the receiving-storing-staging (RSS) warehouse to the PODs within the time allowed has been the focus of this study. However the heuristics presented previously in chapters 3 and 4 are primarily focused on the time constraint. Hence the complexities associated with constraints of the available vehicle capacity for each tour are not considered in the preceding analysis. This chapter conversely analyzes the performance of general partitioning heuristics as it relates to these capacity constraints accordingly. Nevertheless, the minimization of vehicles will remain the core optimization objective for the problem. In a similar manner to the previous chapters, we analyze the performance of a partitioning heuristic constrained by a uniform vehicle capacity, independently from the constraints on time.

5.1. Generalizing the OVRP-UT for Capacity Constraints

We denote the open vehicle routing problem with uniform capacity constraints (OVRP-UC) as a generalization of the OVRP-UT where minimizing m, the number of required vehicles, remains the primary objective. However, we assign $Z \leftarrow \infty$, removing the constraint on time allowed for the completion of all tours. Instead, given the set of all locations,

$V_n = \{0, 1, 2, ..., n\}$, where index 0 denotes the depot and the remaining n elements denote the set of customers, the cost of tours are quantified by the total demand of the customer's visited on the tour. Accordingly, let d_i denote the demand of customer i, where $i = \{1, 2, ..., n\}$ representing all of the customers in V_n. Further let d_i be distributed according to some distribution defined as $d_i \in (a, b]$, and $0 \leq a \leq b \leq Q$, where Q denotes the identical vehicle demand capacity for all vehicles. For the remainder of this chapter we will assume that the customer demands are sorted in a non-increasing order so that

$$d_1 \geq d_2 \geq ... \geq d_n \ . \tag{44}$$

Each vehicle tour is denoted as T_j, where $j = \{0, 1, ..., m-1\}$ representing the set of all m tours. The total demand associated with a single tour is denoted as $\tau(T_j)$. This is calculated as the sum of all customers' demand served by tour T_j, where $j = \{0, 1, ..., m-1\}$ for the m tours respectively. As a consequence of excluding the uniform time constraint, we can simply describe the OVRP-UC as partitioning the set of n customers into a minimum number of m disjoint sets (i.e., tours), subject to the following constraints: (i) all customers must be visited once, (ii) no customer demand shall be split between multiple tours (i.e., each customer must only be visited by a single tour), and (iii) the total sum of demand for the customers within in a single tour must not exceed Q (i.e., the vehicle capacity).

When the objective to minimize the number of required tours is presented in this manner, it is analogous to the one dimensional *Bin Packing problem* (BPP). The objective for both problems can be informally described as minimizing the number m of bins each with the identical capacity Q. With this in mind, the terms *bins* and *items* are used interchangeably with *tours* and *customer demands*, respectively, for the remainder of this chapter as it relates to the current context being discussed. Furthermore we continue to let m^* denote the optimal solution representing the absolute minimum number of tours (i.e, bins) required. Leveraging existing literature for the BPP, the performance of a general partitioning heuristic is observed in solving the OVRP-UC. The analysis of lower bounding procedures for the BPP, such as those presented in [59] and [48] are of particular interest. The worst-case asymptotic performance of these lower bounds demonstrated in [15], [31] and [22] are additionally of use in this chapter.

Considering that the OVRP-UT will always include the time constraint by definition, explicitly defining a lower bound procedure for the OVRP-UC is admittedly not very beneficial in solving a capacitated OVRP-UT instance. Therefore the objectives in this chapter is to demonstrate the effect that a partitioning procedure will have on the performance of optimal capacity utilization. The partitioning procedure in this chapter is similar to the partitioning heuristic presented in Chapter 3. With this in mind, the applicable BPP lower bound procedures that currently exist are presented in Section 5.1.1. The construction of an upper bounding procedure subsequently follows in Section 5.2. Specifically this bound is derived by solving an optimization function that models an optimal partitioning towards the (i) minimization of discrepancy of total demand between the partitions [36], and the (ii) maximization of capacity usage.

5.1.1. Overview of Lower Bound Procedures

In this section we observe two lower bound procedures presented by Martello and Toth in 1990 [48], motivating the construction of the upper bound presented in the following section.[1] It is assumed that all units of measure for the capacity and demand in this chapter are homogeneous.

Lower Bound L_1

The lower bound procedure, L_1, presented in [48] implies an obvious lower bound to the BPP resulting from a continuous relaxation of the problem. It is computed in $O(n)$ time as

$$L_1 = \left\lceil \frac{\tau(d)}{Q} \right\rceil , \tag{45}$$

[1]Martello et al., denote L_1 and L_2 as both the lower bound procedure and the value computed for a specific BPP instance.

where $\tau(d)$ denotes the sum total demand of all n customers calculated as

$$\tau(d) = \sum_{i=0}^{n} d_i \ . \tag{46}$$

To obtain this bound it is assumed that individual customers' demands can be split between different tours. Conversely if phrased in the context of a BPP, the demands and tours represent the items and bins respectively. Moreover, let P represent an instance of the BPP, and $\bar{P}$ be the relaxed instance of P constructed under the assumption that demands can be split between tours. Thus for $\bar{P}$ we observe that $m^* = L_1$ as customers can be iteratively assigned to a tour until the demand of the customers assigned exceeds Q. This process is repeated for the construction of a new additional tour until all n customers are assigned to a tour. The worst-case performance ratio of L_1 for P, when $P \neq \bar{P}$, is

$$\frac{L_1}{m^*} \geq \frac{1}{2} \ \text{[48]}. \tag{47}$$

Lower Bound L_2

The lower bound, L_1, is best suited for problem instances where the customer demands, d_i, are small in relation to Q. Conversely Martello et al., 1990 [48] additionally present the lower bound procedure, L_2, for instances where the size of customer demands are large with respect to Q. Accordingly, instances of this type limit the amount of customers that can be assigned to each tour. To obtain the lower bound L_2, the customers with a demand that exceeds half of the total capacity (i.e., $Q/2$) are partitioned into two subsets. The remaining customers are assigned to a third subset. This partitioning exploits the non-feasible pairings for customer demands whose sum would exceed Q. The worst-case performance ratio of L_2 is proven as

$$\frac{L_2}{m^*} = \frac{3}{4} \ \text{[48]}. \tag{48}$$

Although the formal definition of L_2 is beyond the scope of this research, the intrinsic observations of this bound are beneficial nonetheless. A detailed review of both lower bound procedures L_1, and L_2, are alternatively provided in [15], and additionally analyze these lower bounds both analytically and computationally.

These lower bound procedures are leveraged in the following sections. Namely, Section 5.2 constructs an upper bound for the OVRP-UC that is based on an optimal capacity utilization that exploits the L_1 bound.

5.2. Constructing an Upper Bound for the OVRP-UC

From the lower bound procedure, L_1, it is clear that splitting any customer's respective demand between two or more tours yields a substantial improvement in minimizing m. The OVRP-UC, however, strictly restricts any single customer's demand from being divided in such a manner. Moreover, obtaining the lower bound L_1 is admittedly not always feasible for most practical problem instances. Despite L_1 not always being a good indicator of the optimal value (i.e., m^*) for all instances, it is beneficial nonetheless. Defining the relationship between m^* and the reduction of wasted capacity. Using notation that is similar to that presented in [31], we let $W(V_n)$ denote the total wasted (i.e., unused) capacity. The value for $W(V_n)$ is calculated as

$$W(V_n) = Qm - \tau(d) \ , \tag{49}$$

where m is the minimum number of tours calculated for the OVRP-UC. Initially by assuming the lower bound L_1 as the derived value for m, the total waste will be

$$\begin{aligned} W(V_n) &= QL_1 - \tau(d) \\ &= Q\left\lceil \frac{\tau(d)}{Q} \right\rceil - \tau(d) \ . \end{aligned} \tag{50}$$

If $\left\lceil \frac{\tau(d)}{Q} \right\rceil = \frac{\tau(d)}{Q}$, accordingly, all m tours will have completely utilized the entire available capacity. In other words, the total wasted capacity, $W(V_n)$, is equal to 0. Hence the value of m is a result of the proportion of capacity used. We conversely represent this as $\tau(d)/m$. So for this case where $m = L_1$ with no wasted capacity, the capacity used equals exactly the value of Q.

Given that the optimal value, m^*, is not currently known, we can only infer that the proportion of capacity used must be no greater than Q. Otherwise the solution would not be considered feasible and thus is not an optimal solution. Further, we recall that the OVRP-UC restricts any single customer demand from being split between multiple tours. From these circumstances we observe that any value of m resulting from any algorithm, where $m \geq m^* \geq L_1$, defines the total customer demand to serve for each tour. This is additionally simplified as $m \geq L_1$.

With the association between minimizing m, and the maximization of capacity usage for each tour in mind, we let λ_j denote the proportion of capacity used in T_j. Using the total demand of each tour's respective customers, λ_j is calculated as:

$$\lambda_j = 1 - \frac{\tau(T_j)}{Q} \tag{51}$$

Further observation shows that as the value of $\frac{\tau(d)}{m}$ tends towards Q, the value of m similarly tends towards m^*.

In the following section (sec. 5.2.1) we construct an upper bound by further constraining the OVRP-UC. The worst case performance of this bounding procedure is presented in Section 5.2.2.

5.2.1. Maximizing Identical Usage of Capacity

Given that the optimal value m^* is associated with the minimization of the total unused capacity, we define an upper bound, U_1, based on a general recursive partitioning heuristic. Accordingly, U_1 is constructed by solving the following optimization problem:

$$Minimize \quad m \tag{52}$$

s.t.,

$$\begin{aligned} \tau(d) &= \lambda_j Q m, \qquad j = \{1, 2, ..., m\} \\ \lambda_j &\leq 1, \qquad\qquad j = \{1, 2, ..., m\}, \end{aligned} \tag{53}$$

which consequently implies that

$$\sum_{j=1}^{m} \lambda_j = \lambda_k, \quad k = \{1, 2, ..., m\} \ . \tag{54}$$

We let $\bar{\lambda}$ denote a constant value for the identical proportion of capacity used as for all m tours. This ensures that the total demand of all customers is equal to the sum total of capacity used for all of the tours. The second constraint just further enforces this restriction by preventing any tour's total demand served from exceeding its capacity. As a consequence of these constraints, maximizing capacity utilization will result in the optimal solution. In other words, this formulation is defined to exploit the association previously observed between minimizing m and maximizing capacity utilization. With this in mind, U_1 is calculated as the minimum required vehicles that solves this optimization problem.

To calculate U_1, let μ denote the number of times that the total demand of all customers exceeds the allowed capacity.

$$\mu = \frac{\tau(d)}{Q}, \tag{55}$$

where $\mu \in [0, \infty)$. Given that all tours use must use an equal proportion of their capacity, they must similarly have an equal proportion of capacity remaining. Thus the minimum number of tours required can be derived from the value of h, which denotes the minimum number of times that μ must be equally divided into two, recursively, until the result equals at most 1. This partitioning procedure is exemplified in Figure 5.1. The minimum number of recursive levels, h, must result in the following inequality:

$$\frac{\mu}{2^h} \leq 1 \ , \tag{56}$$

where lemma 5.1 describes the value of h.

LEMMA 5.1. *The minimum value of h that is required for the inequality, $\frac{\mu}{2^h} \leq 1$ to hold is $h = \lceil \log_2(\mu) \rceil$.*

PROOF. We observe that 2^h must be at most equal to μ, otherwise $\frac{\mu}{2^h} > 1$. Moreover if

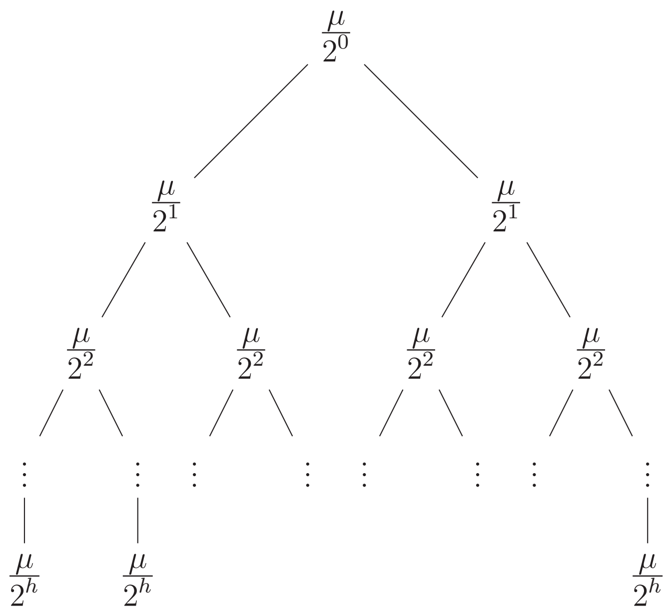


FIGURE 5.1. Each node represents the proportion of demand to the allowed capacity resulting from each phase of a general equal partitioning, where h denotes the height of the full binary recursion tree.

$2^h = \mu$ then $\frac{\mu}{2^h} \Rightarrow \frac{\mu}{\mu} = 1$. Hence this inequality will not hold, accordingly, for any value of 2^h that exceeds μ. When rearranging Equation 56 using the binary logarithm:

$$\begin{aligned} \frac{\mu}{2^h} \leq 1 \quad &\Rightarrow \quad \mu \leq 2^h \\ &= \quad h \leq \log_2(\mu) \end{aligned} \tag{57}$$

Observing that $\lceil \log_2(\mu) \rceil \leq \log_2(\mu)$, we know that h must be at most equal to $\lceil \log_2(\mu) \rceil$ since $h \leq \lceil \log_2(\mu) \rceil \leq \log_2(\mu)$. Accordingly $\mu/2^{\lceil \log_2(\mu) \rceil}$ must be at most equal to 1, given that

$$\frac{\mu}{2^{\lceil \log_2(\mu) \rceil}} \quad \leq \quad \frac{\mu}{2^{\log_2(\mu)}} \quad = \quad 1 \tag{58}$$

Therefore the minimum value of h for $\frac{\mu}{2^h} \leq 1$ is $\lceil \log_2(\mu) \rceil$. □

Due to the fact that $\frac{\mu}{2^h} \leq 1$ when $h = \lceil \log_2(\mu) \rceil$, accordingly, the value of $\bar{\lambda}$ is calculated as

$$\bar{\lambda} = \frac{\mu}{2^{\lceil \log_2(\mu) \rceil}} \ . \tag{59}$$

In this case, the value of m can be solved since all of the variables required to solve for m are now known since the values for $\tau(d)$ and Q are both defined by the specific problem instance.

$$\tau(d) = \bar{\lambda}Qm \quad \Rightarrow \quad \frac{Q}{\bar{\lambda}Q} = m \tag{60}$$

Alternatively, the equation is simplified by substituting the value of $\bar{\lambda}$ as

$$\begin{aligned} \tau(d) = \bar{\lambda}Qm \;\; \Rightarrow \;\; \tau(d) \;&= \left(\frac{\mu}{2^{\lceil \log_2(\mu) \rceil}}\right)Qm \\ \Rightarrow \;\; \tau(d) \;&= \left(\frac{\tau(d)}{2^{\lceil \log_2(\mu) \rceil}Q}\right)Qm \\ &= \frac{\tau(d)m}{2^{\lceil \log_2(\mu) \rceil}} \, , \end{aligned} \tag{61}$$

in order to solve for the value of m:

$$\begin{aligned} \tau(d) = \frac{\tau(d)m}{2^{\lceil \log_2(\mu) \rceil}} \;\; \Rightarrow \;\; 1 \;&= \frac{m}{2^{\lceil \log_2(K) \rceil}} \\ \Rightarrow \;\; m \;&= 2^{\lceil \log_2(\mu) \rceil} \, . \end{aligned} \tag{62}$$

This results in $m = 2^{\lceil \log_2(\mu) \rceil}$ tours, each with the identical demand of $\bar{\lambda}Q$. Hence the upper bound of from this optimization is

$$U_1 = 2^{\left\lceil \log_2(\mu) \right\rceil} \quad \Rightarrow \quad 2^{\left\lceil \log_2\left(\frac{\tau(d)}{Q}\right) \right\rceil} \tag{63}$$

A consequence from the construction of this bounding procedure is that $m^* \leq U_1$ is not always guaranteed. Instead this bound is only guaranteed for OVRP-UC instances when m^* is obtained by supposing the division of individual customer demands. For this reason, the worst-case performance of this bound is quantified in the next section for instances when splitting the demands is not assumed.

5.2.2. An Upper Bound for a Continuous Partitioning Algorithm

The ientical distribution of customer demands is admittedly not likely in practical instances of this problem. However, existing literature does demonstrate the benefit of mitigating a bio-disaster by setting up dispensing facilities such that they all serve approximately the same number of people [34]. Hence uniform resource demand for each facility is often a priority to achieve in this instance. In this section we quantify the performance of U_1 in the worst-case scenario. The focus is centered around an existing continuous partitioning algorithm. In particular the algorithm similarly reflects the heuristics presented in the proceeding chapters for optimizing the time restricted variant of this problem. As a consequence of maintaining a consistent heuristic structure, the properties independently associated with optimizing the OVRP-UT and OVRP-UC are leveraged to design strategies to solve more complex variants of the problem.

Jimenez et al. [34] introduce a universal partitioning algorithm (UPAS) [34] that optimizes the geographic placement of resource dispensing facilities in the event of a bio-emergency. UPAS defines the placement of these facilities by constructing non-overlapping regions. Further, the resulting regions are constructed such that the amount of required resources for each region will be equal or near-equal to the all of the remaining regions. The resources often correspond directly with the affected population to serve within their respective region's boundary. This assumes that all individuals within a partition are served by their closest facility, and that they are all served with the same type of resource.

We abstractly extend the UPAS algorithm to construct uniform partitions to solve the OVRP-UC. Namely, the set of customer demands (i.e., d_i) define the items to partition. This replaces the set of census regions quantified by the total population within each of them of which UPAS is designed for. Since none of the census regions are split between partitions in UPAS, it will represent all instances of the OVRP-UC. This is critical in analyzing the worst-case performance of U_1 when the division of individual customer demands are not assumed.

The resulting partitions obtained from appplying the UPAS algorithm will each denote a tour serving all of the customers' demand assigned to this partition. Therefore we let UPAS(k) denote the resulting k tours from the UPAS algorithm where k is a positive integer required as input to the algorithm. Constructing k partitions from this generalization of the UPAS algorithm for solving the OVRP-UC presents two conflicts that must be noted. The first of these relates to the sole optimization objective of the OVRP-UC, i.e., minimizing the total number of vehicles. In contrast, the objective of the UPAS algorithm is to construct k total partitions while minimizing the discrepancy between each partition's total population assigned (or customer demand for this use case). The second issue is that UPAS does not enforce a capacity restriction for each partition. These issues are directly related as the minimum number of tours required is a consequence of the capacity available. To resolve these conflicts, it can be assumed that UPAS(k) is repeated with an arbitrarily high value of k (i.e., the number of total partitions) initially, that is then reduced until a feasible solution is constructed. Apart from these issues the construction of k partitions from applying the UPAS algorithm does indeed adequately represent a OVRP-UC solution. For additional procedural details of the UPAS algorithm, we refer the reader to the referenced article, Jimenez et al.[34], as this detail is outside the scope of our analysis. The performance analysis presented in this article, however, is crucial in proving the worst-case performance of U_1.

With the objective of determining the worst-case performance of U_1, k is assigned this upper bound according. Thus the UPAS algorithm will attempt to construct k partitions such that $\tau(d)/k \leq Q$. Theorem 5.2 defines the worst-case case performance of U_1. Figure 5.2 exemplifies such a solution demonstrating this worst-case performance.

THEOREM 5.2. *Let k equal the upper bound, U_1, obtained for an OVRP-UC instance with n customer demands, $d_i, \forall i = \{1, 2, ..., n\}$. Further, assume d_i is distributed according to any distribution $d_i \in (a, b]$, such that $0 \leq a \leq b \leq Q$. Then* $\text{UPAS}(k)/m^* \geq 2^{-1}$, *represents the worst-case performance for the upper bound U_1.*

PROOF. To prove the theorem 5.2 we first observe the error in UPAS(k) constructing k identical partitions each with the optimal partition size $\tau(d)/k$. Jimenez et al. prove asymptotically that

$$\lim_{k \to \infty} \triangle_{max} = 2d_{max} \ , \tag{64}$$

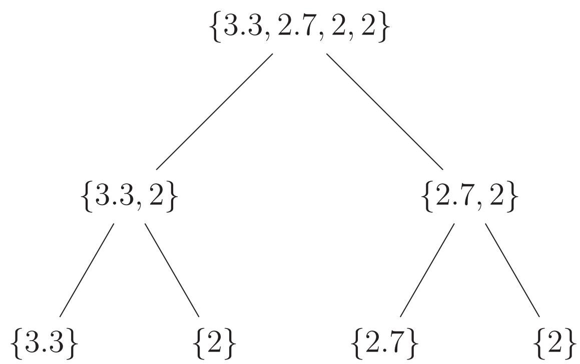


FIGURE 5.2. A scenario where $\{3.3, 2.7, 2, 2\}$ denote the set of customer demands to partition. $\tau(d) = 10$ and the capacity $Q = 5$. The upper bound $U_1 = 2$, yet the minimum tours required in this scenario, 4, is demonstrated.

where $\triangle_{max}$ is the maximum difference between any two partitions for k approaching infinity, and d_{max} is the maximum customer demand of the n customer. It is additionally shown that the total demand assigned to any of the k partitions $\leq d_{max}$. From this error bound we observe that each of the k partitions, respectively, will have a total demand within $(\tau(d)/k) \pm d_{max}$. As a result, it is possible for all k partitions to have a demand equal to $(\tau(d)/k) + d_{max}$ in the worst-case. All that remains is to determine if the addition d_{max} to the *perfect partition* value, $(\tau(d)/k)$, would exceed the capacity Q. Since k is equal to the lower bound U_1, the proportion of capacity used is assumed as $\bar{\lambda}$, where $\bar{\lambda} \leq 1$. Hence we alternatively represented the partitions by their proportion of capacity used. In other words, the proportion of capacity used is $\bar{\lambda} + (d_{max}/Q)$, by assuming $\bar{\lambda} = 1$ for the worst-case. Then if d_{max} equals Q, or any other value in $(0, Q]$ such that $\bar{\lambda}Q + d_{max}$ exceeds Q, it is possible for the demand of all k partitions to exceed Q. Conversely, since $\bar{\lambda} \leq 1$ and $d_{max} \leq Q$, none of the k partitions will exceed $2Q$. It then follows that $\bar{\lambda}Q + d_{max} \leq 2$, proving $\text{UPAS}(k)/m^* \geq 2^{-1}$ in the worst-case. $\square$

CHAPTER 6

TWO-PHASE PARTITIONING ALGORITHM

The objective of this chapter is to provide an efficient and flexible meta-heuristic algorithm for the VRP under *high-consequence* constraints, such as those commonly identified in responding to a disaster situation. Bio-emergencies in particular call for a specific hierarchy of optimization priorities. Accordingly, the focus of the algorithm presented in this chapter is the construction and optimization of vehicle routes that address these type of emergency response scenarios.

In contrast to the classical formulations of VRP variants (e.g., Capacitated VRP (CVRP), Multiple Depot VRP (MDVRP)), vehicle routing logistics for bio-emergency response introduce a hard constraint on time. Unless all resulting routes are completed within the maximum time allowed, mitigation efforts for the entire response plan can be severely impacted. Accordingly, the *high-consequence* term is used to categorize this constraint. The phases of responding to a bio-emergercy, as shown in Figure 2.1, motivate this type of classification. The moment a response plan is activated, the entire affected population must be treated within a specific time frame. This phase of mass prophylaxis is achieved most commonly by treating the population through strategically placed points of dispensing (POD) facilities. Moreover, the amount of time required to complete the mass prophylaxis can be determined [58]. By subtracting the time duration for this mass prophylaxis phase from the total time allowed between the plan activation and completion, the remaining time defines the hard time constraint for the delivery of resources to the PODs. As a result, all vehicle routes constructed under these circumstances must complete their tour within this maximum tour duration. A formal definition of this high-consequence VRP variant is presented in Section 6.2. Namely, the hard constraints for the maximum tour duration and the identical vehicle capacity are reflected in this optimization.

As previously illustrated in Section 2.2, most VRP formulations define a specific type of graph to model the logistical network. These abstract formulations are beneficial when applying this research across different domains. However, it is presumed that networks in most practical instances (e.g., real road network) can be modeled by these formulations. For this reason, a practical network for a classical road network is provided in Section 6.1. Additionally, the procedures to construct a complete graph representation of this road network are presented. The construction of this graph is utilized in Section 6.2 for the formal definition of the aforementioned VRP under high-consequence constraints. A two-phase partitioning algorithm is presented in Section 6.3 that constructs a set of vehicle routes that are optimized for this high-consequence VRP. The chapter concludes with an illustration of the algorithm's application on a simple network instance in Section 6.3.5.

6.1. Modeling the Road Network

Formally we denote the road network as the graph $G_R = (V, E)$, where $V(G_R) = \{v_1, v_2, ..., v_n\}$, and $E(G_R) = \{e_1, e_2, ..., e_k\}$, represent the vertex and edge set respectively, so that $E(G_R)$ defines k total road segments where the end points represent the n vertices. This reflects a common structure in geographic road network data sets where a single traditional road will usually consist of contiguous and intersecting road segments. As is typical in road networks, multiple paths may exist that could be traveled between any two points in the network. As a result, G_R, is assumed to be a multi-graph. Each path is defined by an ordered sequence of edges. In an effort of generalization, we make the assumption that any path between two locations has a symmetric cost resulting in the undirected graph. This assumption is based on the realization that most roads are (i) two-way, representing an equivalent costs to travel in either direction, or (ii) a comparable one-way road in opposite directions exists. For simplicity, each path is described only by its origin and destination location. We denote, Φ_{ab}, as the set of paths that exist from v_a to v_b, $\forall v_a, v_b \in V(G_R)$. According, let $\Phi^p_{ab} \in \Phi_{ab}$ denote the path, p, that begins at v_a and ends at v_b. The unique path, p, is specifically referenced since multiple paths can exist from v_a to v_b. Furthermore, this description of paths define the relation between $V(G_R)$ and $E(G_R)$, as a function $f : \{v_a v_b\} \rightarrow E(G_R)$, such that one or more mappings exist for any vertex pair $\{v_a v_b\} \in V(G_R)$, to the corresponding ordered sequence of edges, $e \in E(G_R)$, that define the path(s) from v_a to v_b. This assumes that the graph is connected accordingly.

The associated weight or cost with each edge can represent various measures such as the travel distance between two locations, classification of the road segments and travel time (i.e, distance $\times$ rate of speed). For the purpose of this research, we assume that the edge cost represents the time to travel the road segment. Accordingly, let τ_{ab}^{p}, denote the associated cost for traveling from v_a to v_b on the path, p.

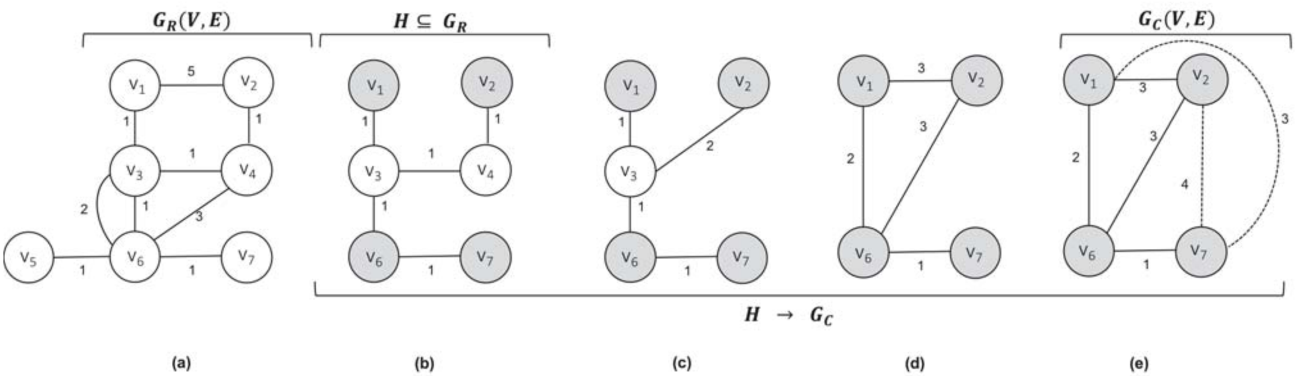


FIGURE 6.1. **(a)** illustrates a multi-graph, $G_R(V, E)$, for a traditional road network. $V(G_R) = \{v_1, v_2, ..., v_7\}$ denotes the set of geographic points in the road network. The edge set, $E(G_R)$, depicts the road segments between any pair of points in $V(G_R)$. The subset, $L = \{v_1, v_2, v_6, v_7\}$, denotes a set of locations to visit for a particular VRP problem, where $L \subseteq V$. (b) depicts the sub-graph, $H \subseteq G_R$, of the road network that only includes the geographic points and road segments that exist in at least one *least cost* path between any location pair in L. (e) illustrates a complete undirected graph, $G_C(V, E)$, commonly used as an abstract representation of the VRP for the locations, L, on the road network; the edge set, $E(G_C)$ denotes the shortest paths between all pairs in $V(G_C)$ derived from constructing the graph. (b) through (d) depicts the suppression and contraction of all geographic points not included in L; (e) illustrates the edges added (dashed lines) to achieve a complete graph.

Currently, G_R provides an accurate representation of a classical road network. Most common formulations of the VRP, however, provide a more abstract model of the network. Namely, if two or more edges exist for a pair of adjacent vertices, only the least cost edge is kept in the formulation. Furthermore, an undirected complete graph is commonly used to define the network of a VRP. For this reason, a sub-graph, H, is constructed from G_R to model a network, optimized for common routing operations (e.g., determining the shortest path), for a VRP instance. An abstract representation, denoted as G_C, is subsequently derived from the sub-graph, H, of this road network. This final representation of the road network is utilized for the VRP presented in Section 6.2.

To define the procedures for the construction of H and G_C, let L, define the set of locations in the road network that are defined for a VRP instance. In other words, let $L \subseteq V(G_R)$ denote the set of customer locations to visit, and at least one depot location that all vehicle routes must begin at. Figure 6.1 illustrates the construction of the graphs H and G_C on a simple road network, G_R, where $L = \{v_1, v_2, v_6, v_7\}$ is the set of locations defined by some VRP instance.

Constructing $H \subseteq G_R$

For the set of locations defined by some VRP instance, $L \subseteq V(G_R)$, the optimized routing network, H, defines a sub-graph of G_R that only includes the geographic points and road segments that exist in one or more *least cost* path between any location pair in the set L. Hence $L \subseteq V(H) \subseteq V(G_R)$ and $E(H) \subseteq E(G_R)$. Formally, we define the set of edges, $E(H)$, as the following

$$E(H) = \{e \in E(G_R) | (e \in \Phi^p_{ab}) \wedge (\tau^p_{ab} \leq \tau^l_{ab}), \forall a, b \in L, \forall p, l \in \Phi_{ab},\ p \neq l, a \neq b\} \quad (65)$$

where $V(H)$ is subsequently defined by the end points for all $e \in E(H)$. This process can be efficiently computed by applying Dijkstra's Algorithm [24] in finding the shortest path between all pairs and creating an edge at the obtained cost. Figure 6.1(b) depicts the sub-graph, H, for the defined set of VRP locations, L.

Constructing $G_C(V, E)$

The graph, $G_C(V, E)$, represents an abstract network for the set of locations, L, defined by some VRP instance. Therefore G_C is derived from the entire road network, G_R, but instead is defined in a manner that is consistent with common formulations of the VRP. G_C is constructed from the sub-graph H, such that $V(G_C) = L$. Accordingly, $E(G_C)$ results from augmenting H by (i) contracting every vertex $v \in V(H) \setminus L$, and then (ii) inserting an edge to join all pairs of non-adjacent vertices. In other words, following the first procedure, the second procedure, (ii), results in a complete graph by adding an edge on all vertices with a degree less than $|L|-1$. Figure 6.1 illustrates these procedures for constructing G_C for a set of locations, L, defined for some VRP instance. Specifically, Figure 6.1(b),(c) and (d) illustrate this first procedure by contracting the vertices, v_3 and v_4, since these points in the road network are not included in the set L. The second procedure is depicted in Figure 6.1(e), representing the graph, G_C, by adding an edge between the vertex pairs, v_1v_7 and v_2v_7.

In the following section (Section 6.2), the graph network model, $G_C(V,E)$ is utilized in the formulation of the aforementioned high-consequence VRP variant.

6.2. Mathematical Formulation

Utilizing the previously defined graph, G_C, constructed from the road network graph, we define a high-consequence variant of the VRP. To establish the context of the VRP, all the locations, $V(G_C)$, are assumed to represent either a depot location (i.e., where all vehicle routes must start at), or the location of a customer that must be visited in a vehicle route. Formally, let the subset, $D \subset V(G_C)$, represent all vertices $v \in V(G_C)$, where v is a depot. Similarly let the subset $C \subset V(G_C)$ represent all $v \in V(G_C)$ where v is a customer. Accordingly, $D = \{d_1, ..., d_\delta\}$ and $C = \{c_1, ..., c_{n-\delta}\}$ denote, respectively, the set of all depot locations and customer locations. Let $\psi(c)$ denote the demand asociated with each customer, $\forall c \in C$.

Additionally we define the solution to obtain for the VRP formulation by a set of vehicle routes, R. Let R_i, for each $i = 1, 2, ..., |R|$, denote a single vehicle route that defines an ordered sequence of vertices as

$$R_i = \langle v_{\phi(0)}, v_{\phi(1)}, ..., v_{\phi(k)} \rangle, \tag{66}$$

where $v_{\phi(t)}$, for each $t = 0, 1, ..., k$, is a vertex $v \in V(G_C)$ whose subscript, $\phi(t)$, denotes the tth position in the route and maps to a specific vertex in $V(G_C)$. This ordered sequence defines a vehicle route by representing the locations that the vehicle will visit, in the order they are visited in. As shown in Equation 66, the first index, $v_{\phi(0)}$, of each route represents the depot, $d \in D$, where the route will begin at. The sub-sequence $\langle v_{\phi(t)} \rangle_{t=1}^{k}$, denotes the customers, $c \in C$, and their order that the route will visit them in.

For this VRP formulation, all vehicles are assumed to have identical constraints. Specifically, each vehicle will have a capacity, denoted by the positive constant β, to serve the set of customers in its route. The total customer demand associated with a vehicle route is denoted as $\psi(R_i)$, calculated as

$$\psi(R_i) = \sum_{t=1}^{|R_i|} \psi(v_{\phi(t)}), \tag{67}$$

where $\psi(v_t)$ denotes the demand of the individual customer in the tth position of the route. Furthermore, all vehicle routes will be restricted to by a constant time constraint, denoted as α, which defines the maximum tour duration allowed. Accordingly, let $\tau(R_i)$ denote the total travel cost for the vehicle route, R_i, calculated as

$$\tau(R_i) = \gamma * |R_i| + \sum_{t=0}^{|R_i|-1} \tau_{v_{\phi(t)} v_{\phi(t+1)}}, \tag{68}$$

where γ denotes a constant unloading time that is assessed for each customer visited, and $\tau_{v_{\phi(t)} v_{\phi(t+1)}}$ represents the amount of time to travel from $v_{\phi(t)}$ to $v_{\phi(t+1)}$.

The VRP optimization problem is then defined as:

$$Minimize \quad |R| \tag{69}$$

subject to:

$$\tau(R_i) \leq \alpha, \quad i = 1, 2, ..., |R| \tag{70}$$

$$\psi(R_i) \leq \beta, \quad i = 1, 2, ..., |R| \tag{71}$$

$$\bigcup_{i=1}^{|R|}\{R_i - v_{\phi(0)}\} \;=\; C, \quad v_{\phi(0)} \in R_i \tag{72}$$

$$\bigcap_{i=1}^{|R|}\{R_i - v_{\phi(0)}\} \;=\; \emptyset, \quad v_{\phi(0)} \in R_i \tag{73}$$

$$v_{\phi(0)} \bigcap D \;\neq\; \emptyset, \quad v_{\phi(0)} \in R_i, \quad i = 1, 2, ..., |R| \tag{74}$$

Objective 69 is the optimization objective of this VRP variant, and is subject to Constraints 70, 71, 72, 73 and 74. Objective 69 minimizes the total number of vehicles required to serve all customers. In this formulation, Constraint 70 restricts every route to be completed within the allowed time. Constraint 71 requires that the resources demanded by the customers on the routes do not exceed the capacity of the vehicle serving it. Constraint 72 requires that each customer must be visited by a vehicle route. Constraint 73 restricts every customer to be visited only once. Constraint 74 forces a route to initially start at one of the depot locations. However, once the resources are delivered and the route is complete, there are no constraints forcing a vehicle to return back to its starting depot within the required time, instead representing a variant of the Open VRP (OVRP).

We define a solution to this problem formulation as the set of vehicle routes, R, that deliver the resources demanded from each customer. A solution is determined to be *feasible* when all routes $R_i \in R$ have satisfied the constraints (i.e., Constraint 70, 71, 72, 73 and 74). A solution is considered *optimal* when it is feasible *and* Objective 69 is minimized. Determining the optimal solution for most practical and non-trivial instances is not computationally feasible since this VRP variant is a generalization of the TSP [45]. Accordingly, a novel two-phase meta-heuristic algorithm is devised in the following section to construct a minimized set of vehicle routes.

The Vehicle Routing Problem: Notation	
$D \subset V(G_C)$	Set of provided depots in the road network
$C \subset V(G_C)$	Set of provided customers in the road network
R	Resulting set of vehicle routes
$R_i = \langle v_{\phi(0)}, v_{\phi(1)}, ..., v_{\phi(k)} \rangle$	Ordered sequence of locations defining the route $R_i \in R$
$v_{\phi(0)} \in R_i$	Depot, $d \in D$, defining the starting location in R_i
$\langle v_n \rangle_{n=1}^{k}$	Ordered sub-sequence of customers, $c \in C$, in R_i
$\psi(R_i)$	Total capacity required to serve all customers in R_i
$\tau(R_i)$	Total travel time required to visit all customers in R_i
α	Maximum tour duration allowed for all routes
β	Maximum capacity for any vehicle in R
γ	Time required for unloading at each customer in the route

Table 6.1: Mathematical model notation for the VRP

6.3. Two-Phase Spatial Meta-Heuristic

In this section we present a meta-heuristic algorithm to solve the problem formulated in Section 6.2. Initially, a high-level overview of the heuristic strategy is presented. The details of the two-phase spatial meta-heuristic algorithm subsequently follow in Section 6.3.1.

Heuristic Overview

To solve our VRP formulation we designed a heuristic that is partially based on the decomposition of the problem into two natural phases generally described as:

(i) **Vertex Clustering**: In this initial phase the customers required to be served are strategically placed into a known constant number of clusters. Membership within each cluster represents the selected customers that are to be incorporated into a single feasible route.

(ii) **Route Construction**: This phase exploits the individual clusters to independently construct a single optimal route for each cluster. Subsequently this phase is the reduction of the VRP by solving a mutually independent set of TSPs so that each route defines the optimal order to visit the customers that are members for the route's respective cluster.

This decomposition of the VRP into these two phases originates from the heuristic stategy put forth by Fisher and Jaikumar [25]. In the initial phase clusters are formed by solving the Generalized Assignment Problem (GAP). Algorithms based on this strategy have previously shown the ability to produce quality solutions while benfiting from accelorated computational execution times [8]. However, heuristics for the GAP often assume that the vehicle fleet size is known. This represents an important distinctive characteristic that deviates from the high-consequence VRP variant defined in Section 6.2.

Based on the benefits observed from the decomposition of the problem, we adopt the underlying strategy by approaching the problem from its two natural phases. Due to the unique conditions enforced by the high-consequence VRP formulation, we alter the initial phase of the problem decomposition and present a heuristic based on spatial partitioning. In particular, the universal partitioning algorithm (UPAS) [34] that optimizes the geographic placement of resource dispensing facilities in the event of a bio-emergency is observed. Jimenez et al. [34] define the placement of facilities to be optimal when every facility has a uniform resource demand or population to be served or treated. UPAS partitions the geographic region of the affected population into a specified number of sub-regions, with the objective of each sub-region maintaining spatial contiguity and a equal population count assigned. We abstractly extend the conceptualization of assimilating uniform spatial partitions put forth in UPAS. Assuming equal capacity availability for each vehicle in the fleet to route, a similar partitioning can be conceived to optimize the customers to be delivered by each route. We embed these observed characteristics into the procedures of the multi-phase algorithmic framework presented in Section 6.3.1. The algorithm incorporates both phases of the problem decomposition, accordingly, to solve the high-consequence VRP variant defined in Section 6.2.

6.3.1. An Algorithmic Framework

Algorithm 4 formally describes the process of constructing a minimized set of open vehicle tours required to serve all customers. Without loss of generality, it is assumed that a minimum of two vehicle routes are required for the context of this research. For smaller

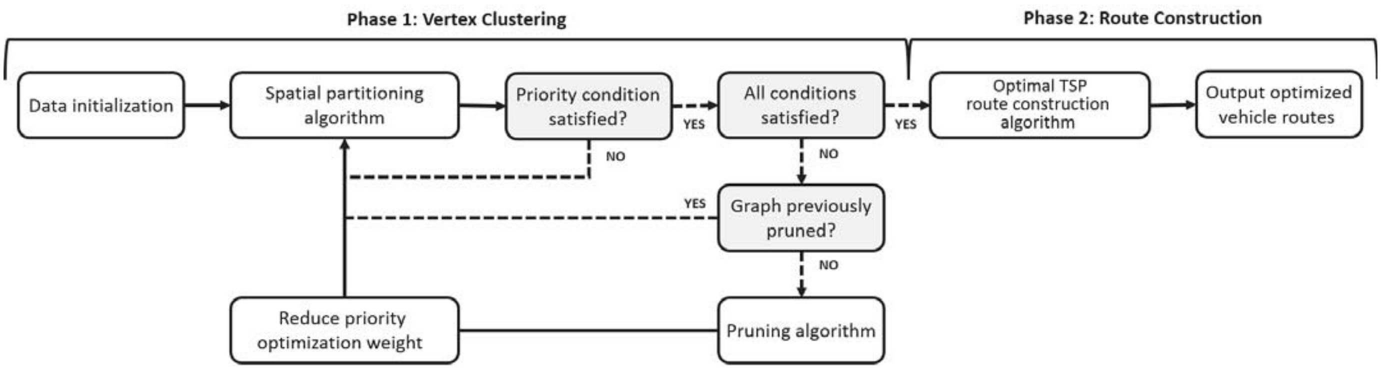


FIGURE 6.2. Multi-phase algorithm framework

instances that only required a single route, the problem can be generalized as a TSP instead. In that case, the exact and approximation algorithms reviewed in Chapter 2 are better suited for solving a TSP instance. Nonetheless, this algorithm can easily be extended to construct a single tour from the initial input to test the feasibility of a single tour. In particular, an existing/known TSP algorithm can be selected for this procedure based on solution quality and computational efficiency preferences.

Figure 6.2 illustrates the flow of Algorithm 4 through both phase 1 (vertex clustering) and phase 2 (route construction). The initial phase employs a priority tiered, multi-stage termination criteria when determining the minimized number of unique clusters of customers to each be constructed into vehicle routes in phase 2. The construction of these clusters in the initial phase primarily consists of two core procedures: continuous partitioning the set customers into new clusters for optimal route construction, and identifying optimal cuts along vehicle routes for further optimization. Due to the possibilities of significant variation in geographic distribution of customer placement and the resource demands by each, these procedures are essential to obtain the clusters of customers that can result in an optimal solution R.

In order to define the process of phase 1 in the algorithmic framework, we first explain the SpatialPartition heuristic in Section 6.3.2 that incorporates a parameterized priority weighting scheme associated with the partitioning. Section 6.3.3 will explicitly illustrate this procedure and describe the evolution of the priority weighting scheme as described

Algorithm 4 Two-Phase Spatial Partitioning Meta-Heuristic Algorithm

Input: C : Set of customers required to be serve by a vehicle route;
D : Set of depots used as the origin of vehicle routes;
α : Maximum tour duration allowed for any vehicle route;
β : Maximum vehicle capacity;
λ : Continuous weight reduction factor
Output: R : Set of vehicle routes

Initialize empty set of routes $R \leftarrow \emptyset$
Initialize optimization priority weight $\omega \leftarrow 1$
Initialize weight reduction factor variable $\lambda \in [0, 0.5]$
$\{R_{T_1}, R_{T_2}\} \leftarrow$ SpatialPartition(C, D, ω, α, β)
$R \leftarrow \{R_{T_1}, R_{T_2}\}$
while $\tau(R_i) > \alpha, R_i \forall R$ **do**
    $R_x \leftarrow \arg\max_{R_x \in R} \tau(R_x)$
    $\eta \leftarrow \{R_x - v_{\phi(0)}\}$
    $\{R_{T_1}, R_{T_2}\} \leftarrow$ SpatialPartition(η, D, ω, α, β)
    $R \leftarrow \{R \setminus \{R_x\}\} \bigcup \{R_{T_1}, R_{T_2}\}$
end while
PruneRoutes(R, D, ω, α, β)
while $\tau(R_i) > \alpha$ **or** $\psi(R_i) > \beta, R_i \forall R$ **do**
    if $\omega = 1$ **and** $\tau(R_i) \leq \alpha, R_i \forall R$ **then**
        $\omega \leftarrow \omega\lambda$
    end if
    $R_x \leftarrow \arg\max_{R_x \in R} \psi(R_x)$ ▷ Route with largest resource demand
    if $\psi(R_x) \leq \beta$ **then**
        $R_x \leftarrow \arg\max_{R_x \in R} \tau(R_x)$
    end if
    $\eta \leftarrow \{R_x - v_{\phi(0)}\}$
    $\{R_{T_1}, R_{T_2}\} \leftarrow$ SpatialPartition(η, D, ω, α, β)
    $R \leftarrow \{R \setminus \{R_x\}\} \bigcup \{R_{T_1}, R_{T_2}\}$
end while
for $i \leftarrow 1$ to $|R|$ **do**
    Solve TSP(R_i)
end for

in the framework. Additionally this section will describe the identification of optimal cuts among unfeasible vehicle routes using the PruneRoutes procedure that follows the state of optimization where the priority will shift to another tier during the continuous partitioning.

6.3.2. Vertex Clustering through Spatial Partitioning

Algorithm 5 formally presents the heuristic SpatialPartition to partition a provided set of customers η and return two temporary vehicle routes (i.e., R_{T_1}, R_{T_2}) that serve

Algorithm 5 Spatial Partition

Input: η : Set of customers to spatially partition into two routes;
D : Set of depots used as the origin of vehicle routes;
ω: Optimization priority weight;
α : Maximum tour duration allowed for any vehicle route;
β : Maximum vehicle capacity

Output: $\{R_{T_1}, R_{T_2}\}$: Two new partitioned vehicle routes

Initialize empty routes $R_{T_1} \leftarrow R_{T_2} \leftarrow \emptyset$
$\{v_a, v_b\} \leftarrow \arg\max_{a,b\in\eta}\{\tau_{ab}\}$
$v_{\phi(1)} \leftarrow v_a$ for $v_{\phi(1)} \in R_{T_1}$
$v_{\phi(1)} \leftarrow v_b$ for $v_{\phi(1)} \in R_{T_2}$
$\eta \leftarrow \eta \setminus \{v_a, v_b\}$
while $|\eta| > 0$ **do**
  $R_x = \max\Big\{\chi(R_{T_1}), \chi(R_{T_2})\Big\}$
  $k = |R_x|$
  $v_i \leftarrow \arg\min_{i\in\eta}\{v_{\phi(1)}, v_{\phi(k)} \in R_x \mid \min\{\tau_{v_i v_{\phi(1)}}, \tau_{v_{\phi(k)} v_i}\}\}$
  if $\tau_{v_i v_{\phi(1)}} \leq \tau_{v_{\phi(k)} v_i}$ **then**
    $R_x \leftarrow \langle v_i \rangle + R_x$
  else
    $R_x \leftarrow R_x + \langle v_i \rangle$
  end if
  $\eta \leftarrow \eta \setminus v_i$
end while
$k = |R_{T_1}|$
$d_i \leftarrow \arg\min_{i\in\eta}\{v_{\phi(1)}, v_{\phi(k)} \in R_{T_1} \mid \min\{\tau_{v_d v_{\phi(1)}}, \tau_{v_{\phi(k)} v_d}\}\}$
if $\tau_{d_i v_{\phi(1)}} \leq \tau_{v_{\phi(k)} d_i}$ **then**
  $R_{T_1} \leftarrow \langle d_i \rangle + R_{T_1}$
else
  $R_{T_1} \leftarrow R_{T_1} + \langle d_i \rangle$
end if
$k = |R_{T_2}|$
$d_i \leftarrow \arg\min_{i\in\eta}\{v_{\phi(1)}, v_{\phi(k)} \in R_{T_2} \mid \min\{\tau_{v_d v_{\phi(1)}}, \tau_{v_{\phi(k)} v_d}\}\}$
if $\tau_{d_i v_{\phi(1)}} \leq \tau_{v_{\phi(k)} d_i}$ **then**
  $R_{T_2} \leftarrow \langle d_i \rangle + R_{T_2}$
else
  $R_{T_2} \leftarrow R_{T_2} + \langle d_i \rangle$
end if

all customers in η. Although the objective of the first phase of the problem decomposition is to construct clusters of customers that are optimally configured into vehicle routes in the following phase, we initially represent each of the clusters by naively constructing a vehicle route from its members. This is necessary for determining the clusters feasibility of constructing a vehicle route and measuring its progress towards the optimization objectives.

Input Parameters	
$C \subset V(G_C)$	Set of customers required to be visited
$D \subset V(G_C)$	Set of depots that the routes must begin at
α	Maximum tour duration (i.e., travel time) allowed for any route
β	Maximum capacity for any vehicle in R
ω	Weight of optimization priority constraint; value initially assigned as 1
λ	Continuous weight reduction factor $[0, 0.5]$
$\eta \subseteq C$	Subset of customers to be partition into two new tours, R_{T_1}, R_{T_2}
Output Parameters	
R	Resulting set of vehicle routes
R_{T_1}, R_{T_2}	Temporary routes

Table 6.2: Parameter variable notation used in Algorithms 4, 5, and 6

The goal of this procedure is to assign all customers in the set, η, to construct two uniform clusters. Specifically, the uniformity will be measured by the temporary route constructed for each cluster (i.e., R_{T_1}, R_{T_2}) independently. Ideally this measure of uniformity should describe the equivalence of the vehicle properties so that R_{T_1} is uniform to R_{T_2} when $\tau(R_{T_1}) = \tau(R_{T_2})$ and $\psi(R_{T_1}) = \psi(R_{T_2})$. However unless the resources demanded by each customer or the distance between all pairs of customers can be defined by a known functional relationship, maintaining this type of uniformity is not feasible. Furthermore, the formulation of this VRP variant under high consequence constraints calls for a priority on certain properties of the route, such as the constrain on time in emergency response. As a result of these circumstances Algorithm 5 attempts to build two routes that are uniform with respect to their individual value obtained from the scoring function $\chi(R_i)$. We define this function in Equation 75 to associate the route with the allocation of priority weighted resources.

$$\chi(R_i) = \left(1 - \frac{\tau(R_i)}{\alpha}\right)(\omega) + \left(1 - \frac{\psi(R_i)}{\beta}\right)(1 - \omega) \tag{75}$$

This allows the time and capacity constraints to be prioritized by the associated weight ω. Additionally it will penalize routes that have limited resources still available or even surpassed the limitations outline by the optimization constraints. This situation occurs when the time or capacity required to serve the customers in the vehicle route exceeds the vehicle constraints, specifically when $\tau(R_i) > \alpha$ or $\psi(R_i) > \beta$ respectively. The priority weight variable ω is a parameter of this heuristic and its value is managed by the algorithm framework (Algorithm 4), assigning ω the value associated with the current priority tier at each phase of the mutli-stage termination criterion. The evolution of ω and the determination of its value is explained in Section 6.3.3 for the illustration of Algorithm 4.

Utilizing Equation 75 as a measurement in obtaining uniformity, we extend the concepts in UPAS previously discussed to partition η. As described in Algorithm 5, we initially initialize two temporary routes by assigning them each the two customers v_a, v_b that have the largest distance between them, followed by removing these customers from η (lines 1-5). Note that this meta-heuristic SpatialPartition must return two routes that contain all customers in η and as a result a feasible solution might not exist. As a consequence we allow the temporary routes at each stage of this algorithm to be infeasible. Following this initialization we iteratively select R_x of the temporary routes where $R_x = \max\left\{\chi(R_{T_1}), \chi(R_{T_2})\right\}$ at each iteration and find the customer $v_i \in \eta$ that minimizes the distance to either end of R_x and insert it into the selected route (lines 6-16). After the insertion, v_i is removed from η and this process repeats until $|\eta| = 0$ indicating that all customers are partitioned into two clusters and naively formed into the two temporary routes R_{T_1}, R_{T_2}. The final steps of this heuristic is to simply add to each temporary route the depot $d \in D$ that minimizes the distance to either end of the routes so that they confirm to the formal route structure so that global optimization and feasibility can again be determined (lines 17-29).

6.3.3. Evolving Resource Prioritization from Infeasible Routes

Using the SpatialPartition heuristic we now briefly describe the first phase of the general framework as illustrated in Figure 6.2. Initially we define the priority weight ω in the first optimization tier such that we can continuously partition the set of customers C as a set of routes $R_i \in R$ until the priority condition ($\tau(R_i) > \alpha$) for all routes are met. With this intention we assign $\omega \to 1$, effectively prioritizing the vehicle routes with respect to time

Algorithm 6 Prune Routes

Input: R : Set of vehicle routes;
D : Set of depots used as the origin of vehicle routes;
ω: Optimization priority weight;
α : Maximum tour duration allowed for any vehicle route;
β : Maximum vehicle capacity
Output: R : Set of vehicle routes

$\eta \leftarrow \emptyset$ ▷ Empty set of customers pruned from routes
$S \leftarrow \emptyset$ ▷ Empty list of pruned route indexes
$A \leftarrow \emptyset$ ▷ Empty set of new pruned routes
for $i \leftarrow 0$ to $|R| - 1$ **do**
  if $\psi(R_i) > \beta$ **then**
    $S \leftarrow S \bigcup R_i$
    $d \leftarrow v_{\phi(0)} \in R_i$
    $R_i = \{\langle v_{\phi(t)} \rangle_{t \in R_i} \mid \tau_{dv_{\phi(t)}} \geq \tau_{dv_{\phi(t+1)}},\ t = 0, 1, ..., |R_i| - 1\}$ ▷ Sort R_i
    $k \leftarrow 0$ ▷ Capacity required in route
    for $j \leftarrow 0$ to $|R_i| - 1$ **do**
      $v \leftarrow v_{\phi(j)} \in R_i$
      if $k + \psi(v) \leq \beta$ **then**
        $k \leftarrow k + \psi(v)$
      else
        $R'_i = \langle v_{\phi(t)} \rangle_{t=0}^{j-1}$ ▷ R'_i represents the pruned route of R_i
        $l = |R'_i| - 1$
        $d_x \leftarrow \arg\min_{x \in D}\{v_{\phi(0)}, v_{\phi(l)} \in R'_i \mid \min\{\tau_{d_x v_{\phi(0)}}, \tau_{v_{\phi(l)} d_x}\}\}$
        $R'_i = \langle d_x \rangle + R'_i$ ▷ Add closest depot to either end of route
        $A \leftarrow A \bigcup R'_i$
        $\eta \leftarrow \eta \bigcup \left\{ \langle v_{\phi(t)} \rangle_{t=j}^{|R_i|-2} \right\}$
        Break ▷ Continue to the next route over capacity
      end if
    end for
  end if
end for
$R \leftarrow R \setminus S$
$\{R_{T_1}, R_{T_2}\} \leftarrow \textsc{SpatialPartition}(\eta, D, \omega, \alpha, \beta)$
$R \leftarrow \{R\} \bigcup \{A\} \bigcup \{R_{T_1}, R_{T_2}\}$

when applying the scoring function χ previously described in Equation 75. Consequently this allows a temporary relaxation of the capacity constraints when building the uniform clusters (i.e., temporary routes) in Algorithm 5. To this end we initialize the empty solution set of all vehicle routes R, by appending the temporary routes $\{R_{T_1}, R_{T_2}\}$ returned by the SpatialPartition procedure. Observe that for each instance of the procedure the values of the first and third parameters (η, ω) are dynamic, while the remaining parameters (D, α, β)

represent static values. In particular when initializing R (Algorithm 4 line 5), $\omega \rightarrow 1$ reflects the value of the first tier in priority weighting, and $\eta \rightarrow C$ such that all customers in the road network are to be initially partitioned.

Once R is initialized, Algorithm 4 will continuously partition any route $R_i \in R$ exceeding the time constraint $(\tau(R_i) > \alpha)$ until all routes meet the constraint (70) defining the initial stage of the termination criteria (lines 6-11). At each iteration R_x is selected as the route that maximally exceeds the time constraint. The customers served by R_x are removed from R (excluding the depot) and assigned to the parameter η. Before the start of the next iteration, R_x is removed from the set R and replaced by the two routes obtained from performing the SpatialPartition procedure on the customers in η.

In this first priority tier where $\omega = 1$, the scoring function (Equation 75) assigns the sole priority on improving the component of time constraints accordingly. This is specifically demonstrated in Algorithm 5 (line 7) when constructing two new routes. As a consequence the routes are continually partitioned until the initial termination criteria in Algorithm 4 (line 6) has been met. At this point every route will have met the priority criteria constraint, enforcing the maximum tour duration allowed. In other words, all routes are now guaranteed to serve their respective customers within the allowed time. But even so, the capacity criteria has not been addressed. Hence any number of these routes could be exceeding their maximum capacity. For this reason, the optimization priority weight ω then evolves into its second priority tier for the remainder of phase 1 of the algorithm framework.

This second tier is represented by reassigning $\omega \leftarrow \omega\lambda$ where $\lambda \in [0, 0.5]$. The value of λ defines the factor by which to reduce ω. The resulting value represents this second tier for ω to shift the optimization to the capacity criteria. The degree by which the priority changes corresponds directly to parameter value λ. Moreover, the factor by which the time criteria is reduced by will equal the proportion of priority the capacity criteria will receive. When $\lambda = 0$ only capacity in vehicle routes are considered. Conversely if $\lambda = 0.5$, equal priority to both time and capacity constraints are assigned. This updated value of ω will therefore adjust the partitioning procedure by including the capacity utilization when constructing the routes.

We assume λ is provided as a parameter to Algorithm 4, and obtained by experimentation of this value based on the specific network defined for optimization.

With the shift of ω into the second tier, we repeat the continuous partitioning steps but instead with a different termination criteria. Namely, satisfying all of the problem constraints will trigger the termination of the partitioning. Before we do this however, the current set of routes are strategically reorganize to improve the optimization performance of the partitioning. The PruneRoutes procedure in Algorithm 6 performs this process by identifying optimal customers to cut from routes that exceed their capacity.

To identify theses optimal cuts we first select all routes that currently exceed their allowed capacity. Since all routes are currently about to be completed with the maximum time allowed, those that exceed their capacity are all that is required to partition further. Furthermore, it is observed that no removal of any customer from these routes will result in a route that exceeds the maximum tour duration. In other words, since there currently exists a permutation (i.e., path) of customers for every route that can be performed within the time allowed, any subset of customers can be similarly visited in the same time or less.

The number of customers pruned from each route is determined by the proportion the route exceeds the available capacity. Customers are continually pruned from the route until the route's remaining customers' demand can be fully served by the available capacity. As described in the algorithm, these remaining customers are strategically picked such that they are near each other. The customers are removed in an increasing order of their distance to the depot. Once the pruning process has completed, all remaining routes will satisfy both the time and capacity criteria. Those customer removed will define the final set of customers to construct into routes. As previously mentioned, the continuous partitioning on this set of routes is adjusted to prioritize the capacity criteria. After feasible routes have been formed from the pruned customers, Algorithm 4 begins a post optimization procedure as define by its second phase.

6.3.4. Route Construction

Following the first phase, we additionally utilize existing exact and heuristic approaches (see general surveys [12] [11]) to solve the TSP for each cluster in the second phase of Route Construction in the problem's decomposition. Essentially this step simply optimizes the order by which the route visits the customers to improve the resource utilization. This phase of the algorithm is a common post-optimization step in many VRP algorithms. Further this step can be formulated as solving a TSP for each individual cluster and existing research is utilized. Algorithm 4 (lines 25-27) demonstrate this optimization by independently treating each route in R as its own general Open TSP to be solved. Of course, selecting the algorithm to use for this procedure will depend on the specific problem domain and size. For example, if the average customer only requires a very insignificant amount of vehicle capacity in relation to the total capacity available, each route is likely to include a large number of customers. In this case, a faster approximation algorithm might be the most appropriate procedure to use here. Conversely for longer distance routes each with minimum customers included, accuracy from exact algorithms might be the best suited. Overall there are many acceptable choices for this stage that can be used in satisfying the requirements of the application. For the bio-emergency response domain that has motivated the development this algorithm, simulated annealing [39] in particular demonstrated to be the best fit. At the conclusion of this second phase the algorithm returns R, the optimized set of feasible vehicle routes, solving the optimization problem formulated in Section 6.2.

6.3.5. An Illustrative Example

The application of Algorithm 4 is exemplified in this section using the simplistic network illustrated in Figure 6.3. This network consists of 14 customers and a single depot. From this network a set of routes are construct such that the fleet size is minimized. Figures 6.4 and 6.5 illustrate the steps and procedures associated with constructing the solution for this network.

Algorithm 4 begins by first selecting the two customers who are the furthest apart. It is assumed that the cost to travel between any two customers is calculated by their euclidean

Figure 6.3. Illustration of a simple network with labeled customers for identification in Figures 6.4, 6.5, and 6.6

distance as mentioned in the previous section. The yellow circles in Figure 6.4(b) identify the customers furthest from each other. Each of these two customers will be included in one of the two routes that result from this first level of partitioning. Using each of the selected customers as an endpoint for a route, the next closest customer to each are assigned to their respective partition. Figure 6.4(c) accordingly demonstrates the current customer assignment following this step. Both sets of the assigned customers each represent a route defined by the order by which these customers are connected. Since the euclidean distance provides an asynchronous travel cost for each connection, the route can be assumed to begin at either of the two the ends. The remaining construction of these two tours is continued from this assignment of the next closest insertion heuristic at either end. This process is formally defined by the aforementioned SpatialPartition procedure. Accordingly, both routes in are each scored using the Equation 75 following every customer assignment. As shown in Figures 6.5(d) and (e), the route with the highest score is assigned the next customer. Until all customers have been assigned to a route where the complete set of routes do not exceed the maximum tour duration, routes with the shortest travel time are assigned the highest scores. Once all customers are assigned to one of these two tours the depot is attached to the closest endpoint of each route. This is illustrated in Figures 6.5(f) and (g) and signifies the

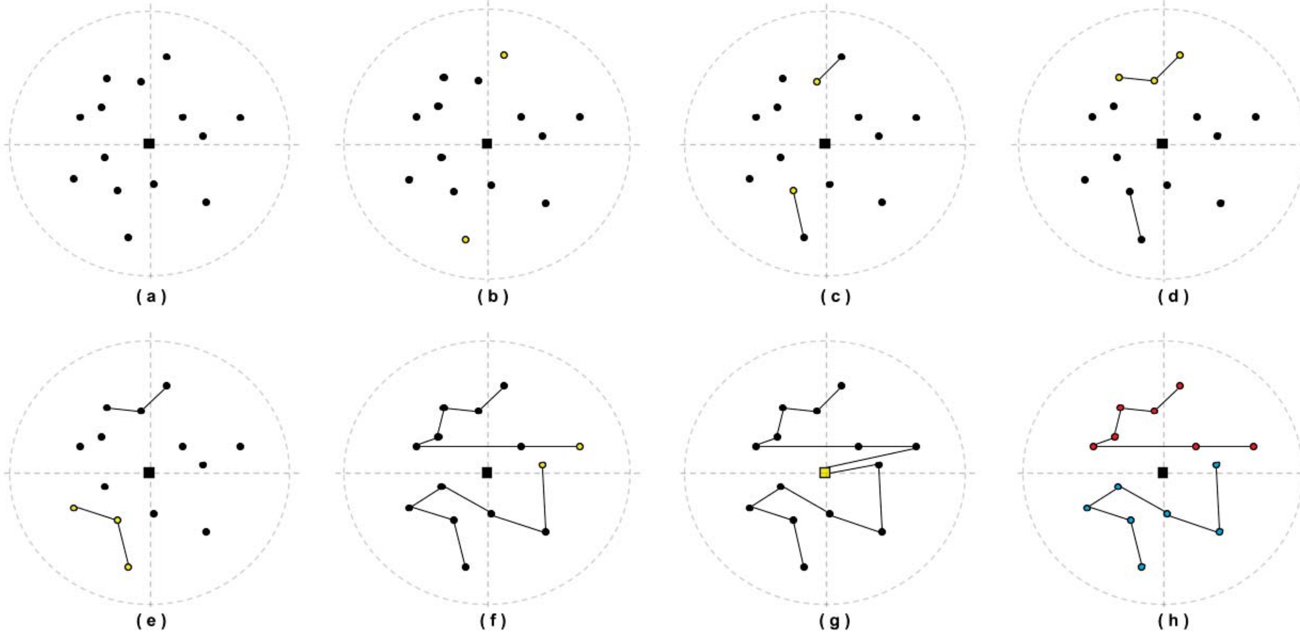


FIGURE 6.4. Illustration (1 of 2) of the SPATIALPARTITION procedure as called in Algorithm 4

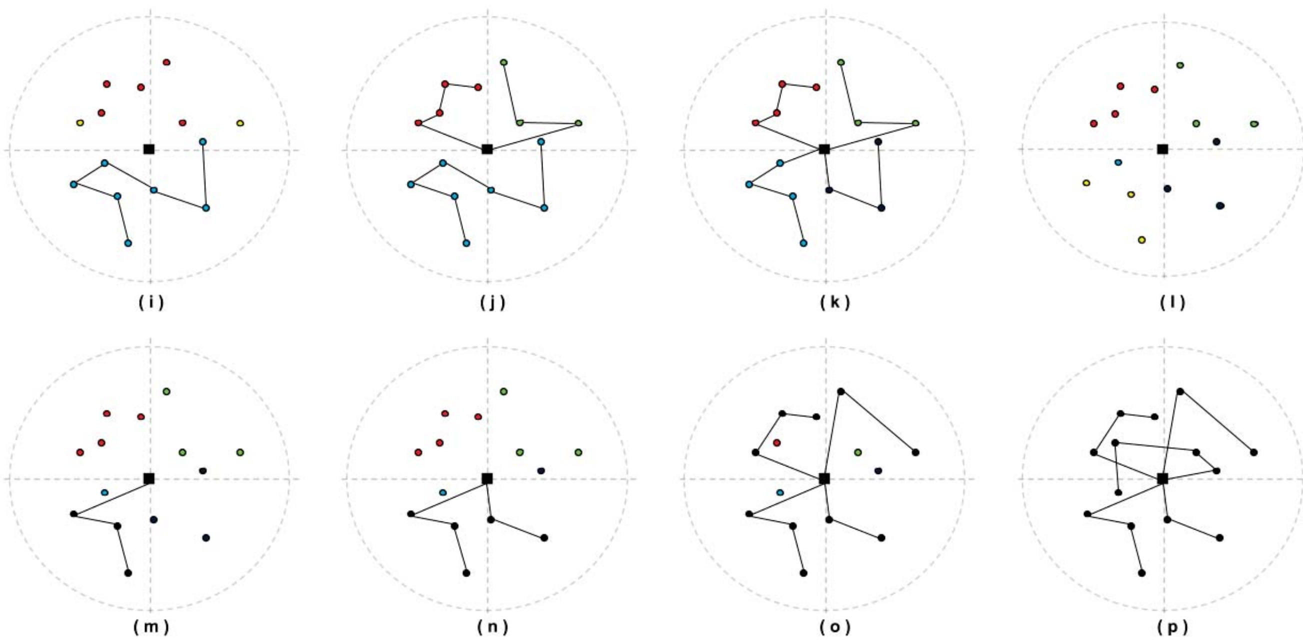


FIGURE 6.5. Illustration (2 of 2) of the continued partitioning from Figure 6.4 in (i),(j), and (k). (l) through (p) illustrate the PRUNEROUTES procedure as called in Algorithm 4

completion of the SPATIALPARTITION procedure for the first iteration. The children of the root node in the tree illustrated in Figure 6.6 represents these two routes accordingly. This procedure is independently repeated for any route that exceeds the maximum duration. It is assumed in Figure 6.4(h) that both routes exceed the maximum time allowed in this example. As a result, this process is repeated for one of the routes and then again for the remaining route, as illustrated by Figures 6.5(i)(j) and Figure 6.5(k) respectively. The resulting set of the four routes constructed at this point are defined by the children nodes at the second level of the tree in Figure 6.6. For the purpose of this example, it is assumed that all four of these routes no longer exceed the maximum tour duration allowed. However, it is assumed that all of the routes do exceed their vehicle capacity. Figure 6.5(k) illustrates each of the four current routes of which are identified accordingly with a single color for each one.

To demonstrate the PruneRoutes procedure in Algorithm 4, Figure 6.5(l) depicts the maximum number of customers in yellow that can be selected from the blue route (i.e., R_1) without exceeding the vehicle capacity. These customers are selected in order according to their distance from the depot. As a result, the remaining customers are also the closest to the depot. Figure 6.5(m) shows the resulting pruned route where the remaining blue circle is the only removed customer. The pruning of the remaining routes are additionally illustrated in Figures 6.5(n) and (o). Once the set of customers, $\{4, 6, 10, 11\}$, are removed from this pruning procedure, the resulting routes will all be feasible. The set of the removed customers will then continue to be partitioning; basically restarting the entire process using this set alone. The only change is that the optimization priority on time will be reduced such that the scoring function will instead now prioritize the optimization of capacity usage. However, for the purpose of this example, it is assumed that the set of customers pruned from the routes can be constructed into a single feasible route. This result is demonstrated in Figure 6.5(p) and the routes are defined accordingly by the leaf nodes in Figure 6.6.

In Chapter 7 we illustrate the direct application of this two-phase algorithm in the context of bio-emergency response. Namely, a case study is constructed from a large region in the state of Texas to exemplify the performance and outcomes of the algorithm.

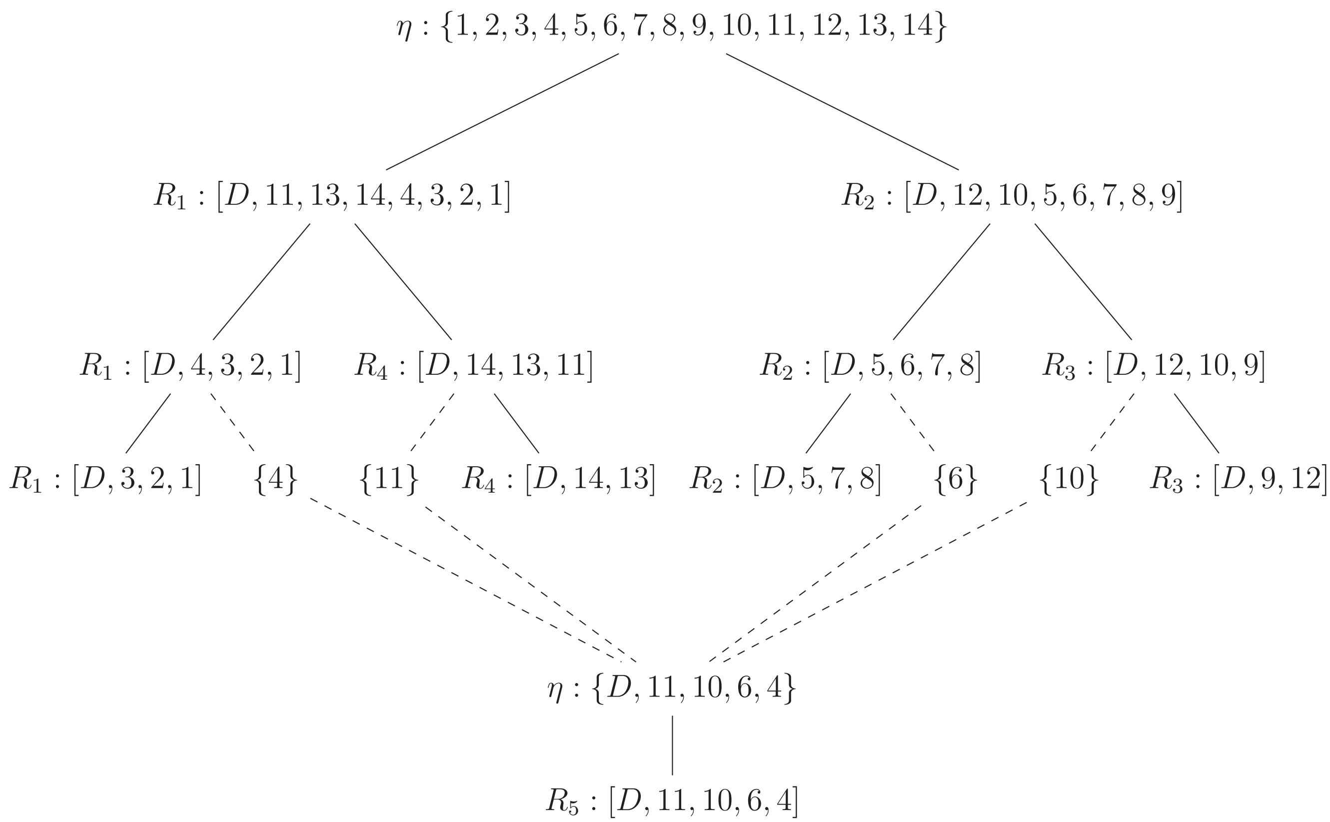


FIGURE 6.6. Tree representation of the route partitioning and pruning performed in Figures 6.4nd 6.5 The leaf nodes denote the five routes as the solution. The dashed lines denote the pruning of customers from a route. The numbers in each route correspond to the customers as labeled in Figure 6.3

CHAPTER 7

APPLICATION TO RESPONSE PLANNING

To demonstrate the applicability of the Algorithm 4 (Two-Phase Partitioning Algorithm) previously defined in Chapter 6, an illustrative large-scale case study is created in Section 7.1. The first phase of a bio-emeregency response plan (Resource Transportation & Distribution) as depicted in Figure 2.1 is the focus of the study. Multiple variants of the priority strategy for Algorithm 4 are described in Section 7.2. Two of these strategies in particular are modeled after the analysis of the partition heuristics solving the OVRP-UT and the OVRP-UC defined in the previous chapters. The remaining strategies are constructed specifically for solving the formal optimization problem defined in Section 6.2. The performance of these optimization procedures are presented in Section 7.3. Section 7.4 describes the integration of these methods into a computation framework to aid public health officials in bio-emergency response planning.

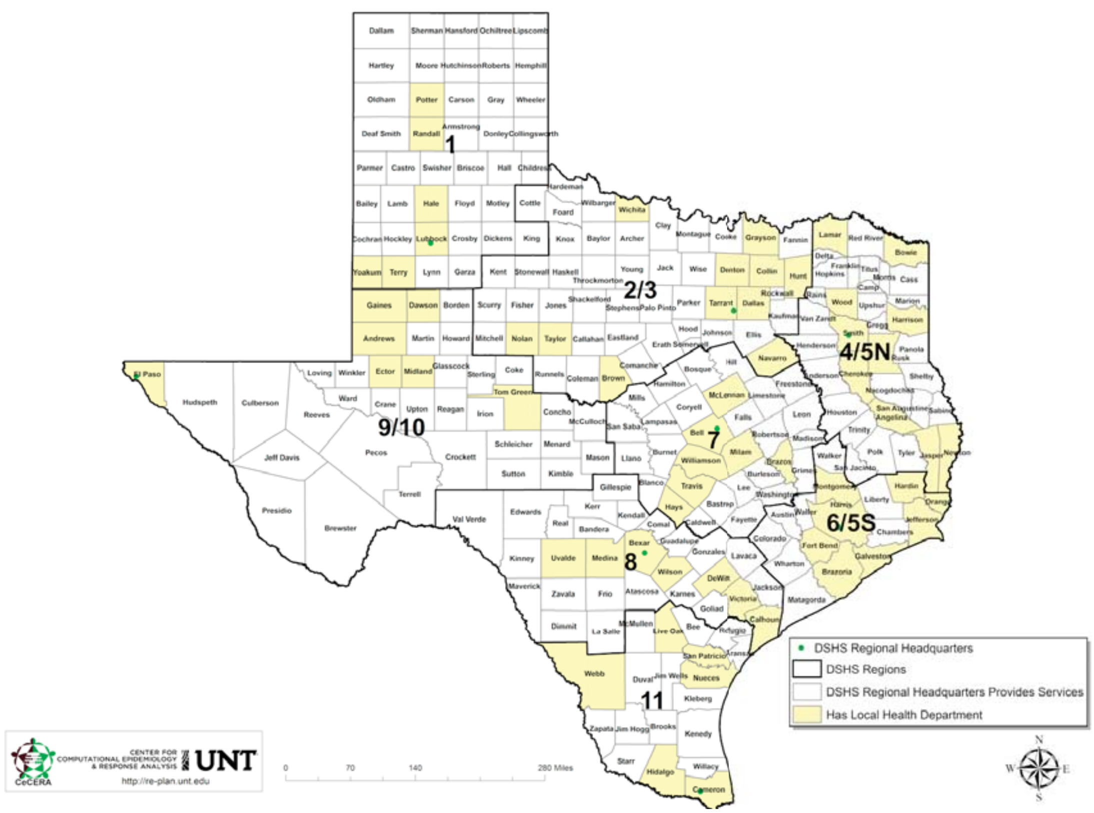


FIGURE 7.1. Texas Department of State Health Services (DSHS): Health Service Regions

7.1. Regional Case Study

To analyze the performance Algorithm 4, a bio-emergency scenario is constructed. Moreover, this provides verification of the correctness of the computational framework presented in Chapter 6 as a consequence of the experimental evaluation.

In the event of a bio-emergency (as described previously in Section 2.1), the federal government will ship Medical Countermeasures (MCMs) from the Strategic National Stockpile (SNS) to the local state government's Receiving, Storing and Staging (RSS) warehouses. These MCMs must then be distributed to the affected region's strategically placed Point of Dispensing (POD) facilities to treat the population. Short time frames (e.g., 24 or 36 hours) for dispensing the MCMs are mandated by the federal government to reduce the severity of the emergency. We illustrate these mitigation efforts by constructing a scenario a large geographic region in Texas containing a significantly sized population. Figure 7.1[1] demonstrates geographic boundaries defining the health service regions by the Texas Department of State Health Services (DSHS). The health service region 2/3 is observed for this case-study. A response plan was created for this region accordingly in determining the quantity and placement of the PODs by using existing response planning software and methodology as presented in [51] and [57]. Figure 7.2 illustrates the placement of 341 PODs and 2 RSS warehouses. The resource demand (i.e., supplies required) is defined by the total population count assigned/affiliated with each individual POD. Figure 7.3 illustrates the demand for each POD relative to the overall distribution by the size of each red circle. The red circles denote the location of a POD while the size of the circles represent the total amount of resources that are required by the POD. The larger sizes indicate a larger demand accordingly.

As previously described in Figure 2.1, the resources required by each of these PODs will be delivered by a set of vehicle routes that all start from the two RSS warehouses. Furthermore, the following assumptions complete the configurations required in constructing the scenario before executing the experiments:

(i) All resources must be delivered form the RSS warehouses to all POD locations and

[1]Image courtesy of UNT Center for Computational Epidemiology & Response Analysis (CeCERA)

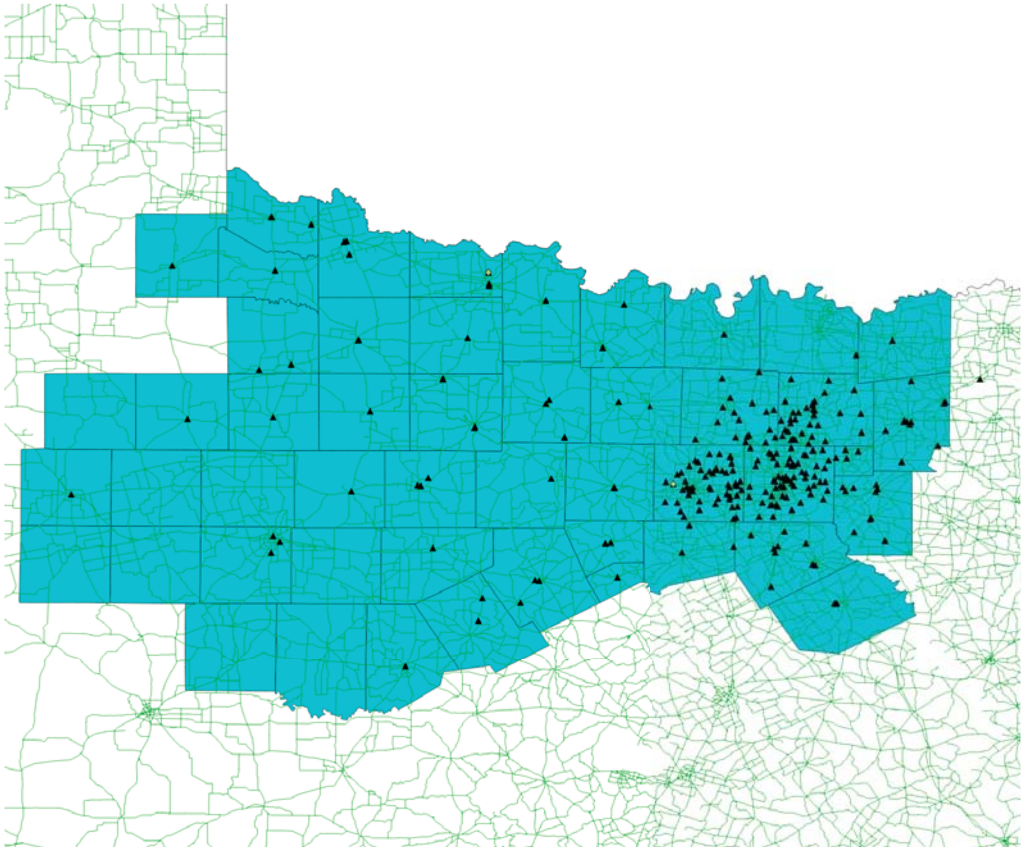


FIGURE 7.2. Texas DSHS Region 2/3 (Case Study): Bio-emergency response plan

unloaded within 12 hours

(ii) For every POD visited in each tour, the process of unloading the resources to deliver will take 30 minutes to complete

(iii) Each POD must be delivered enough supplies to fulfill the resources required to serve its specified population demand

(iv) Population demand (i.e., required resources for each individual POD) have been determined by assigning the sourcing population to their respective closest POD location to calculate the total population demand

(v) Each delivery vehicle must not exceed its max capacity allowed

(vi) Vehicles are assumed to be large distribution trucks with trailers that can sufficiently carry enough supples to fulfill a POD population demand of up to 211,000 people, calculated by the following:

- 22 regular sized pallets per vehicle

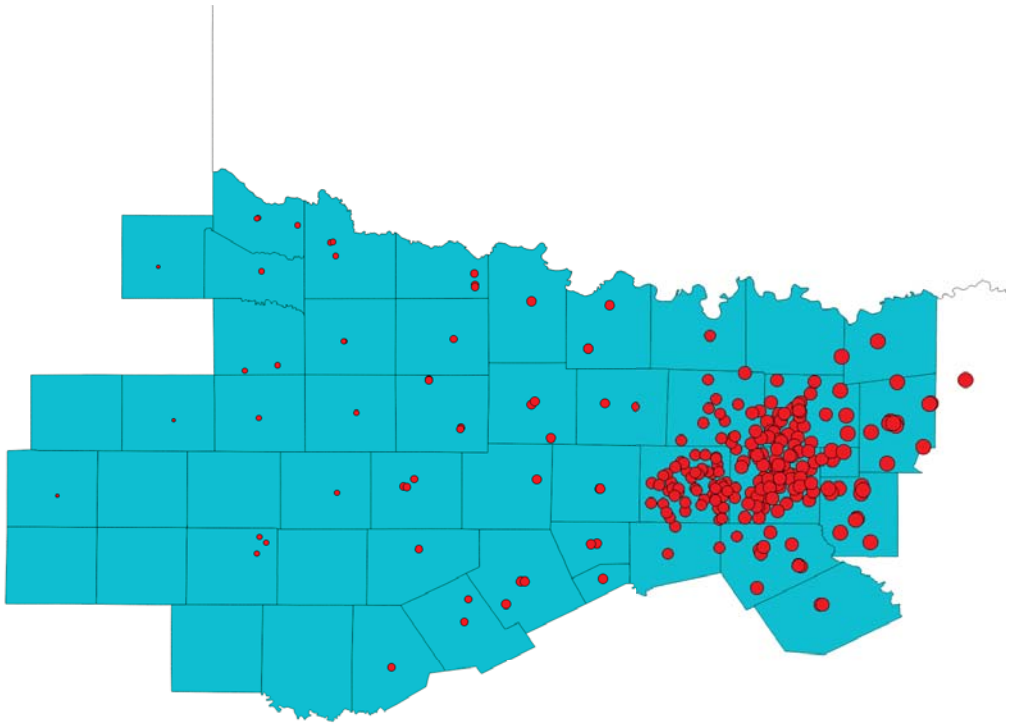


FIGURE 7.3. Texas DSHS Region 2/3 (Case Study): POD demand by location

- 9,600 bottles per pallet
- 1 bottle per person for sufficient treatment

The following section defines multiple variants of Algorithm 4. We apply an implementation of these optimization strategies to the illustrated bio-emergency scenario and examine their respective performance in Section 7.3.

7.2. Optimizing Vehicle Routes with Two-Phase Partitioning

This section describes multiple optimization priority strategies associated with Algorithm 4. The implementation of each strategy will each generate a set routes as the solution to the optimization problem in Section 6.2 for the illustrated bio-emergency scenario in region 2/3.

Each strategy places a different a optimization priority on the available time and vehicle capacity. As a result, each provides a different perspective on the performance of utilizing resources while all attempting to achieve the identical goal of minimizing the fleet size. These strategies are each denoted and described as the following:

(i) Time Only Optimization (TOO): This strategy attempts to minimize the number of required vehicles by optimizing only the time constraint while disregarding the capacity constraint. In other words, it assumes that each vehicle has an unlimited capacity. Thus the sole priority during the recursive partitioning is placed on optimizing the required time for each route to complete it's respective tour.

(ii) Capacity Only Optimization (COO): This strategy is similar to the TOO strategy but instead places the sole optimization priority on optimizing the capacity utilization for each route. It assumes that routes are not restricted by any maximum tour duration. Instead only the capacity constraints are considered during the recursive partitioning.

(iii) Time Priority Optimization (TPO): The goal of this strategy is to optimize on the time constraint, while still considering the capacity constraints. Similar to the TOO strategy, the sole optimization priority initially is the utilization of time allowed for each route complete its tour. In the contrary to the TOO strategy however, once all routes adhere to the time constraints, all routes that still exceed their vehicle capacity are further partitioned with the sole optimization priority on capacity utilization.

(iv) Capacity Priority Optimization (CPO): This strategy is similar to the TPO strategy but instead first places the sole optimization priority on optimizing the capacity utilization for each route. Once all routes adhere to the capacity constraints, the sole optimization priority is shifted to utilization on time until all routes are feasible.

(v) Two Phase Optimization (2P0): This strategy represents Algorithm 4 exactly as described in Algorithm 4. The 2P0 strategy utilizes the TOO strategy and the COO strategy in two sequential phases with an additionally optimization procedure between them. Consistent with the TOO strategy, the first phase assigns the sole optimization priority on optimizing the capacity utilization for each route. The second phase is similarly to the COO in this same manner. Following the first phase, all routes exceeding their capacity are pruned as described in Algorithm 4 before proceeding with the second phase.

The TOO, COO, TPO, and CPO optimization strategies are all implemented in a manner similarly described by Algorithm 4 with slight modifications. The 2PO strategy on the other hand is implemented exactly as stated in the algorithm. Using the constructed bio-emergency scenario in region 2/3, the maximum tour duration allowed, α, and the maximum vehicle capacity, β, are assigned the values 12 (denoting the unit of time in hours) and 211,000 respectively.

Implementation of the TOO strategy follows the algorithm exactly with the exclusion of executing lines 13 through 24, and also assigns $\omega \leftarrow 1$. The COO strategy is implemented similarly as TOO, but instead assigns $\omega \leftarrow 0$, and includes two modifications: (mod. i) change the termination criteria on line 6 to terminate when all of the routes no longer exceed their maximum capacity, and (mod. ii) replace line 7 with the statement on line 17. The (mod. ii) modification returns the route with the largest resource demand instead of the route that completes in the maximum time. As a consequence, both of the TOO and COO strategies do not guarantee a solution that can be classified as feasible according to the formal problem defined in Section 6.2. Nonetheless, these theoretical solutions are beneficial to include in this case-study. Namely, observations of their performance in minimizing the fleet size can provide insights to what a specific problem instance is most constrained by.

The implementation of the TPO strategy similarly follows Algorithm 4 with only the exclusion of the PruneRoutes procedure on line 12. Furthermore, the TPO strategy assigns the priority parameters as $\omega \leftarrow 1$ and $\lambda \leftarrow 0$. Similar to the TOO strategy, the implementation of the CPO strategy excludes the PruneRoutes procedure. This implementation also includes both of the modifications made for the COO strategy, i.e., (mod. i) and (mod. ii). Moreover, the CPO strategy implementation includes the following remaining modifications: (mod. iii) replace the test case ($\omega = 1$ **and** $\tau(R_i) \leq \alpha$) with the test case

($\omega = 0$ **and** $\psi(R_i) \leq \beta$) on line 14, (mod. iv) swap the statements on line 17 and line 19 with each other (i.e., replacing each line with the other), and (mod. v) replace the test case ($\psi(R_i) \leq \beta$) with ($\tau(R_i) \leq \alpha$) on line 18. The parameters for the CPO strategy are assigned as $\omega \leftarrow 0$ and $\lambda \leftarrow 1$. For the final modification (mod. vi), swap the assignment statement ($\omega \leftarrow \omega\lambda$) on line 15 with the statement ($\omega \leftarrow \lambda$) instead. Therefore ω will be assigned the value of λ directly. This prevents the value of λ from being discarded given that $\omega\lambda = (0)\lambda = 0$.

As defined by Algorithm 4, the parameters assigned for the 2PO strategy are $\omega \leftarrow 1$ and $\lambda \leftarrow 0$. The *Input Parameters* section in Table 7.1 reflects this assignment of the algorithm parameters for each of the strategies. The following sections presents the results from applying these strategies based on their implementations as previously described.

7.3. Case Study Results

	Optimization Strategy				
Input Parameters	TOO	COO	TPO	CPO	2PO
α : (Maximum Tour Duration)	12	N/A	12	12	12
β : (Vehicle Capacity)	N/A	211,000	211,000	211,000	211,000
ω : (Priority Constraint)	1	0	1	ε	1
λ : (Priority Reduction Factor)	N/A	N/A	0	1	0
Optimization Performance					
m : (Fleet Size)	32	61	59	58	51
Proportion Exceeding Z	0.000	0.032	0.000	0.000	0.000
Proportion Exceeding Q	0.500	0.000	0.000	0.000	0.000
Route Statistics					
Min. Population Served	19,379	77,245	19,379	33,852	19,379
Avg. Population Served	233,550	122,518	126,671	128,855	146,541
Max. Population Served	502,165	202,207	210,719	207,113	210,572
Min. Elapsed Time	6.81	0.91	1.28	1.68	1.20
Avg. Elapsed Time	8.24	4.66	4.62	4.70	5.25
Max. Elapsed Time	10.84	14.24	11.49	11.28	10.84

Table 7.1: Texas DSHS Region 2/3 (Case Study): Optimization strategy results. The unit of time, α, is represented in hours. Vehicle capacity, β, is represented by the max number people it can pack mitigation resources for.

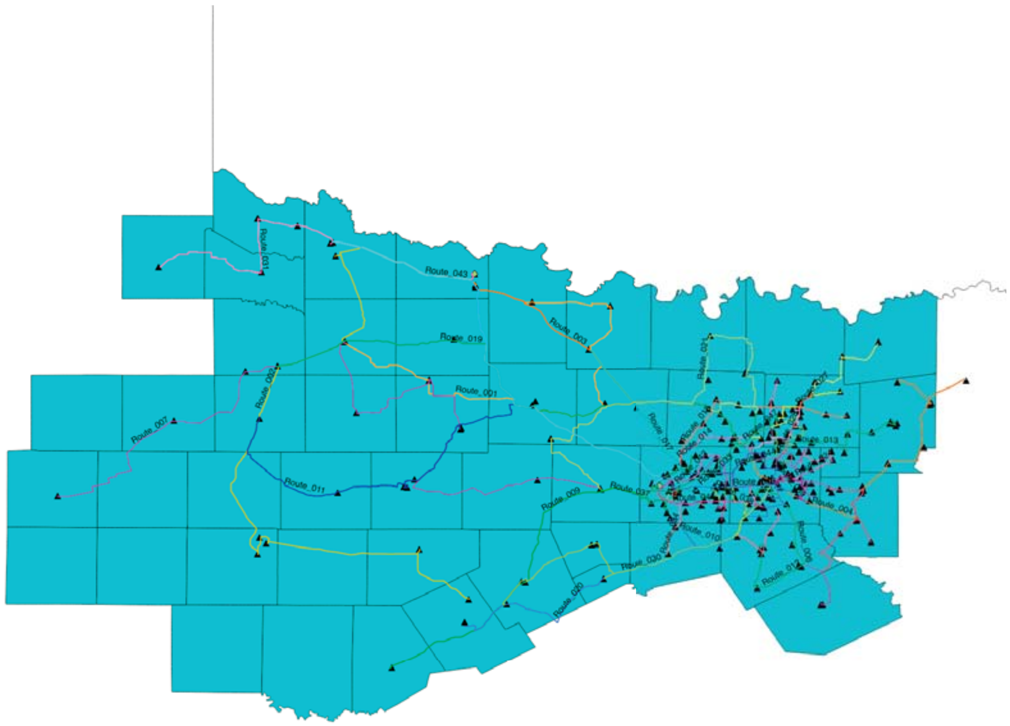


FIGURE 7.4. Texas DSHS Region 2/3 (Case Study): Resulting vehicle routes

The results of the previously defined five priority optimization strategies in their application to the bio-emergency scenario constructed in region 2/3 is shown by Table 7.1. In particular we observe that the 2PO strategy produced the best feasible solution, resulting in a fleet size of 51. Conversely, both TOO and COO produced solutions that are not feasible as some portion of the solution exceeded either the time or capacity constraints.

When reviewing the results from the different implementations, it is beneficial to recall a basic bound of the problem itself. Namely, the Texas DSHS Region 2/3 has a total population that is over 7.4 million people. Assuming that a single vehicle has a capacity capable of carrying enough supplies to serve 211,000 people, then without any consideration for time it is clear that 36 vehicles will be required at minimum.

$$\lceil (7.4 \text{ million people})/(211,000 \text{ people per vehicle capacity}) \rceil = 36 \text{ vehicles} \qquad (76)$$

However, this is under the assumption that the vehicles can perform partial delivery to PODs so that they may fully utilize their capacity. In other words, instead of a vehicle delivering only to PODs it has enough capacity for, underutilized capacity could be used to deliver at least part of the supplies. Therefore these results demonstrate the best feasible solution for this specific scenario will require a fleet size of 51 when using the 2PO optimization strategy. The exact tour for each of these 51 routes in this solution are shown in Table A.1. Furthermore, a visual illustration of this solution is shown in Figure 7.4.

$\boldsymbol{\alpha}$	$\boldsymbol{m}$	Number of pallets in vehicle						
(Max Time Allowed)*	(Fleet Size)	1-3	4-6	7-9	10-12	13-15	16-18	19-22
12	51	1	3	3	9	3	9	23
11	51	1	3	3	9	3	9	23
10	54	1	7	3	8	4	9	22
9	54	4	6	3	5	3	9	24
8	66	10	10	6	5	10	7	18
7	69	10	14	7	5	9	5	19
6	73	15	14	6	5	9	5	19
5	83	23	17	8	1	9	7	18
4	122	51	24	12	12	10	4	9
		Vehicle Frequency						

** The unit of time, α, is represented in hours.*

Table 7.2: Comparing the distribution of demand across a set of vehicle tours as the maximum tour duration is reduced

Further analysis of the performance of the 2PO strategy for the same constructed scenario is provided as it relates the an incremental reduction in the maximum tour duration allowed. Table 7.2 shows the resulting fleet size using the 2PO strategy where α is reduced by one hour at each row of the table. It is clear that the solution significantly worsens for $\alpha < 7$. Furthermore, from the number of pallets used for each vehicle it can be observed that the efficiency of utilizing vehicle capacity is similarly reduced as the α is reduced.

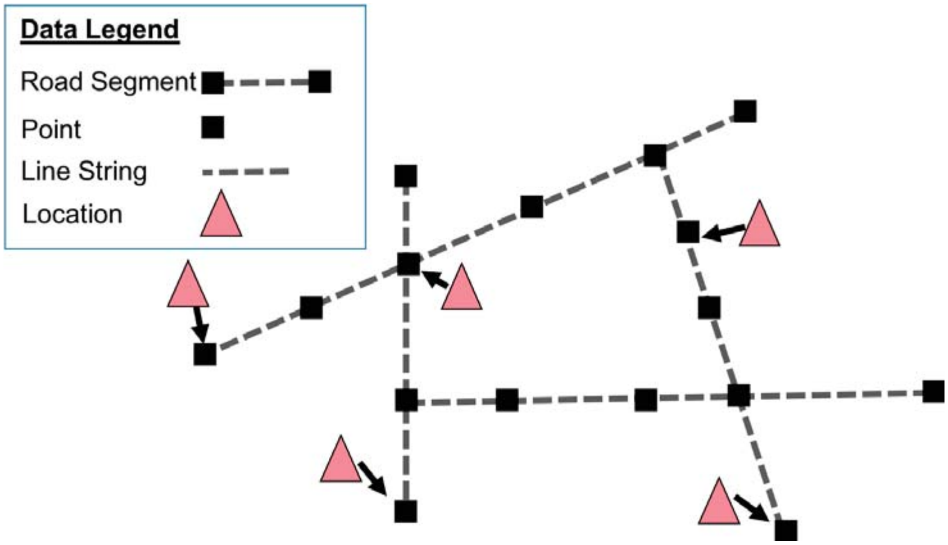


FIGURE 7.5. Abstract diagram showing a data editing step for associating realistic travel distances with point locations. Locations that do not correspond exactly with a road segment end point are moved onto the closest such point.

7.4. Integration of Computational Framework

Due to limitations in the granularity of the population data available, individuals are accounted for by their defined grouping (i.e., population block) with respect to their geographic location provided in the data. For example, data collected by the United States Census Bureau provides populations counts at various geographic scales such as Cencus blocks, block-groups and tracts that define this grouping. Further, these population groupings provide no guaranteed uniformity of population counts or demographic makeup.

Data initialization is included in the framework to process and format the road network data for consistency with the algorithm approach and the graph representation described previously in Section 6.1. The final graph representation described in that chapter assumes a complete graph with a provided set of locations that exist on the road network that are also consistent with the road segment vertices. However in a practical situation this may not be the case. Facilities (i.e., depots and customers) will usually be located at some measure of distance away from the road and not directly on the road. Therefore to enforce this consistency we redefine the geographic location of each facility to its nearest road segment vertex on the network. This is shown in Figure 7.5. Depending on the level

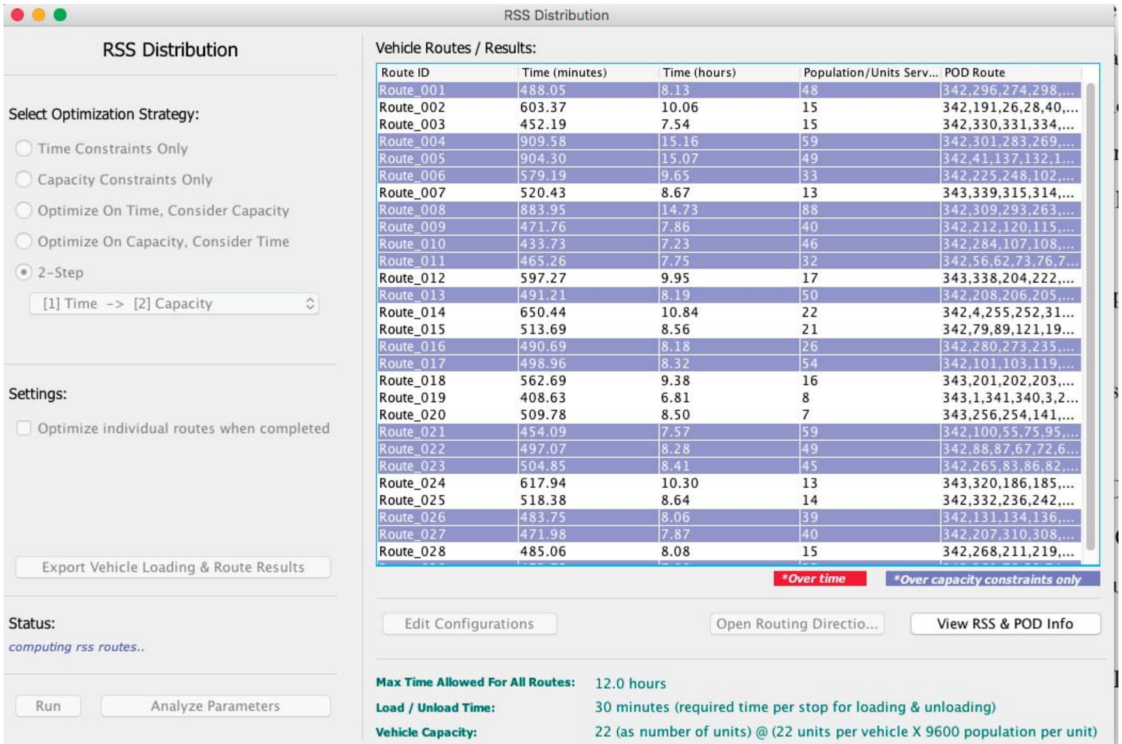


FIGURE 7.6. Software UI simulating the optimization in real-time while solving the problem

of granularity defined by the data of the road network, this process can introduce error into the travel time by not not accounting for this modification of the facilities actual location in the path cost. However the error introduced by this process is likely to be minimal and can easily be ignored. Figures A.1, A.2, and A.3 in the appendix demonstrate how this is presented using a real road network data set such as Open Street Maps.

All of these optimization strategies presented in this chapter have been integrated into a computational framework. These tools will enhance the ability of public health officials and emergency response planners constructing a set of routes in bio-emergencies similar to the case-study presented in this chapter. Figure 7.6 shows the main interface for selecting and executing one of the optimization strategies.

This interface simulates all tours at every step of the partitioning during the execution of the algorithm. As illustrated in Figure 7.6, information describing the routes are updated and displayed in the table view following each step in the simulation. Each row describes a single route in this table accordingly. Furthermore, the background color of each row indi-

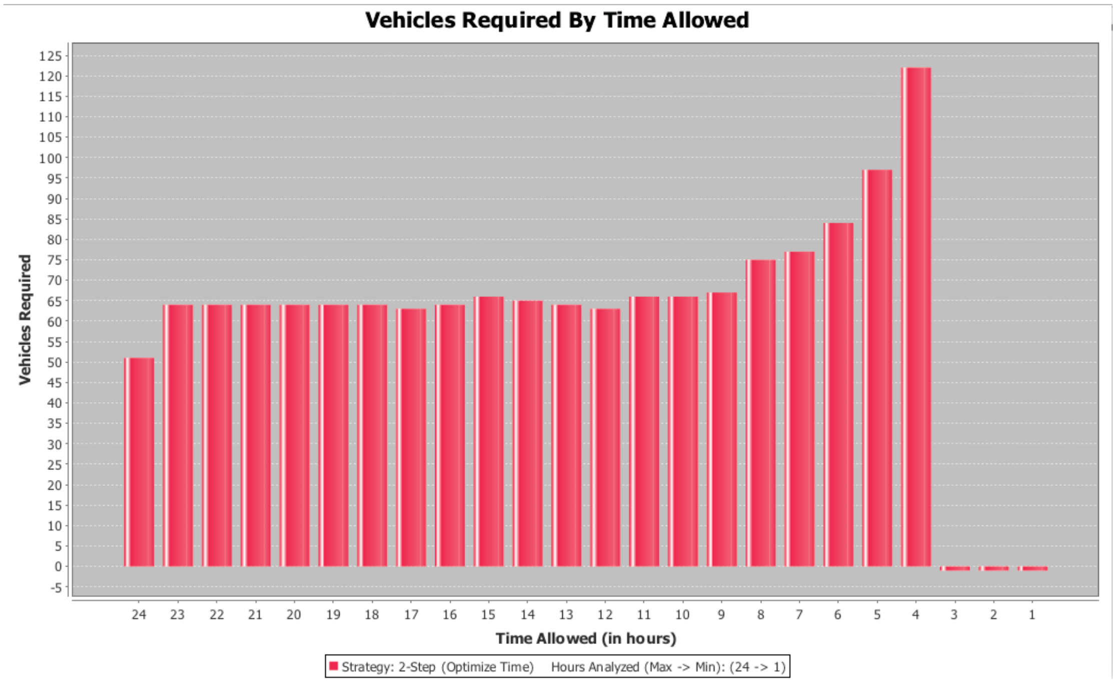
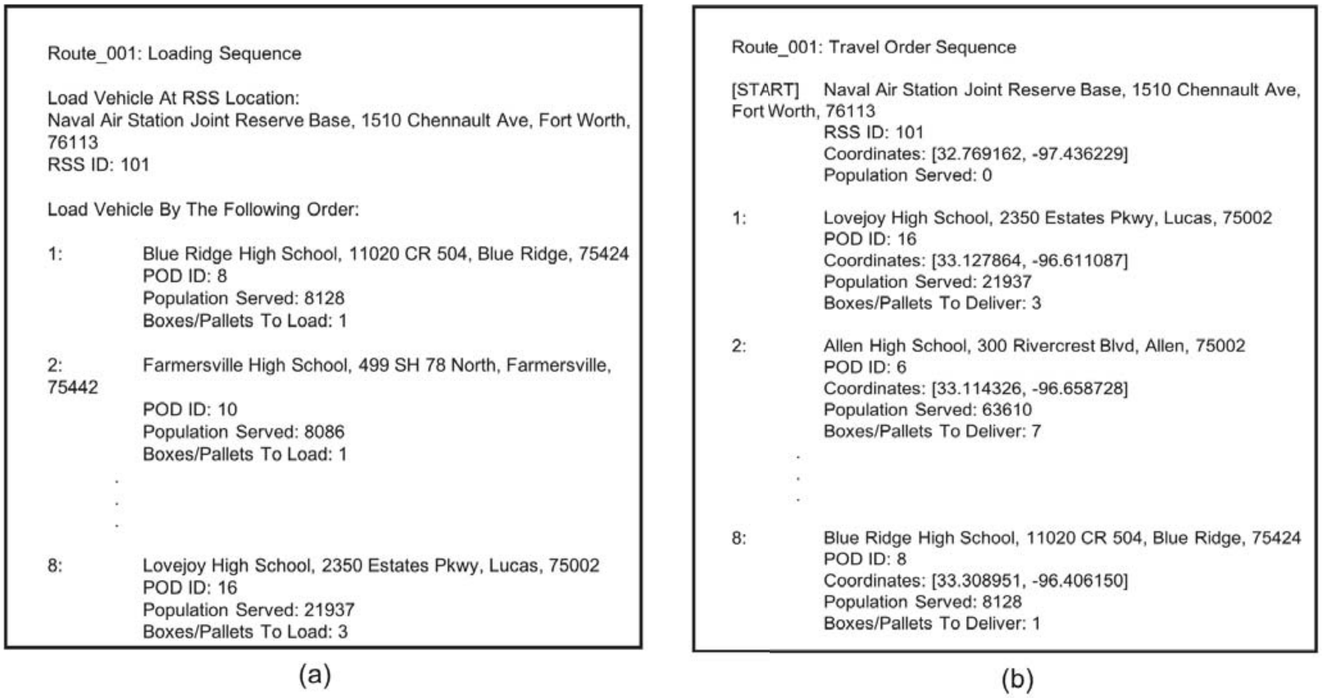


FIGURE 7.7. Software interface showing an algorithm performance analysis based on the maximum allowed time

cates if the routes are currently exceeding the maximum time and/or capacity. Specifically, rows with a red background indicate that the route currently exceeds the maximum time allowed. The blue background indicates the route can be completed with the allowed time, but sum total of the POD's demand included in the route exceed the vehicle's capacity. Rows with a white background indicate that the route is in compliance with all of the constraints.

Figure 7.7 depicts the interface for presenting an analysis that is similar to Table 7.2. This enables planners to determine the additional costs of reducing the allowed time for all routes to be completed by. Once a strategy has been picked, all of the parameters have been set, and a solution is determined, the framework will present the resulting routes so that they can be used by both the planners and those carrying out the plans. Figure 7.8 shows an example of the reports generated for each route within a single solution.



(a)

Route_001: Loading Sequence

Load Vehicle At RSS Location:
Naval Air Station Joint Reserve Base, 1510 Chennault Ave, Fort Worth, 76113
RSS ID: 101

Load Vehicle By The Following Order:

1: Blue Ridge High School, 11020 CR 504, Blue Ridge, 75424
POD ID: 8
Population Served: 8128
Boxes/Pallets To Load: 1

2: Farmersville High School, 499 SH 78 North, Farmersville, 75442
POD ID: 10
Population Served: 8086
Boxes/Pallets To Load: 1

.
.
.

8: Lovejoy High School, 2350 Estates Pkwy, Lucas, 75002
POD ID: 16
Population Served: 21937
Boxes/Pallets To Load: 3

(b)

Route_001: Travel Order Sequence

[START] Naval Air Station Joint Reserve Base, 1510 Chennault Ave, Fort Worth, 76113
RSS ID: 101
Coordinates: [32.769162, -97.436229]
Population Served: 0

1: Lovejoy High School, 2350 Estates Pkwy, Lucas, 75002
POD ID: 16
Coordinates: [33.127864, -96.611087]
Population Served: 21937
Boxes/Pallets To Deliver: 3

2: Allen High School, 300 Rivercrest Blvd, Allen, 75002
POD ID: 6
Coordinates: [33.114326, -96.658728]
Population Served: 63610
Boxes/Pallets To Deliver: 7

.
.
.

8: Blue Ridge High School, 11020 CR 504, Blue Ridge, 75424
POD ID: 8
Coordinates: [33.308951, -96.406150]
Population Served: 8128
Boxes/Pallets To Deliver: 1

FIGURE 7.8. Software result reporting for enhancing public health officials' and response planners' bio-emergency mitigation efforts. (a) shows the report for each individual route defining the amount and sequence to load the supplies into the truck. (b) shows the report for each route of their sequence of stops in the order they must be visited in. (b) is in the reverse order of the report shown in (a).

CHAPTER 8

SUMMARY AND CONCLUSION

Optimization of relief networks in humanitarian logistics often exemplifies the need for solutions that are feasible given a hard constraint on time. The distribution of medical countermeasures immediately following a biological disaster event is often required to be completed within a short time-frame. When these supplies are not distributed within the maximum time allowed, the severity of the disaster is quickly exacerbated. Therefore emergency response plans that fail to facilitate the transportation of these supplies in the time allowed are simply not acceptable. Thus certain optimization solutions that fail to satisfy this criterion would be deemed infeasible. This creates a conflict with the priority optimization objective in most variants of the generic VRP. Instead of efficiently maximizing the usage of vehicle resources available to construct a feasible solution, these variants ordinarily prioritize the construction of a minimum cost set of vehicles routes. A comparison of these conflicting objectives is presented in Section 1.1.

The research presented in this dissertation focused specifically on the design and analysis of computational methods to address these high-consequence variants of the vehicle routing problem. The minimization of the vehicle fleet size, in particular, is selected as the VRP's primary optimization objective. Accordingly, this optimization priority describes the fundamental objective underlying the research conducted in this dissertation. The context of these studies are structured consistently around a continuous partitioning heuristic.

Due to the conflict between the objectives of the minimization of total vehicles and minimizing the total travel time, the optimization of the time and capacity constraints in the context of minimizing vehicles are independently examined. Lower bounding procedures to determine the minimum number of vehicles required for instances of the OVRP-UT (i.e., only the time constraint is considered while vehicle capacity is disregarded) were described in Chapter 3. Given a set of cities, Algorithm 1 defines the strategies to construct a lower bound on the total number of vehicles required for the OVRP-UT constraints to be satisfied.

The analysis and quantification in determining the feasibility of three tours with complete certainty for any independent set of cities for the OVRP-UT were presented. Accordingly, methods to estimate this lower bound of vehicles without an explicitly defined solution is shown in Chapter 4. An upper bounding procedure for the minimum fleet size that is instead based around the OVRP-UC (i.e., only the vehicle capacity is considered while the time constraints are disregarded) were introduced in Chapter 5. A worst-case performance analysis of this bounding procedure is also examined.

A formal definition of the optimization problem for the high-consequence VRP variant that models a bio-emergency response was presented in Chapter 6. Moreover, the design of an efficient meta-heuristic algorithm based on a continuous spatial partitioning scheme were presented. Specifically, Algorithm 4 constructs a minimized set of vehicle routes for variants of the VRP that are defined by the high-consequence constraints in a bio-emergency scenario. Notably, the hard constraints for the maximum tour duration and the identical vehicle capacity are prioritized in this optimization.

The analysis and application of Algorithm 4 was illustrated in Chapter 7. Namely, a case study is constructed from a large region in the state of Texas to exemplify the direct application of this two-phase algorithm in the context of bio-emergency response. Multiple priority optimization strategies were introduced and their associated performance is compared and examined accordingly from the regional case-study.

8.1. Broader Impacts

These methods and strategies have been implemented as a set of computational tools designed specifically for constructing vehicle routes optimized for the delivery of resources to dispensing facilities during a bio-emergency. Furthermore, these tools have been integrated into an existing bio-emergency response planning framework[51], currently being used by public health officials and emergency response planners. This integration enhances their ability to deriving a set of vehicle routes that maximize mitigation efforts in coordination with the usage of available resources.

Although these tools were developed development in the context of a bio-emergency they are easily extended to a broader domain. For example, most of the natural disasters illustrated in Section 2.1 can be modeled similarly as the high-consequence VRP variant described in this research. This would support the usage of the resulting methods and algorithms from this research in an almost exact manner. Moreover, any VPP problem instance that demonstrates these hard-constraints in time and defines the primary objective as the minimizing the number of required vehicles is applicable. For this reason, most of the research conducted in this dissertation is described and formalized in a manner consistent with that of existing VRP literature and research.

8.2. Research Limitations and Future Work

Performance Benchmarks. One limitation of the research conducted in this dissertation is inability to benchmark the performance of the meta-heuristic algorithm (Algorithm 4) presented in Chapter 6 for direct comparison with other VRP approximation algorithms. As illustrated throughout this dissertation, the primary optimization objective defined by public VRP instances commonly used in literature to benchmark an algorithm's performance is not directly aligned with the VRP variants of focus in this research. As a consequence, analyzing the performance of this algorithm in a comparative manner with existing approximation algorithms is not included in this dissertation. Moreover, due to most existing heuristics being tailored directly for the individual problem instance used as the benchmark, comparing any solution constructed from this algorithm with the *best known solutions* would similarly not provide a good indication of its performance. Nonetheless, modification of these published/known VRP instances provides potential opportunities for performance comparison. Furthermore, the design of new benchmarks that measure the robustness of the algorithm as it relates to the parameter input constraints would be of interest. Specifically, quantifying the relative change in performance as the distribution of the customer's location and associated demand are adjusted towards the worst-case scenario.

Resource Priority Optimization Strategies. In Chapter 6, a scoring function was presented in that constructed two new routes with each partition, independently assigning customers to these tours using a two-tier priority weighting scheme. Depending on the tier, usage of the time or capacity available for each route would dictate the assignment of these customers. The value assigned to these weights is restricted between certain values, as defined in Algorithm 4, thus guaranteeing a feasible solution resulting from all instances. However, observing the performance of the varying priority strategies resulting from the case study presented in Chapter 7 illustrates the potential for improvement. In particular, if certain problem instances are dominated by one of two core constraints (i.e., time or capacity), then improvements are likely to be achieved when prioritizing the optimization of the dominating constraint. For example, assume all of the customers are sparsely distributed throughout a very large region. Moreover, assume the average customer demand is small relative to the vehicle's capacity. Clearly this describes a scenario that prioritizes the time component for each route. With this in mind, additional research demonstrating the relationships between these two constraints would be beneficial. Accordingly, the independent analysis of these constraints presented in this dissertation could be extended for this purpose. Properties of this relationship that we believe would be particularly of use are as follows: (i) compare the growth of complexity of each constraint (i.e., which one grows at a faster rate), (ii) demonstrate how the optimization of one constraint can negatively affect the other, (iii) quantify how the relative change to one of the constraint's distribution either directly or indirectly affects the other.

Heuristic Improvements. In our implementation of the SpatialPartition procedure presented in Algorithm 5, the route with the highest score is assigned a customer using the *Double-ended nearest neighbor heuristic* (DENN) [13]. Since each set is continuously partitioned, each route in this entire first phase is constructed just to determine the feasibility for a tour through each resulting subset of customers. Recall that the second phase of Algorithm 4 is to use the final partitioning from the first phase and apply more robust TSP heuristics for the construct of the final resulting routes. Hence, this insertion heuristic is a reasonable option due to its simplicity and performance. Accordingly, the constructed tour is never for more than $(\lceil \log(n) \rceil)/2$ times the optimal tour for these assigned customers with this heuristic [13]. Yet the minimization of cost for both resulting routes defined by the partition is not considered in the optimization of this heuristic. Instead new insertion heuristics can be designed that optimize the assignment of customers defining the partition as well. This would likely improve the efficiency of the partitioning procedure in the algorithm and consequently has the potential to further reduce of the fleet size. The design of an insertion heuristics that select a customer for a route based on its high insertion cost to be included in the other route is one such possibility that could provided improvements. In other words, instead of selecting the customer with the lowest insertion cost, the customer with the largest insertion cost to the other tour is selecting. Additionally, this type of heuristic could provide improvements without much increase to the overall execution time.

Further, optimization strategies need to be explored for allowing vehicles to fully utilize their capacity by delivering a portion of a POD's supplies even if they cannot deliver all of the supplies due to capacity constraints. Other strategies could include identifying optimal auxiliary staging facilities (e.g., Alpha PODs) that have the capability to deliver to smaller PODs within their proximity, leading to improvements in time and vehicle requirements. Improvements can also be made by allowing multi-stage delivery strategies that would further decrease capacity and vehicle requirements by allowing vehicles to drop off only the supplies that a POD will need for a specific time window, and to complete delivery before the end of that time window (i.e., before the MCM resources have all be distributed to the population).

Pre- & Post-Optimization. From the results displayed in Table 7.1 for the 2PO optimization strategy described in Chapter 7, the *Route Statistics* indicate the potential for improvement with pre- and post-optimization strategies. Specifically, examination of the actual routes constructed as shown in Table A.1 reveals characteristics with the potential to be further improved. For example, the three routes within the solution that require the lowest total time to complete their tour is shown in Table 8.1. These routes indicate the possibility of combining two or more routes as a single solution. As a result, the required fleet size can be reduced.

Route ID	Elapsed Time (hours)	Population Served	Tour (by POD ID)
Route_049	1.20	100,324	[342, 283, 294]
Route_051	1.59	102,386	[342, 55, 56]
Route_037	1.71	125,217	[342, 301, 384, 310]

Table 8.1: Texas DSHS Region 2/3 (Case Study): The three routes with the minimum elapsed time out of the total fifty-one route 2PO solution

Assuming that we only have to make improvements on these three routes (Route_049, Route_051, Route_037), we know that because the population served in Route_049 plus the population served in Route_051 has a total of 202,710 which is less than the capacity of a single vehicle, that if the travel time from combining these routes is lower than the allowed time of 12 hours, we can reduce the total vehicle count by 1. For example, if the travel time from the end of Route_049 (POD 294) to the first POD in Route_051 (POD 55) is less the remaining allowed time calculated as: 12 hours (total allowed time) − 1.2 hours (Route_049 total time) − 1.59 hours (Route_051 total time) + (longest travel time from RSS 342 to either POD 283 or POD 54), then these routes can be combined by just easily serving one immediately after the other. To combine the third route (Route_037), we must first return to the RSS to pick up the MCMs required to fulfill the population served by this route because after the combination of the first two routes, there is not enough capacity remaining in the vehicle to append this route's PODs without a return to the RSS. Although this example seems to be relatively straightforward when looking only at three routes, finding the best solution for merging routes is a complex problem for which approximation algorithms must be developed. Nonetheless, Figure 8.1 presents potential improvements if return trips to the depot are allowed by adding a post-optimization step to the result of the 2PO optimization. The implementation of this post-optimization step basically is a modified version of a dynamic programming O-1 knapsack algorithm.

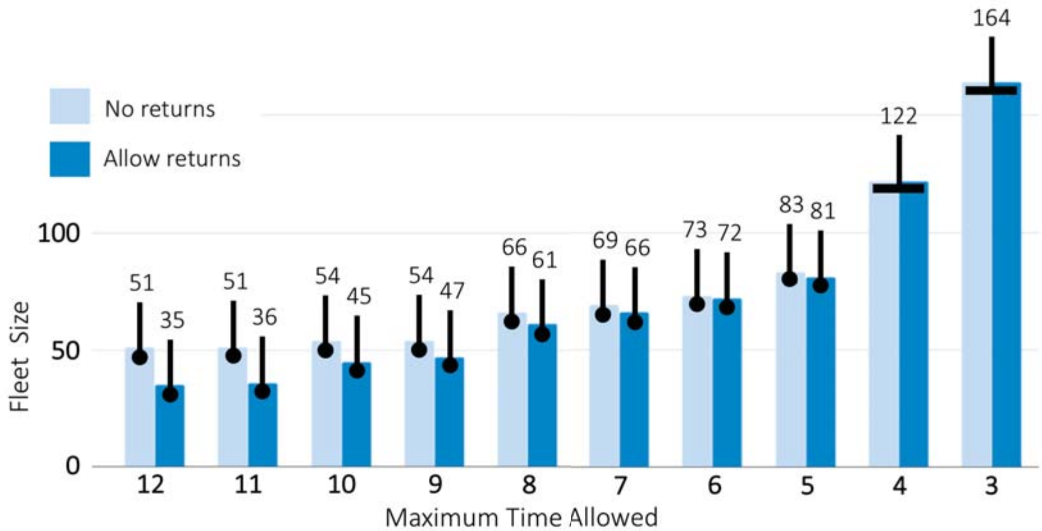


Figure 8.1. Comparison of the minimum fleet size required if routes are allowed to return to the depot

In addition to the potential improvements from combining routes, one other possible improvement is examined regarding the distribution of POD demand. Figure 8.2 illustrates the population (i.e., demand) distribution for the PODs in Region 2/3.

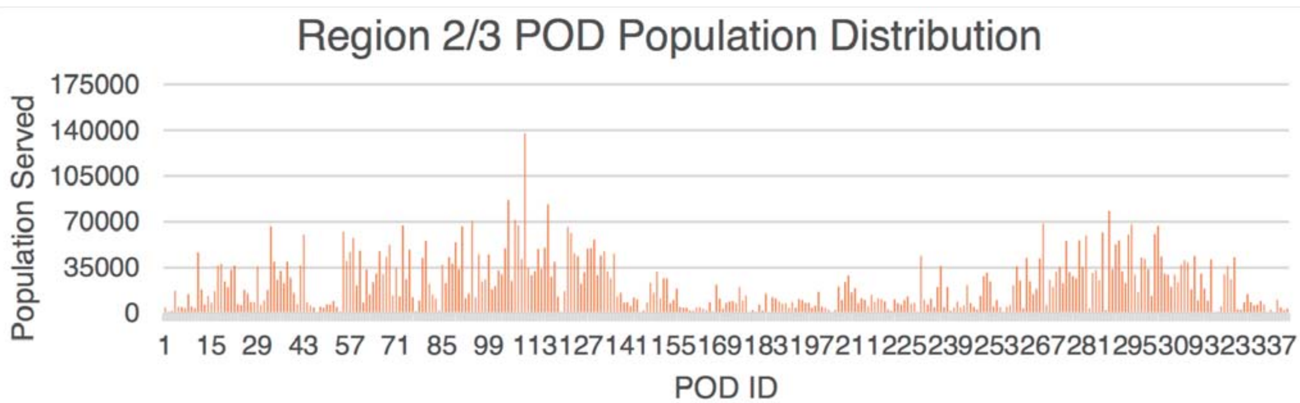


Figure 8.2. Texas DSHS Region 2/3 (Case Study): Population demand distribution

It is evident from this figure that the population assigned per POD is far from being identical. The analysis of the OVRP-UC conducted in Chapter 5 indicates the impact that the demand distribution can have on the optimization of capacity utilization. Accordingly, modifications to the POD placement that result in a more identical population assignment per POD could lead to improved results (i.e., fewer vehicles required).

APPENDIX

SUPPORTING MATERIAL

Route ID	Total Time (hours)	Population Served	Tour (By POD ID)
Route_001	3.87	163,013	[342,106,295,266,305,148,209]
Route_002	9.36	96,247	[342,191,200,166,159,161,160,177,13,15,40,28,26]
Route_003	7.22	111,669	[342,,133,329,335,334,330,331,183,187,253]
Route_004	8.34	74,321	[342,,175,174,171,172,173,165,163,162,164,157,158]
Route_005	8.29	34,090	[342,,170,176,46,223,10,181]
Route_006	6.06	197,054	[342,,112,114,85,113,210,217,199,190]
Route_007	8.67	93,312	[343,339,315,314,317,312,8,5,7]
Route_008	5.70	198,245	[342,292,281,286,291,285,288,304,241,237]
Route_009	5.16	190,276	[342,34,38,18,19,30,22,23]
Route_010	4.47	187,739	[342,259,57,97,111,65,63,77]
Route_011	7.08	192,353	[342,62,73,76,70,92,93,149,144,143,146,145]
Route_012	9.02	161,185	[343,325,324,323,322,321,327,9,222,203,204,202]
Route_013	3.72	198,023	[342,282,270,276,275,58,135]
Route_014	10.84	174,830	[342,4,255,252,318,316,313,230,257]
Route_015	8.37	152,527	[342,79,89,198,121,192,193,197,196,195,194,189]
Route_016	8.08	173,482	[342,280,273,235,244,247,246,243,239,240,238,336,337,333]
Route_017	3.60	195,946	[342,33,36,21,11,25]
Route_018	9.16	31,830	[343,201,338,182,167,180,179,178,326,328]
Route_019	6.81	19,379	[343,1,340,341,3,2,221,220,54]
Route_020	8.49	43,166	[343,256,254,141,140,142,45,154,47]
Route_021	2.94	210,308	[342,64,105,104,116]
Route_022	4.80	175,966	[342,69,251,249,250,118,14]
Route_023	4.77	208,282	[342,59,81,96,60,124,31]
Route_024	10.02	73,934	[343,320,6,44,155,156,188,184,185,186]
Route_025	8.63	85,161	[342,332,236,242,245,232,231,234,233,319]
Route_026	6.49	193,456	[342,126,129,130,139,53,50,49,48,51,52]
Route_027	4.18	187,691	[342,311,302,122,128,127,138]
Route_028	8.05	97,561	[342,268,211,219,218,216,214,213,215,228,227,226]
Route_029	7.87	176,478	[342,261,78,98,74,150,152,153,151,147,224,229]
Route_030	5.26	199,493	[342,109,125,17,16,20,12,168]
Route_031	3.74	92,232	[342,132,137,41,169]
Route_032	2.57	172,303	[342,297,279,289,303]
Route_033	4.26	166,083	[342,94,87,67,72,225]
Route_034	2.61	113,557	[342,100,91,103]
Route_035	3.50	186,115	[342,298,293,263,258,260,264]
Route_036	4.22	176,822	[342,117,35,32,29,24,27]
Route_037	1.71	125,217	[342,301,284,310]
Route_038	5.26	210,572	[342,131,37,42,119,101,102,84]
Route_039	2.19	180,048	[342,267,108,99]
Route_040	1.93	197,379	[342,308,309,110]
Route_041	2.09	93,628	[342,115,43]
Route_042	1.99	159,922	[342,287,290,277]
Route_043	3.24	161,010	[342,265,95,71,75,66]
Route_044	2.55	150,038	[342,107,274,207]
Route_045	3.42	182,445	[342,296,134,136,123,39]
Route_046	4.64	177,211	[342,262,271,269,306,205,206,208]
Route_047	2.78	166,109	[342,307,300,299,272,278]
Route_048	3.63	135,149	[342,80,68,120,212,248]
Route_049	1.20	100,324	[342,283,294]
Route_050	3.74	128,034	[342,83,86,82,61,90,88]
Route_051	1.59	102,386	[342,55,56]

Table A.1: Texas DSHS Region 2/3 (Case Study): 2PO optimized vehicle route solution

FIGURE A.1. Example road network data

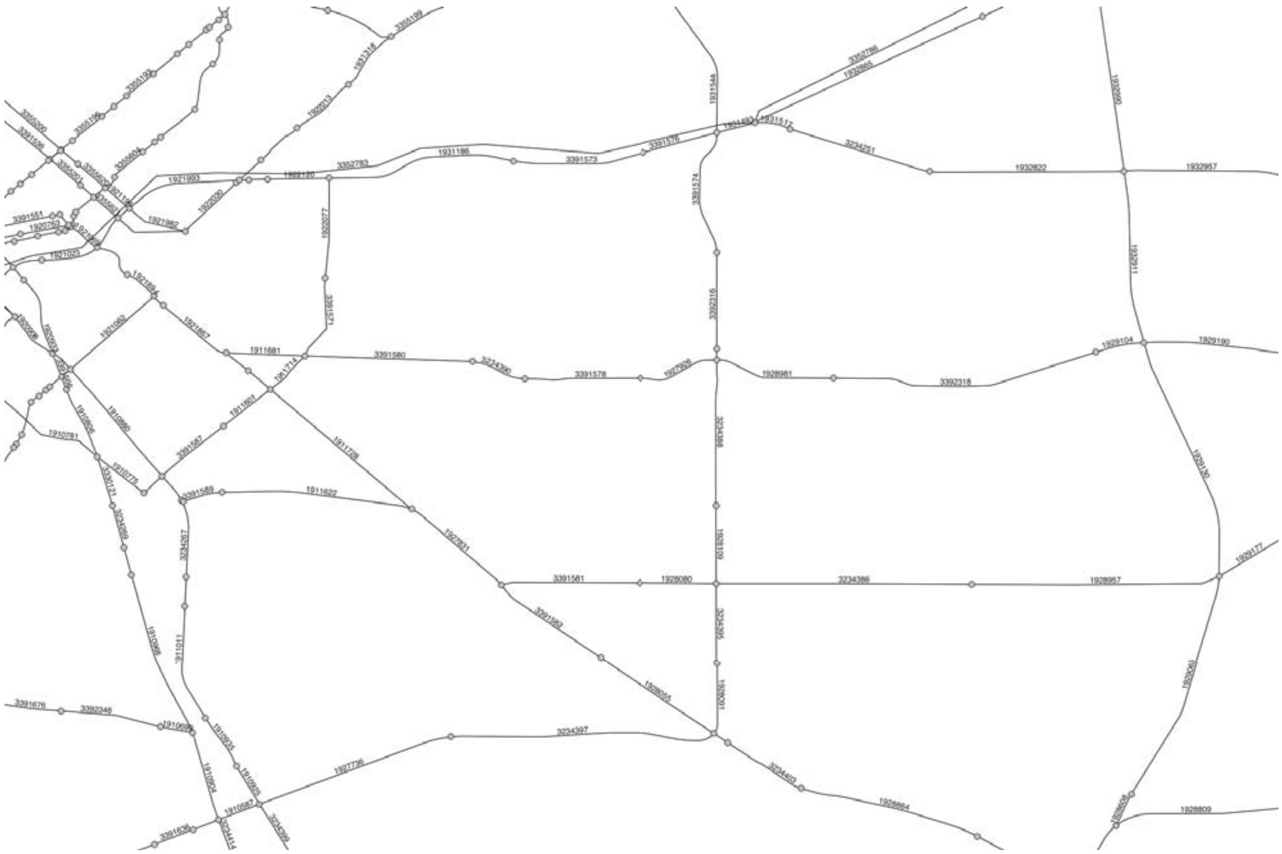


Figure A.2. Spatial database representation of example road network data

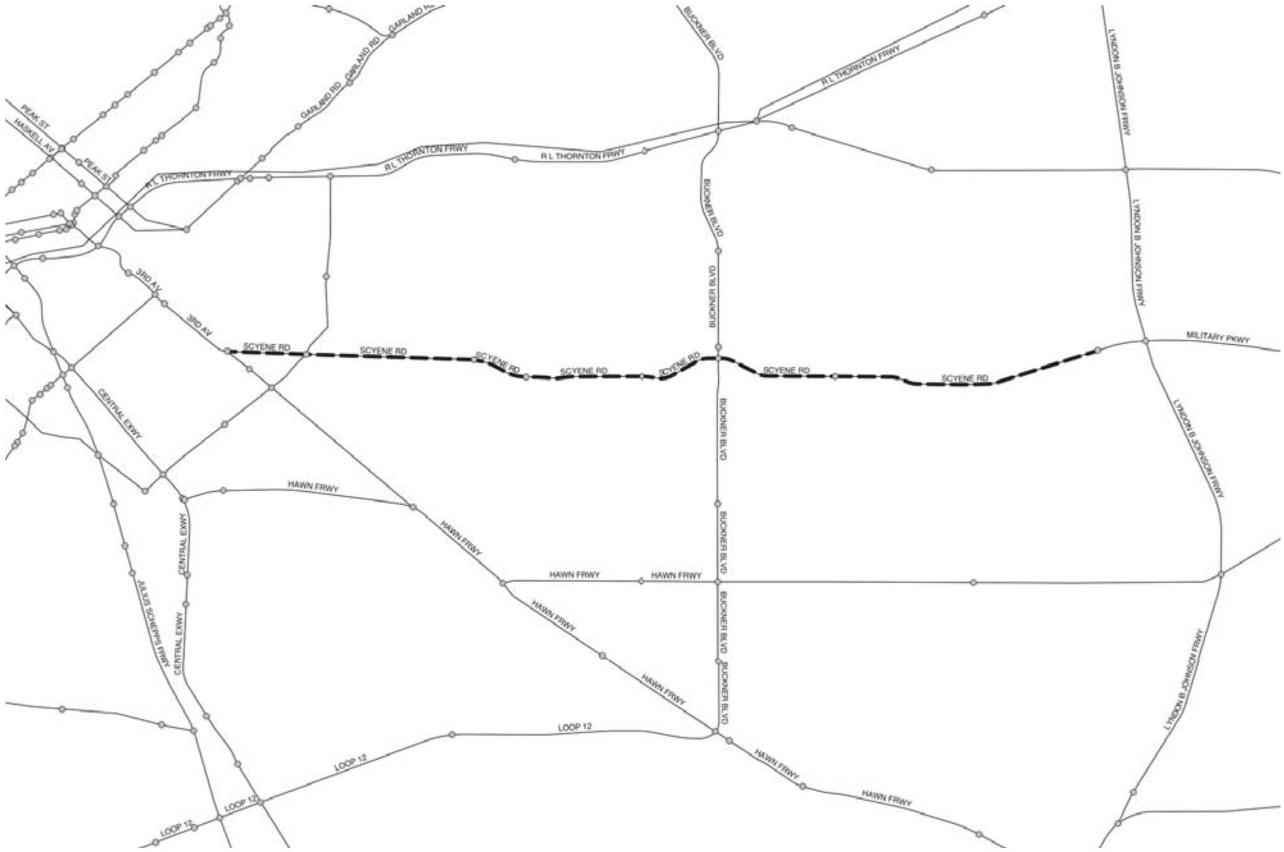


FIGURE A.3. Example showing distance calculations between two points by combining line-string segments within the road network data

BIBLIOGRAPHY

[1] A, Environment Canterbury, Anantha Kumar Duraiappah, Shahid Naeem, Tundi Agardy, Neville J. Ash, H. David Cooper, Sandra Díaz, Daniel P. Faith, Georgina Mace, Jeffrey a. McNeely, Harold a. Mooney, Stephen Alfred A. Oteng-Yeboah, Henrique Miguel Pereira, Polasky, Christian Prip, Walter V. Reid, Cristián Samper, Peter Johan Schei, Robert Scholes, Frederik Schutyser, Albert Van Jaarsve, Millennium Ecosystem Assessment, A.G Fallis, Heather M Hunter, T. Larsen, L. T. Bach, R. Salvatteci, Y. V. Wang, N. Andersen, M. Ventura, M. D. McCarthy, and MEA, *Receiving, Distributing, and Dispensing Strategic National Stockpile Assets: A Guide to Preparedness, Version11*, vol. 12, 2013.

[2] Mahdieh Allahviranloo, Joseph Y.J. Chow, and Will W. Recker, *Selective vehicle routing problems under uncertainty without recourse*, Transportation Research Part E: Logistics and Transportation Review 62 (2014), 68–88.

[3] K Altnel and T Öncan, *A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem*, Journal of the Operational Research Society 56 (2005), no. 8, 954–961.

[4] D. L. Applegate, R. E. Bixby, V. Chvatal, and W.J Cook, *The traveling salesman problem: A computational study (Princeton Series in Applied Mathematics)*, 2006.

[5] Alessandro Arlotto and J. Michael Steele, *Beardwood-halton-hammersley theorem for stationary ergodic sequences: A counterexample*, Annals of Applied Probability 26 (2016), no. 4, 2141–2168.

[6] Sanjeev Arora, *Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*, Journal of the ACM 45 (1998), no. 5, 753–782.

[7] Assistant Secretary for Preparedness & Response, *Pandemic and All-Hazards Preparedness Act Progress Report on the Implementation of Provisions Addressing At-Risk Individuals*, Public Law (2008), no. August.

[8] Barrie M. Baker and Janice Sheasby, *Extensions to the generalised assignment heuristic*

for vehicle routing, European Journal of Operational Research 119 (1999), no. 1, 147–157.

[9] Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi, *An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts*, Mathematical Programming 115 (2008), no. 2, 351–385.

[10] Jillian Beardwood, J. H. Halton, and J. M. Hammersley, *The shortest path through many points*, Mathematical Proceedings of the Cambridge Philosophical Society 55 (1959), no. 04, 299.

[11] Tolga Bektas, *The multiple traveling salesman problem: An overview of formulations and solution procedures*, Omega 34 (2006), no. 3, 209–219.

[12] M. Bellmore and G. L. Nemhauser, *THE TRAVELING SALESMAN PROBLEM : A SURVEY*, Operations Research 16 (1968), no. 3, 538–558.

[13] Jon Jouis Bentley, *Fast Algorithms for Geometric Traveling Salesman Problems*, ORSA Journal on Computing1 4 (1992), no. 4, 387–411.

[14] Dimitris J Bertsimas and David Simchi-Levi, *A NEW GENERATION OF VEHICLE ROUTING RESEARCH: ROBUST ALGORITHMS, ADRESSING UNCERTAINTY*, Open Geospatial Consortium, Inc 44 (1996), no. 2, 286–304.

[15] Jean Marie Bourjolly and Vianney Rebetez, *An analysis of lower bound procedures for the bin packing problem*, Computers and Operations Research 32 (2005), no. 3, 395–405.

[16] Leo Brewer and R. H. Lamoreaux, *The Impact of Metaheuristics on Solving the Vehicle Routing Problem*, 1 (1980), no. 2, 93–95.

[17] Tonci Caric and Hrvoje Gold, *Vehicle Routing Problem*, 2008.

[18] N Christofides, A Mingozzi, and Paolo Toth, *Exact Algorithms for the Vehicle Routing Problem , Based on Spanning Tree and Shortest Path Relaxations*, Mathematical Programming 20 (1981), 28.

[19] G Clarke and J.W. Wright, *SCHEDULING OF VEHICLES FROM A CENTRAL DEPOT TO A NUMBER OF DELIVERY POINTS A NUMBER*, (1962), 568–581.

[20] Jean-François Cordeau, Gilbert Laporte, Martin W P Savelsbergh, and Daniele Vigo, *Chapter 6 Vehicle Routing*, Transportation 14 (2007), no. 06, 367–428.

[21] Jean-frangois Cordeau, *A Branch-and-Cut Algorithm for the Dial-a-Ride Problem*, 54 (2006), no. 3, 573–586.

[22] Teodor Gabriel Crainic, Guido Perboli, Miriam Pezzuto, and Roberto Tadei, *Computing the asymptotic worst-case of bin packing lower bounds*, European Journal of Operational Research 183 (2007), no. 3, 1295–1303.

[23] G. B. Dantzing and J. H. Ramser, *The Truck Dispatching Problem*, Management Science 6 (1959), no. 1, 80–91.

[24] Edsger Wybe Dijkstra, *A note on two problems in connection with graphs*, 1959, pp. 269–271.

[25] T Duane, *A Generalized Assignment Heuristic for Vehicle Routing*, Aerospace (1974), no. January.

[26] S. R. Finch, *Mathematical constants*, Encyclopedia of Mathematics and its Applications, Cambridge University Press 94 (2003).

[27] Marshall Fisher, *Vehicle routing*, Handbooks in Operations Research and Management Science 8 (1995), 1–33.

[28] Billy E Gillett and Leland R Miller, *A Heuristic Algorithm for the Vehicle-Dispatch Problem*, (1971), 340–349.

[29] D. Guha-Sapir, R. Below, and Ph. Hoyois, *EM-DAT: The CRED/OFDA International Disaster Database*, Tech. report, Université Catholique de Louvain, Brussels, Belgium, 2016.

[30] D. Hochbaum, *Approximation Algorithms for NP-Hard Problems*, 1997.

[31] Dorit S Hochbaum, *Geometric Problems*, Approximation Algorithms for NP-Hard Problems, 1997.

[32] Harold Hochman and James Rodgers, *Pareto Optimal Redistribution*, The American Economic Review 59 (1969), 524–557.

[33] Sung Jin Hwang, Steven B. Damelin, and Alfred O. Hero, *Shortest path through random points*, Annals of Applied Probability 26 (2016), no. 5, 2791–2823.

[34] Tamara Jimenez, Armin R. Mikler, and Chetan Tiwari, *A novel space partitioning algorithm to improve current practices in facility placement*, IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 42 (2012), no. 5, 1194–1205.

[35] Aa Juan, J Faulin, J Jorba, D Riera, D Masip, and B Barrios, *On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics*, Journal of the Operational Research Society 62 (2010), no. 62, 1085–1097.

[36] Narenda Karmarkar and Richard M. Karp, *The Differencing Method of Set Partitioning*, 1983, pp. 181–203.

[37] R. M. Karp, *Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane*, Mathematics of Operations Research 2 (1977), no. 3, 209–224.

[38] M Khorsi, A Bozorgi-Amiri, and B Ashjari, *A Nonlinear Dynamic Logistics Model for Disaster Response under Uncertainty*, Journal of Mathematics and Computer Science 7 (2013), 63–72.

[39] Scott Kirkpatrick, D Gelatt Jr., and Mario P Vecchi, *Optimization by simmulated annealing*, Science 220 (1983), no. 4598, 671–680.

[40] Suresh Nanda Kumar, *A Survey on the Vehicle Routing Problem and Its Variants*, Intelligent Information Management 04 (2012), no. 03, 66–74.

[41] G. Laporte, *Fifty Years of Vehicle Routing*, Transportation Science 43 (2009), no. 4, 408–416.

[42] G. Laporte, *The vehicle routing problem: An overview of exact and approximate algorithms*, European Journal of Operational Research 59 (1992), no. 3, 345–358.

[43] G. Laporte, *What You Should Know about the Vehicle Routing Problem*, Naval Research Logistics 55 (2007), no. April 2007, 541–550.

[44] Public Law, *Title I Strengthening National Preparedness and Response for Public Health Emergencies*, (2013), 161–197.

[45] J Lenstra and a Rinnooy Kan, *Complexity of Vehicle Routing and Scheduling Problems*, 1981, pp. 221–227.

[46] R. W. Lysgaard, J., Letchford, A. N., & Eglese, *A New Branch-and-Cut Algorithm for the Capacitated Vehicle Routing Problem*, Mathematical Programming 100 (2004), no. 2, 423–445.

[47] Thomas Magnanti, *Combinatorial optimization and vehicle fleet planning: Perspectives and prospects*, Networks 11 (1981), no. 2, 179–213.

[48] Silvano Martello and Paolo Toth, *Lower bounds and reduction procedures for the bin packing problem*, Discrete Applied Mathematics 28 (1990), no. 1, 59–70.

[49] Joseph L. Nates and Virginia A. Moyer, *Lessons from Hurricane Katrina, tsunamis, and other disasters*, Lancet 366 (2005), no. 9492, 1144–1146.

[50] G LaporteandY. Norbert, *Exact algorithms for the vehicle routing problem*, Annals of Discrete Mathematics 31 (1987), 147–184.

[51] Marty ONeill, A R Mikler, Indrakanti, Tamara Schneider, *RE-PLAN: An Extensible Software Architecture to Facilitate Disaster Response Planning*, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2014), 1569–1583.

[52] Mike Paterson and Vlado Dančík, *Longest common subsequences*, Mathematical Foundations of Computer Science 1994 (Berlin, Heidelberg) (Igor Prívara, Branislav Rovan, and Peter Ruzička, eds.), Springer Berlin Heidelberg, 1994, pp. 127–142.

[53] Tantikorn Pichpibul and Ruengsak Kawtummachai, *An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem*, ScienceAsia 38 (2012), no. 3, 307–318.

[54] V Pillac, M Gendreau, C Gueret, and A Medaglia, *A review of dynamic vehicle routing problems*, European Journal of Operational Research 225 (2013), no. 1, 1–11.

[55] M. L. Quandt and Balinski R. E., *On an Integer Program for a Delivery Problem*, Operations Research 12 (1964), no. 2, 300–304.

[56] Xiao-xia Rong, Yi Lu, Rui-rui Yin, and Jiang-hua Zhang, *A Robust Optimization Approach to Emergency Vehicle Scheduling*, 2013 (2013).

[57] Tamara Schneider, A R Mikler, and Marty ONeill, *Analyzing Response Feasibility for Bioemergencies*, International Joint Conferences on System Biology, Bioinformatics and Intelligent Computing (IJCBS09) (Shanghai, China), aug 2009.

[58] Tamara Schneider, A R Mikler, and Marty ONeill, *Computational Tools for Evaluating Bioemergency Contingency Plans*, Proceedings of the 2009 International Conference on Disaster Management (New Forest, England), sep 2009.

[59] Armin Scholl, Robert Klein, and Christian Jürgens, *Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem*, Computers and Operations Research 24 (1997), no. 7, 627–645.

[60] Jiuh Biing Sheu, *Dynamic relief-demand management for emergency logistics operations under large-scale disasters*, Transportation Research Part E: Logistics and Transportation Review 46 (2010), no. 1, 1–17.

[61] Washington Square, *A BRANCH AND CUT ALGORITHM FOR THE RESOLUTION OF LARGE-SCALE TRAVELING SALESMAN PROBLEMS*, 33 (1991), no. 1, 60–100.

[62] Eliana M. Toro O., Antonio H. Escobar Z., and Mauricio Granada E., *Literature Review on the Vehicle Routing Problem in the Green Transportation Context*, Luna Azul (2015), no. 42, 362–387.