8-2016

Incorporating A New Class of Uncertainty in Disaster Relief Logistics Planning

Emre Kirac
University of Arkansas, Fayetteville

Incorporating A New Class of Uncertainty in Disaster Relief Logistics Planning

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Industrial Engineering

by

Emre Kirac
Uludag University
Bachelor of Science in Textile Engineering, 2008
Colorado State University - Pueblo
Master of Science in Systems and Industrial Engineering, 2011

August 2016
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Dr. Ashlea Bennett Milburn
Dissertation Director

Dr. Edward A. Pohl
Committee Member

Dr. Chase Rainwater
Committee Member

Dr. Clarence Wardell III
Committee Member

Abstract

In recent years, there has been a growing interest among emergency managers in using social data in disaster response planning. However, the trustworthiness and reliability of posted information are two of the most significant concerns, because much of the user-generated data is initially not verified. Therefore, a key tradeoff exists for emergency managers when considering whether to incorporate social data in disaster planning efforts. By considering social data, a larger number of needs can be identified in a shorter amount of time, potentially enabling a faster response and satisfying a class of demand that might not otherwise be discovered. However, some critical resources can be allocated to inaccurate demands in this manner. This dissertation research is dedicated to evaluating this tradeoff by creating routing plans while considering two separate streams of information: (i) unverified data describing demand that is not known with certainty, obtained from social media platforms and (ii) verified data describing demand known with certainty, obtained from trusted traditional sources (i.e. on the ground assessment teams). These projects extend previous models in the disaster relief routing literature that address uncertainty in demand. More broadly, this research contributes to the body of literature that addresses questions surrounding the usefulness of social data for response planning.

Acknowledgements

There are many people whose support enabled the completion of this dissertation. First and foremost, I would like to thank my advisor, Dr. Ashlea Bennett Milburn, for her guidance, understanding, patience, and most importantly, her friendship during my Ph.D. education. I appreciate all her contributions of time, ideas, and funding opportunities to make my Ph.D. education productive and stimulating. She has given me endless invaluable direction and support throughout my study. The joy and enthusiasm she has for her research was contagious and motivational for me, even during tough times in the Ph.D. pursuit. She has taught me how to do research and how to write scientific texts. I am also thankful for the excellent example she has provided as a successful advisor. She genuinely cares about her students and serves as a life mentor as well as an academic mentor. I am forever grateful for her great mentorship, and endless personal and professional support.

I would also like to thank my committee members for their insightful comments: Dr. Edward Pohl, Dr. Chase Rainwater, and Dr. Clarence Wardell. I am also grateful to all my professors for sharing their knowledge consistently and without any hesitation to enable me to succeed in my studies.

I would like to thank all of my friends who become like a second family to me for the past five years. Specifically, I express my sincere gratitude to Ridvan Gedik, Furkan Oztanriseven, Serdar Kilinc, Sami Keskek, Ege Ozdemir, and Steve Sharp. I gladly express my sincere gratitude to the wonderful Industrial Engineering staff: Karen Standley, Carrie Pennington, Garn LeBaron, Sandy Sehon, and Tamara Ellenbecker for their tremendous help. I would also like to thank Emily Cruz for doing such a terrific job editing and correcting my grammar.

Finally, and most importantly, I am deeply thankful for my parents and my brother for their love, support, and sacrifices. Without them, this dissertation would never have been written. This last word of acknowledgements I have saved for my mom, Murvet Kirac, for her continuous prayers and unconditional support during all phases of my life.

Contents

List of Figures

List of Tables

List of Publications

1. Emre Kirac, Ashlea Bennett Milburn, and Clarence L. Wardell,"*The traveling salesman problem with imperfect information with application in disaster relief tour planning*," IIE Transactions, vol. 47(8), pp. 783-799, 2015.

1. INTRODUCTION

Throughout the world, many countries have faced a series of unpredictable natural disasters such as the Japanese earthquake and tsunami in 2011, Hurricane Sandy in the United States in 2012, and Typhoon Haiyan in the Philippines in 2013. Each year between 2003 and 2013, approximately 368 disasters were reported, causing over 99 thousand deaths and affecting almost 182 million people on average [1]. Disasters cause substantial social, cultural, and environmental damages with significant economic impacts. Humanitarian relief agencies and nongovernment organizations play a critical role in minimizing loss of life and alleviating human suffering after disasters by providing relief supplies such as water, food, shelter, and medical services. As such, logistics is a key to their activities and vital for the success of their operations.

Effective and efficient logistics management is important for timely response and recovery activities after a catastrophe [2]. These logistic activities are defined as providing the right support in the right amount at the right time at the right place to the right people. However, precise information regarding affected population– such as how many people need help, where they are located, and what type of assistance they need – is required for an effective and quick response. Traditionally, on-the-ground assessment teams are dispatched to the affected area within the first 24 hours after a disaster to collect this information [3]. These teams may be unfamiliar with the environment and the cultural structure of the population, which may inhibit active communication. This traditional information-gathering approach struggles to utilize local knowledge and engage the affected people [4]. Critical hours pass while required information is collected. However, social media facilitates gathering massive amounts of information regarding people in need in a short period of time following a disaster. In June 2013, Congresswoman Susan W. Brooks (R-IN), Chairman of the United States House Subcommittee on Emergency Preparedness, Response and Communications, said "*while social media originally started out as a way to share information among family* *and friends, it is evident that it has evolved to serve other functions…its use in preparing for, responding to and recovering from disasters.*" [5]. Many in the humanitarian response community are exploring ways to employ social media to collect more situational awareness information. As reported by a survey carried out by CNA and The National Emergency Management Association (NEMA) in 2012, all state emergency management agencies surveyed use social media in some capacity [6].

According to two national surveys conducted for the Red Cross in 2011 and 2012, 80% of Americans expect emergency responders to monitor and respond to social media, and 76% of survivors of natural disasters use social media to contact family and friends to make sure that they are safe [7, 8]. Consequently, several emergency agencies have increasingly utilized social media tools to find critical pieces of information posted online by affected individuals in recent years. For example, after superstorm Sandy, the social engagement team of the Red Cross tracked more than 2 million social media posts related to "Sandy" and "Red Cross" in the early weeks of the response [9]. The Federal Emergency Management Agency (FEMA) also used social media to monitor approximately 20 million tweets regarding Sandy in order to provide timely safety information [10]. During the relief efforts throughout the Philippines in the aftermath of Typhoon Haiyan in 2013, United Nations agencies and nongovernmental organizations employed social media in their disaster response operations [11].

Information posted to social media by affected people can be useful for disaster response planning if it describes a specific need and geographic location. Consider for example these two posts to the Ushahidi platform during the 2010 Haiti earthquake. Ushahidi is a non-profit software company that develops free and open source software for information collection, visualization, and interactive mapping. This request was posted at 16:55 on January 21, 2010, nine days after the earthquake [12]:

> "Our home address is Tabbare impasse laurent #9 we got also hit by the earthquake we are a number of 13 people we need water so we could drink."

And this post was at 13:32 on February 14, 2010 [12]:

> "I am in Simon in Militaire in Haiti. I have 70 People with me, I need shelter, food and medical Help."

These requests provide actionable information identifying specific needs, precise addresses and the number of individuals in need that emergency response agencies can utilize when planning distribution efforts. However, requests posted to social media platforms cannot be verified initially and some of them might be inaccurate. For example, during Hurricane Sandy, several individuals spread rumors and fake pictures via social media. Thus, inaccurate, false, or outdated information could complicate situational awareness and slow disaster response efforts [13]. Some emergency responders consider the accuracy of social data as a major barrier to their decision making process. However, not all requests posted on social media platforms are inaccurate, and in certain situations needs might not otherwise be discovered. In a December 2011 presentation on Real-Time Awareness, FEMA Administrator Craig Fugate proposed acting on all social data requests without attempting to verify them. Mr. Fugate says, "*If you are waiting to react to the aftermath of an event until you have a formal assessment, you are going to lose 12-to-24 hours... Perhaps we shouldn't be waiting for that... We looked at social media as the public telling us enough information to suggest this was worse than we thought and to make decisions to spend [taxpayer] money to get moving without waiting for formal request, without waiting for assessments, without waiting to know how bad because we needed to change that outcome.*" [14]. Trustworthiness and reliability of posted information are the biggest concerns in disaster response process. **Therefore, a key tradeoff exists when considering whether to incorporate unverified information from social media in additional to traditionally verified information when planning for disaster response.**

In this dissertation, we consider a response planning process that makes resource allocation decisions by potentially considering two separate streams of information: (i) unverified data describing demand that is not known with certainty, obtained from social media platforms, and (ii) verified data describing demand known with certainty, obtained from trusted traditional sources (i.e. on-the-ground assessment teams). **A new class of decision support models capable of considering these input streams for disaster response planning is developed.** Specifically, this dissertation investigates whether it is worthwhile to consider actionable information posted to social media platforms when developing disaster response plans.

The major questions investigated (i) what are the benefits and drawbacks associated with acting on social media information during disaster response, (ii) what are some alternative social data strategies an emergency manager could adopt in practice, and (iii) what are the trade-offs among those strategies. This dissertation extends previous models in the disaster relief routing literature that address uncertainty in demand. More broadly, this research contributes to the body of literature that addresses questions surrounding the usefulness of social data for response planning. An overview of each dissertation chapter is provided next.

In the second chapter, a new problem and framework for assessing the tradeoff described above in the context of the classic traveling salesman problem (TSP) is introduced. Two decision approaches are modeled, one that considers social data and one that does not, and their performance is compared based on length (a proxy for response time) of resulting tours. The alternative decision approaches are assessed through a set of case study instances with uniformly distributed request locations using an analytical study and a computational study. In the analytical study, competitive ratios for the alternative decision approaches are approximated without detailed specifications of request locations in each instance. In the computational study, competitive ratios for the alternative decision approaches are computed exactly.

The methodology introduced in the second chapter is extended in the third chapter to incorporate realistic problem characteristics such as multiple vehicles, service times at demand locations, and time limits. The problem under consideration is modeled as a Team Orienteering Problem (TOP) variant under uncertainty where the objective is to maximize demand served throughout the planning horizon. An Adaptive Large Neighborhood Search

(ALNS) method is developed to solve this problem, and a computational study is performed to compare the performance of alternative data strategies. A key feature of the chapter is a case study that is developed using real data (both traditional and social) from the 2010 Haiti earthquake. In this chapter, various new alternative strategies are introduced in order to represent a broad range of emergency manager preferences. The computational results provide managerial insight associated with incorporating social data in disaster relief tour planning.

In order to extend the static decision frameworks from chapters two and three into a dynamic decision framework, we introduce and define a new problem: the dynamic team orienteering problem (DTOP) in the fourth chapter. To solve the DTOP, we adapt the Multiple Plan Approach (MPA) first introduced in [15]. We introduce new benchmark instances, which are modified from the TOP benchmark instances, for DTOP. The performance of the MPA for DTOP is assessed using two methods. First, an online comparison for MPA is provided using a greedy algorithm in which a single plan is constructed using ALNS. Second, a reference offline algorithm is used to assess the performance of MPA. Average percentage deviation from offline solutions are computed in order to evaluate the performance of the MPA and greedy algorithms.

Bibliography

[1] EM-DAT: The OFDA/CRED International Disaster Database, "2013 disasters in numbers," tech. rep., Universit Catholique de Louvain, Brussels (Belgium), 2014. Available at www.cred.be/sites/default/files/Disasters-in-numbers-2013.pdf. Accessed May 19, 2014.

[2] M. Celik, O. Ergun, B. Johnson, P. Keskinocak, A. Lorca, P. Pekgun, and J. Swann, "Humanitarian logistics," P. Mirchandani, ed. INFORMS TutORials in Operations Research, vol. 9, pp. 18–49, 2012.

[3] A. Thomas, "Humanitarian logistics: enabling disaster response," tech. rep., Fritz Institute, 2003. Available at http://www.fritzinstitute.org/PDFs/WhitePaper/EnablingDisasterResponse.pdf. Accessed May 22, 2014.

[4] J. Heinzelman and C. Waters, "Crowdsourcing crisis information in disaster-affected haiti," tech. rep., US Institute of Peace, 2010. Available at www.usip.org/sites/default/files/SR252 - Crowdsourcing Crisis Information in Disaster-Affected Haiti.pdf. Accessed May 22, 2014.

[5] A. Sawchak, "Business continuity and social media how social tools are affecting organizational resilience." the SunGard Availability Services Blog, August 26 2013. Available at http://blog.sungardas.com/2013/08/business-continuity-and-social-media-how-social-tools-are-affecting-organizational-resilience/#sthash.HGnwMckH.dpbs.

[6] Y. S. Su, C. W. III, and Z. Thorkildsen, "Social media in the emergency management field," tech. rep., Report, CNA, 2010. Available at http://www.cna.org/sites/default/files/research/SocialMedia_EmergencyManagement.pdf.

[7] A. R. Cross, "Disaster services program review: fiscal year 2011." Available at http://www.redcross.org/images/MEDIA_CustomProductCatalog/m8540111_Disaster-Relief-Program-Review-2011.pdf, 2011.

[8] A. R. Cross, "More americans using mobile apps in emergencies.." Available at http://www.redcross.org/news/press-release/More-Americans-Using-Mobile-Apps-in-Emergencies, 2012.

[9] A. R. Cross, "Superstorm sandy one-year-update." Available at http://www.redcross.org/images/MEDIA_CustomProductCatalog/m23216738_Superstorm_Sandy_One-Year_Update.pdf, 2013.

[10] S. E. Cohe, "Sandy marked a shift for social media use in disasters," Emergency Management, 2013. Available at http://www.emergencymgmt.com/disaster/Sandy-Social-Media-Use-in-Disasters.html.

[11] C. for Excellence in Disaster Management & Humanitarian Assistance, "Lessons from typhoon haiyan (yolanda)," tech. rep., 456 Hornet Avenue, JBPHH, Hawaii, 2014. Available at http://hhi.harvard.edu/sites/default/files/publications/Lessons from Civil Military Disaster Management.pdf.

[12] H. dataset, Accessed June 2012. Available at http://community.ushahidi.com/research/datasets/.

[13] B. R. Lindsay, "Social media and disasters: current uses , future options, and policy considerations," tech. rep., Congressional Research Service, 7-5700, 2011. Available at http://www.infopuntveiligheid.nl/Infopuntdocumenten/R41987.pdf.

[14] P. Meier, "Big data for development: Challenges and opportunities," June 5 2012. Available at http://irevolution.net/2012/06/.

[15] R. W. Bent and P. V. Hentenryck, "Scenario-based planning for partially dynamic vehicle routing with stochastic customers," Operations Research, vol. 6, pp. 977–987, 2004.

Permissions

T & F Reference Number: P072716-03

7/27/2016

Emre Kirac
PhD Candidate and Graduate Teaching Assistant
Department of Industrial Engineering
University of Arkansas
emrekiracc@gmail.com

Dear Sir or Madame,

We are in receipt of your request to reproduce your article

> Emre Kirac, Ashlea Bennett Milburn, and Clarence L. Wardell (2015)
> The Traveling Salesman Problem with Imperfect Information
> with Application in Disaster Relief Tour Planning
> Special Issue: Scheduling and Logistics
> *IIE Transactions*, 47 (8): 783-799.
> DOI: 10.1080/0740817X.2014.976351

for use in your dissertation

You retain the right as author to post your Accepted Manuscript on your departmental or personal website with the following acknowledgment: "This is an Accepted Manuscript of an article published in *IIE Transactions* online on November 17, 2014, available online: http://www.tandfonline.com/doi/full/10.1080/0740817X.2014.976351

This permission is all for print and electronic editions.

We will be pleased to grant you permission free of charge on the condition that:

This permission is for non-exclusive English world rights. This permission does not cover any third party copyrighted work which may appear in the material requested.

Full acknowledgment must be included showing article title, author, and full Journal title, reprinted by permission of Institute of Industrial and Systems Engineers (IISE) (http://www.iienet2.org/Default.aspx).

Thank you very much for your interest in Taylor & Francis publications. Should you have any questions or require further assistance, please feel free to contact me directly.

Sincerely,

Mary Ann Muller

Permissions Coordinator

Telephone: 215.606.4334

E-mail: maryann.muller@taylorandfrancis.com

530 Walnut Street, Suite 850, Philadelphia, PA 19106 • Phone: 215-625-8900 • Fax: 215-207-0050
Web: www.tandfonline.com

UNIVERSITY OF ARKANSAS

College of Engineering
Department of Industrial Engineering

To: University of Arkansas Graduate School
From: Ashlea Bennett Milburn, Ph.D.

Date: July 27, 2016
Subject: Multi-author Documentation

This memorandum is to confirm that Emre Kirac is the first author of the following article and completed at least 51% of the work for the article.

"The Traveling Salesman Problem with Imperfect Information with Application in Disaster Relief Tour Planning"

4207 Bell Engineering Center • Fayetteville, Arkansas 72701 • 479-575-3156 • 479-575-8431 Fax

2. THE TRAVELING SALESMAN PROBLEM WITH IMPERFECT INFORMATION WITH APPLICATION IN DISASTER RELIEF TOUR PLANNING

2.1 Introduction

Over 224.4 million people worldwide were impacted by natural disasters each year during the period 2000-2011, causing over 90,000 annual deaths and more than $115 billion in economic damage [16]. Disasters leave impacted populations in need of medical attention, food, water and shelter, among other things. As a result, the rapid delivery of relief supplies and services can be critical in saving lives in their immediate aftermath. In order to plan how to deliver support effectively and efficiently, information regarding the location and need of the impacted population is required. Traditionally, this information becomes available as flyovers and other forms of on-the-ground assessment are completed. Critical hours pass while this information is collected and certain needs may not be identified in this manner. Consequently, many in the disaster response community have begun to explore ways to use information posted on social media platforms to identify a larger set of needs in a shorter amount of time following a disaster. Testifying before the U.S. Congress in May 2011, U.S. Federal Emergency Management Administrator Craig Fugate stated, "The sooner we are able to ascertain the on-the-ground reality of a situation, the better we will be able to coordinate our response effort in support of our citizens and first responders. Through the use of social media, we can disseminate important information to individuals and communities, while also receiving essential real-time updates from those with first-hand awareness" [19].

Across two surveys conducted in 2011 and 2012, the American Red Cross has shown individuals within the U.S. are increasingly using social media to post emergency-related information [11]. In the midst of Hurricane Sandy, individuals sent more than 20 million tweets about the storm from October 27 through November 1, 2012, more than twice the

[1] Emre Kirac, Ashlea Bennett Milburn, and Clarence L. Wardell, "*The traveling salesman problem with imperfect information with application in disaster relief tour planning*," IIE Transactions, vol. 47(8), pp. 783-799, 2015.

usage from the two previous days [6]. Similarly, following the Japanese earthquake and tsunami in March 2011, Twitter reported an event spike of up to 5,530 postings per second on their platform [22]. Furthermore, research has shown that some disaster-related posts contain information that is actionable and can be used in the decision making process [43].

If posts to social media by affected individuals describe their geographic location and what type of assistance they require, this information can be useful in the response planning process. Consider for example this post to the Ushahidi platform, a citizen event reporting platform, at 11:30 am on February 1, 2010, nineteen days after the magnitude 7.0 earthquake struck near Port-au-Prince, Haiti [41]:

> "Food water needed for group of 30 people (15 children). The address of the estate is #7 Marin 878 with the Blue Gate. At the gate they need to ask for Mondesire. This address is next to a collapsed church, and the name of the church is Pastor de ll'Fortune church. There is a lottery bank past the Church".

This request presents actionable data describing how many people need help (30 people including 15 children), what specifically they need (food and water) and where to reach them. If not uncovered by traditional means of assessment, response personnel could use information of this type to augment resource allocation decisions during response operations [24]. However, despite the existence of potentially life saving information on these platforms, many within the emergency response community remain skeptical over the reliability of such information and its usefulness for emergency response decision making [25, 36]. Needs communicated through social media platforms have initially not been verified. As such, there exists the possibility of inaccurate requests - either purposeful or unintended - that misrepresent location, type of need, or the level of need. During Hurricane Sandy, several individuals used the Twitter platform to post intentionally false information about storm damage [6].

When emergency managers dispatch valuable resources such as trucks, drivers and supplies to a location only to find no help is needed there (or more, less, or a different type), overall response time to verified need locations is delayed. Inaccurate information about needs has been identified as one of the primary impediments to the rapid delivery of goods immediately following a disaster [14]. While concerns over the accuracy of citizen reported information existed prior to the advent of social media, the volume and speed of information posted to popular platforms makes the challenge of separating true from false even more difficult. Consequently, as emergency managers consider whether to incorporate social media data in disaster planning efforts, a key tradeoff must be assessed. For instance, one could increase confidence in the accuracy of needs to which resources will be allocated by ignoring information discovered on social media, but there is potential to leave populations that have not yet been discovered through traditional means unassisted.

In this paper, we evaluate this tradeoff in the setting of relief supply delivery tour planning. We consider the problem of planning a vehicle tour to deliver relief supplies to all accurate need locations, regardless of how emergency managers discovered those needs. The unique challenge of this planning problem is that the accuracy of all needs cannot be known in advance because information from some sources is initially not verified. The objective of the planning problem is to minimize tour duration so that response time to accurate need locations is minimized. It is important to note the distinction that response time to *accurate* need locations is minimized; not response time to *all* need locations regardless of their accuracy.

The are four primary contributions of this work. First, a new problem framework is introduced that describes a formal method for quantitatively assessing the impact of including unverified information in disaster relief planning. While we present this framework in the context of the traveling salesman problem, the concepts can be employed for various classic decision problems with application in disaster relief planning (for example, the facility location problem for the placement of points of distribution). Second, a simple case study is included to demonstrate the usefulness of the framework. Because its primary purpose is illustrative, the scope of the case study is limited to disaster scenarios where request locations are uniformly distributed. Third, the vehicle routing models presented here extend those in the disaster relief routing literature that address uncertainty in demand, by considering the effects on decision making when two distinct classes of information are taken into account [14]. Finally, this work contributes more broadly to the body of literature that addresses questions around the usefulness of information provided through citizen event reporting [28]. Through the models and case study presented in this paper, we demonstrate the potential value of social media information when developing disaster relief routing plans for a stylized set of demand scenarios.

In a very practical way, by considering variations in report accuracy and quantity we address the tradeoffs that an emergency manager must make when choosing to either ignore or include data available through social media platforms when making resource allocation decisions. We save for future work variations in geographic distribution and constraints on the resource set, among other things. While many challenges exist with respect to extracting and verifying information from social media platforms, those issues are beyond the scope of this study. As such, we assume that the emergency manager uses one of many existing tools to extract information, to which they then assign *a priori* beliefs about the accuracy of the extracted set.

The remainder of this paper is organized as follows. In Section 2.2, we present formally the problem being addressed. In Section 2.3, we review the literature regarding disaster relief vehicle routing and the use of social media during disaster response. In Section 2.4, alternative decision approaches are described and in Section 2.5, we describe the models developed to assess the tradeoff associated with considering (or not considering) social media information in disaster relief tour planning. In Sections 2.6 and 2.7, we present the results of our computational experiments and give concluding remarks.

2.2 Problem statement

To introduce models aimed at assessing the tradeoff associated with incorporating (or not incorporating) social media information in disaster relief tour planning, the concepts of requests, their classification as accurate or inaccurate and their classification as verified or unverified are defined.

Definition. *A request is a declared need for relief supplies that specifies a location and type and quantity of good required.*

A request may be identified through social media information or other forms of assessment. A request specifies what type and quantity of good is needed (e.g., food or water) and where it is needed. The quantity required may be explicitly stated, or it may be derived if the number of people at the request location is specified. Note that a request could also identify a need for a service in applications in which service personnel (e.g., search and rescue teams, medical teams) must be routed. Without loss of generality, we choose to consider only goods here, as we focus on the delivery of supplies.

Definition. *An accurate request is a request specifying a correct location and correct type and quantity of good required.*

Definition. *An inaccurate request is a request in which at least one of the location, type of good and quantity of good specified is incorrect.*

All attributes of a request - location, type, quantity - must be correct in order for the request to be classified as accurate. If one or more of these attributes is incorrect, the request is classified as inaccurate.

Definition. *A verified request is a request that has been confirmed as an accurate request by a trusted source or method.*

Definition. *An unverified request is a request that has not been confirmed as an accurate request by a trusted source or method.*

Requests identified through trusted means (e.g., ground assessments conducted by qualified personnel) are considered verified immediately upon the request being revealed. Requests identified through untrusted means, such as social media, are considered unverified

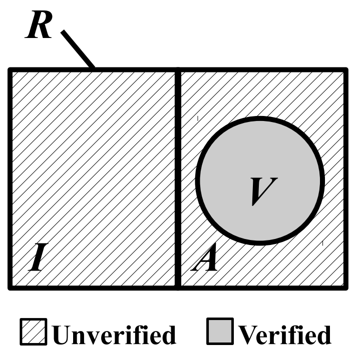


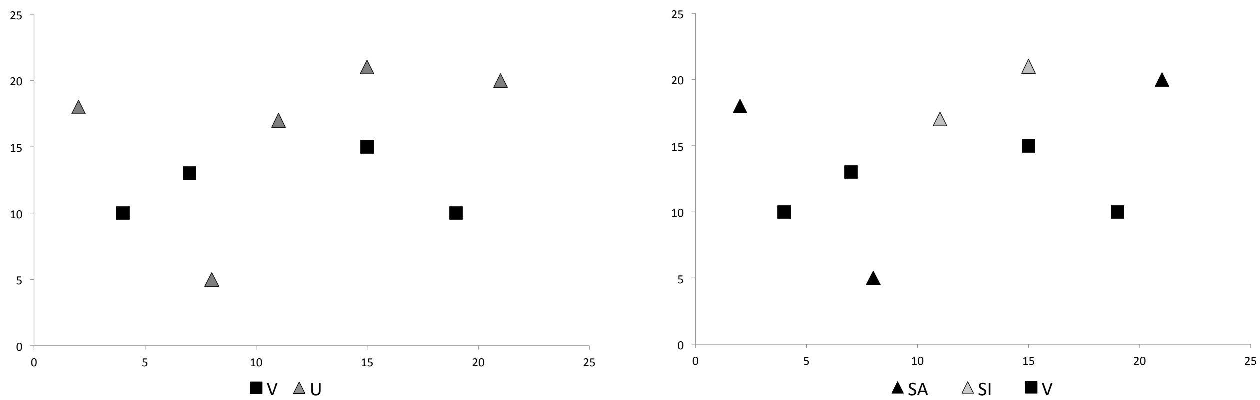
Figure 2.1: Request types

until their accuracy can be determined. Once an unverified request has been properly investigated by a trusted source or method, it is re-classified as either inaccurate, or both verified and accurate. Note that by definition, all verified requests are accurate. These concepts are illustrated in Figure 2.1, where the set of all requests is denoted $\mathcal{R}$. The set of accurate and inaccurate requests are denoted $\mathcal{I}$ and $\mathcal{A}$, respectively. The set of verified requests are denoted $\mathcal{V}$ and note that $\mathcal{V} \subseteq \mathcal{A}$ by definition. Finally, the set of unverified requests, denoted $\mathcal{U}$, is the area in the figure described by $\mathcal{R} \setminus \mathcal{V}$.

We define a new problem, disaster relief tour planning (DRTP), as follows. Two types of information are available at the time of planning: (1) the set of requests that have been verified prior to the time of planning, $\mathcal{V}$, and are thus by definition accurate, and (2) the set of unverified requests, $\mathcal{U}$, that may contain both accurate and inaccurate requests. Define the sets $\mathcal{U}_A$ and $\mathcal{U}_I$ to contain the accurate and inaccurate requests from $\mathcal{U}$, respectively; $U = \mathcal{U}_A \cup \mathcal{U}_I$. There is a complete undirected network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the node set and $\mathcal{A}$ is the arc set. Each node corresponds to a request location, therefore $\mathcal{N} = \mathcal{V} \cup \mathcal{U}$. A symmetric travel time matrix is defined on $\mathcal{G}$. In addition to travel times between locations, vehicle stopping times at request locations, s_i for $i \in \mathcal{N}$, may be considered. The stopping time at an accurate request location represents the time required to unload the items needed at the location and may depend on the type and quantity of items being delivered, among other things. The vehicle stopping time at an inaccurate request location represents the time required to confirm a request location as inaccurate upon arrival; no supplies are unloaded. The problem is to plan a vehicle tour to serve each **accurate request** (that is, each request in $\mathcal{V} \cup \mathcal{U}_A$) exactly once, with the objective of minimizing the total tour duration required to do so. Information regarding the accuracy of each request $r \in \mathcal{U}$ is available only in hindsight.

For simplicity of exposition, we assume throughout the remainder of this paper that a single vehicle with unlimited capacity will execute the tour and that stopping times at each request location are negligible. We assume also that no demand exists at inaccurate request locations. That is, we do not allow non-zero demand to exist at such locations in greater or lesser magnitudes than were stated. These assumptions enable the construction of preliminary models to begin assessing the tradeoff associated with using or not using unverified information in tour planning. The methods presented in this paper are generalizable to other routing problem variants.

Clearly an optimal solution to an instance of DRTP is to serve only the accurate requests (those in $\mathcal{V} \cup \mathcal{U}_A$) in the order specified by an optimal traveling salesman problem (TSP) tour through the associated locations. However, the unique challenge of DRTP is the accuracy of each request $r \in U$ is not known *a priori*, therefore the composition of $\mathcal{U}_A$ and $\mathcal{U}_I$ are also not known to the emergency manager at the time of planning. The information available to the emergency manager at the time of planning is illustrated in Figure 2.2a. The location associated with each request and the type and quantity of good requested are known, but requests can only be differentiated based on whether they are verified (represented by squares) or unverified (triangles). The verified requests are by definition accurate, but it is not known whether each unverified point is accurate. Once the accuracy of each unverified request is assessed, the information available in hindsight allows for the classification of all unverified points in Figure 2.2a as either accurate (black triangles) or inaccurate (light gray triangles) in Figure 2.2b. However, tour planning must be completed and the vehicle dispatched before the information available in hindsight, demonstrated in Figure 2.2b, is revealed.



(a) Information available when planning

(b) Information available in hindsight

Figure 2.2: Information available to emergency manager

In this paper we describe two decision approaches — new and traditional — that an emergency manager could use for solving DRTP. By comparing the durations of tours developed using these approaches to tours developed using a hindsight approach that benefits from knowing all information depicted in Figure 2.2b, we address the research question of assessing the tradeoffs between using only verified information, or also adding unverified requests in tour planning efforts.

2.3 Literature review

We review literature that describes how social media information has been used to support recent disaster relief efforts and documents continuing questions regarding its usefulness. To provide context for the models we present in Section 2.5, we also review literature on the impact of uncertainty in vehicle tour planning.

2.3.1 Role of social media in disaster relief

Social media platforms can serve as important tools for people trying to contact family and friends after a disaster and seek help from emergency response personnel. In a national survey administered by the American Red Cross to determine potential social media use during emergencies, 24 percent of respondents indicated they would use social media to let others know they are safe, and 20 percent of respondents indicated they would seek an online channel to communicate their need for assistance if they were unable to contact Emergency Medical Services [10]. Studies that investigated the role of social media in disaster relief efforts following earthquakes in Chile, Japan and Haiti demonstrated people used social media to share information about the disaster, express their opinions and feelings, and to help those in need of aid [10, 23, 31, 38].

Recent disasters have demonstrated that, in some instances, emergency managers have been able to improve their operational response by utilizing information posted to social media. McClendon and Robinson, in summarizing a report from UN-SPIDER, note Ushahidi data was used by the US Marines to coordinate locations and direct relief efforts and by the US Coast Guard to direct Coast Guard responders for search and rescue [32, 40]. Recent efforts have seen the United Nations Office for the Coordination of Humanitarian Affairs work closely with digital volunteer organizations to extract information from social media for use during crisis response [33].

During disasters, filtering and clarifying information content of messages posted to social media are challenging tasks [37]. However, these tasks are critical if actionable data to support the operational response is to be extracted from the large volume of information posted to social media following a disaster. Research is underway to develop tools for the efficient extraction of actionable data from social media platforms for use during emergencies. Examples include Twitcident, Tweak the Tweet and the SwiftRiver platforms [15, 37, 42].

In addition to filtering the high volume of data posted to social media to find the relevant data for use during emergency response, there is a concern over verifying the accuracy of data [34]. Scholars point out social media provides opportunities to determine the location and need of populations affected by disasters but they also highlight inaccurate or false information that has been posted to social media during disasters [32, 39, 38]. Consider an example from the 2010 Chilean earthquake: a report was sent, once through the Ushahidi platform and again through Twitter, describing the location of an English-speaking person supposedly trapped under a building [18]. The report was determined to be false after an emergency response team was dispatched to the location and determined the persons there were safe. Similar cases occured following the 2011 Japanese earthquake [38]. These examples highlight the concern of emergency managers regarding the use of social media information in disaster relief planning. Lindsay [31] pointed out inaccurate or false information could complicate awareness of a case and consequently delay or slow response efforts [31]. While these concerns are clearly valid, we attempt in this paper to demonstrate the potential benefits of incorporating social media information in disaster relief tour planning.

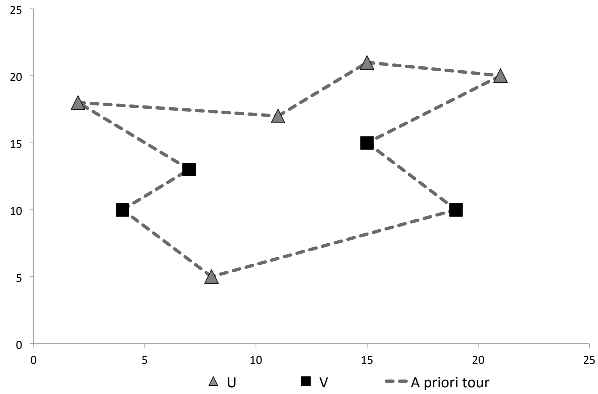
2.3.2 Vehicle routing with uncertain information

A comprehensive review of disaster relief routing problems is provided in de La Torre et al. [14]. In it, problems associated with the routing of goods to distribution points and individuals in areas impacted by disasters are described. Sources of uncertainty in these problems such as supply, demand, travel times, road network availability and vehicle/driver availability are discussed and the current state of incorporating them in operations research models is presented. The authors point out a need for "many years of potential future work" in disaster relief routing models [14].

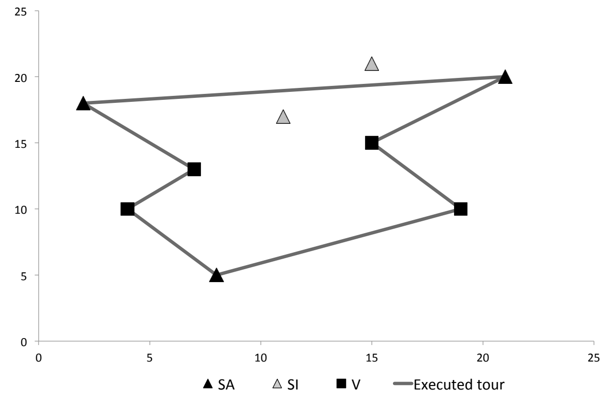
In this paper, we consider a disaster relief routing problem where there is uncertainty in demand. Specifically, two distinct classes of information are taken into account. One class of information is known with certainty *a priori*; this is the set of request locations identified through traditional means. Another class of information specifies request locations that cannot be known with certainty; this is the set of request locations identified through social media. A similar problem was first presented in Jaillet [26]. The paper defined the Probabilistic Traveling Salesman Problem (PTSP) in which only a random subset of customers need to be visited in any instance of the problem. The set of customers are classified into two sets: those requiring visits in every problem instance and those requiring a visit in each problem instance according to a known probability distribution. The objective of the PTSP is to design an *a priori* sequence of all customers to result in a tour of expected shortest total length when the customers not requiring a visit in a particular instance of the problem are skipped. The PTSP is a hard combinatorial problem and exact methods can solve only instances of limited size [5]. For example, Laporte et al. (1994) develop an exact approach for solving PTSP instances containing up to 50 customers [30].

For the model presented in this paper, the set of verified requests can be conceptualized as those requiring visits in every problem instance in the PTSP. Additionally, the set of unverified requests can be conceptualized as those not requiring a visit in every problem instance (it depends on the demand realization). However, the problem under consideration is different from the PTSP in a number of ways. First, it is not assumed probabilistic information regarding the accuracy of an unverified request communicated through social media is available. Thus, there is not a known distribution describing the probability each request will require a visit in each problem instance. Second, in this paper we do not seek to find an *a priori* sequence that minimizes the expected length of a tour visiting all accurate request locations in an instance; this is not possible without the aforementioned probabilities. Instead, we seek the sequence minimizing the actual length of a tour that visits all accurate need locations in a realized scenario, and possibly some inaccurate need locations as well. Finally, an inherent assumption for PTSP is customers not requiring visits can be skipped. Skipping customers is not possible in the problem we study, because it is not clear which need locations communicated through social media do not require visits when the tour begins. This information does not become available until the vehicle arrives at the need location and it is discovered no demand is there, or until other resources are allocated to verifying the information. In other words, information regarding the existence of each customer in this instance is revealed at a later point in time than in PTSP.

Figure 2.3 is included to illustrate an example solution to PTSP for an instance with four verified requests and five unverified requests; the latter each having a probability $p = 0.5$ of needing a visit in a particular demand realization. Figure 2.3a illustrates the *a priori* tour



(a) *A priori* tour



(b) Tour executed after scenario revealed

Figure 2.3: Example PTSP solution

minimizing expected total length and Figure 2.3b illustrates the tour that will be executed if two of the unverified request locations are revealed as inaccurate before the tour begins. Later in Section 2.4, Figure 2.4 illustrates how solutions to an instance of DRTP containing the same customer locations differ. We note again that probabilistic information and the realization of each unverified request as accurate or inaccurate prior to tour commencement are not available for instances of DRTP.

Many other routing models that incorporate demand uncertainty have been published in the literature (e.g., [3, 21]). However, because these papers also assume known probability distributions describing demand at customer locations, we do not review them in detail.

2.4 Alternative decision approaches for DRTP

Three decision approaches for DRTP are described. The ensuing discussion makes use of the following notation:

$TSP(\mathcal{N})$: the optimal traveling salesman tour through locations of requests in $\mathcal{N}$,

$L^*(\mathcal{N})$: the length of $TSP(\mathcal{N})$.

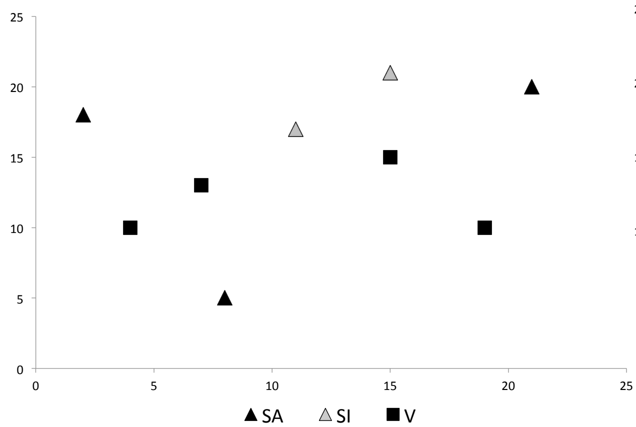
First let us consider the hindsight approach, denoted HIND. This approach, not available to the emergency manager at the time of planning, considers verified requests and unverified

Approach	Response time	Description
HIND	$L^*(\mathcal{V} \cup \mathcal{U}_A)$	Plan a single tour visiting verified requests ($\mathcal{V}$) and unverified requests determined as accurate ($\mathcal{U}_A$) in hindsight
NEW	$L^*(\mathcal{V} \cup \mathcal{U})$	Plan a single tour visiting verified requests ($\mathcal{V}$) and all unverified requests ($\mathcal{U}$); depart inaccurate requests immediately upon arrival
TRAD	$L^*(\mathcal{V}) + L^*(\mathcal{U}_A)$	Plan tour visiting verified requests ($\mathcal{V}$); prior to its completion, each request in $\mathcal{U}$ is classified as accurate ($\mathcal{U}_A$) or inaccurate ($\mathcal{U}_I$); plan second tour through $\mathcal{U}_A$ to begin after first tour ends

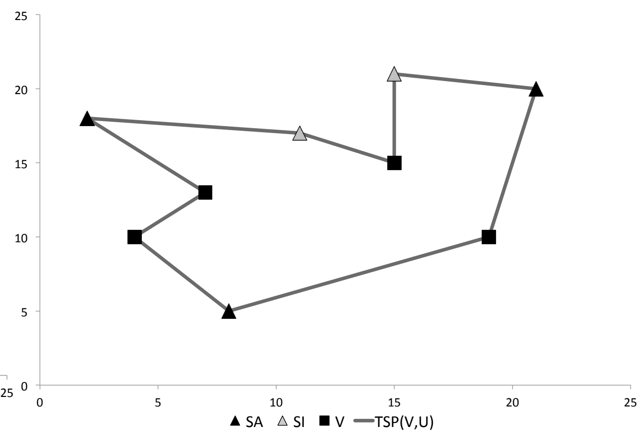
Table 2.1: Decision approaches for DRTP

requests that have, since the time of planning, been confirmed as accurate. Thus, $TSP(\mathcal{V} \cup \mathcal{U}_A)$ is identified and total response time is $L^*(\mathcal{V} \cup \mathcal{U}_A)$. Consider next the "new" approach, denoted NEW. When solving an instance of DRTP using NEW, all verified and unverified requests are considered and the tour $TSP(\mathcal{V} \cup \mathcal{U})$ is identified. Here we assume that if a vehicle reaches an inaccurate request during the execution of $TSP(\mathcal{V} \cup \mathcal{U})$, they immediately confirm the request as inaccurate and depart for the next location on the tour with no delay. Thus, the total response time associated with the tour developed using NEW is $L^*(\mathcal{V} \cup \mathcal{U})$. Consider finally the "traditional" approach, denoted TRAD. When solving an instance of DRTP, this individual considers only the verified requests, identifying $TSP(\mathcal{V})$. Here we assume that while $TSP(\mathcal{V})$ is being executed, the accuracy of each request in $\mathcal{U}$ is concurrently assessed using other resources. Therefore, before the vehicle completes $TSP(\mathcal{V})$, the emergency manager has new information characterizing each request in $\mathcal{U}$ as accurate or inaccurate and plans a second tour $TSP(\mathcal{U}_A)$ that commences as soon as $TSP(\mathcal{V})$ is completed and the vehicle becomes available. Thus the traditional approach will require at minimum two consecutive tours to service identified needs. The total response time associated with this approach is $L^*(\mathcal{V}) + L^*(\mathcal{U}_A)$. These approaches are summarized in Table 2.1.

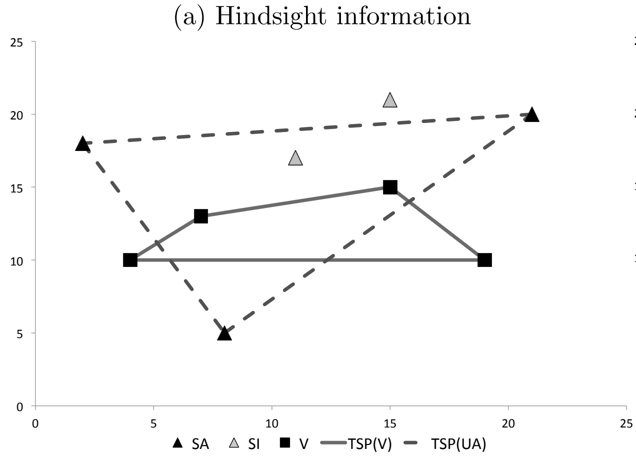
The alternative decision approaches for DRTP are illustrated using an example instance in Figure 2.4. The demand scenario from Figure 2.1 is included again in Figure 2.4a for clarity. Recall all triangle-shaped points represent unverified requests, but are confirmed as either accurate ($\mathcal{U}_A$, black) or inaccurate ($\mathcal{U}_I$, light gray) in hindsight. Figure 2.4b depicts the route developed using NEW. The tour includes visits to all locations, including those that are unverified, so then necessarily also some that are inaccurate. Figure 2.4c depicts the route developed using TRAD. All verified points are visited first on the tour depicted by a solid line and a second tour depicted by a dotted line visits all unverified points confirmed as accurate while the first tour was executed. In TRAD, no inaccurate points are visited, but two tours are required in order to reach all accurate points. Finally, Figure 2.4d depicts the route developed in hindsight with full information regarding the accuracy of each unverified request. This is the shortest possible tour visiting all accurate points, with length 60.0. In this example, the NEW tour has length 65.6 and the TRAD tours together have length 95 (41.7+53.3).



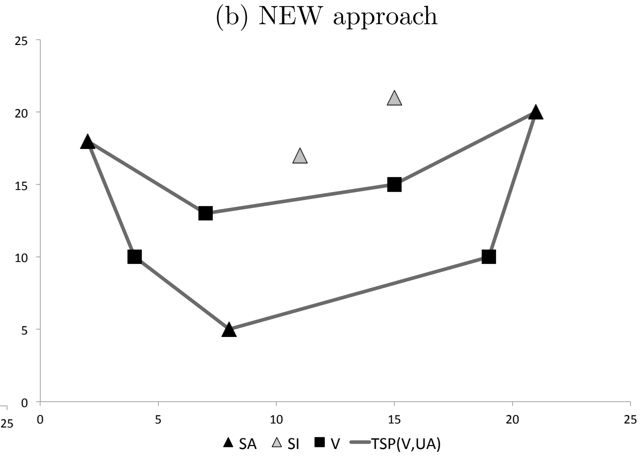
(a) Hindsight information



(b) NEW approach



(c) Route developed using TRAD



(d) Route developed using HIND

Figure 2.4: Routes resulting from alternative decision approaches

2.4.1 Observations

The performance of alternative decision approaches in the presence of various decision scenarios is discussed.

Observation. *When there are no unverified requests, the performance of HIND, NEW and TRAD are equivalent.*

When there are no unverified requests ($\mathcal{U}$ is empty), all three decision approaches will by definition produce $TSP(\mathcal{V})$ because $\mathcal{V} \cup \mathcal{U} = \mathcal{V}$ and $\mathcal{V} \cup \mathcal{U}_A = \mathcal{V}$.

Observation. *When there are both verified and unverified requests but all unverified requests are accurate, the performance of HIND and NEW is equivalent.*

When the set of unverified requests is not empty but is comprised solely of accurate requests ($U = \mathcal{U}_A$), the performance of HIND and NEW are equivalent, because $TSP(\mathcal{V} \cup \mathcal{U}_A) = TSP(\mathcal{V} \cup \mathcal{U})$.

Observation. *When there are both verified and unverified requests but all unverified requests are inaccurate, the performance of HIND and TRAD is equivalent. Furthermore, both HIND and TRAD perform strictly better than NEW.*

When the set of unverified requests is not empty but is comprised solely of inaccurate requests ($U = \mathcal{U}_I$), the performance of HIND and TRAD are equivalent, both producing $TSP(\mathcal{V})$. A secondary tour is not needed by TRAD in this case, as all accurate requests have been visited once $TSP(\mathcal{V})$ is completed. These approaches produce tours having duration $L^*(\mathcal{V})$, which is strictly less than $L^*(\mathcal{V} \cup \mathcal{U})$, the duration of the tour produced by NEW.

Combining insights from the above observations, it is clear the hindsight approach always produces tours with duration no greater than tours produced by NEW or TRAD. However, it is not always possible to say whether NEW is better than TRAD and vice versa. The relative performance of NEW and TRAD will depend on at least two dimensions: (1) the proportion of requests that are verified versus unverified, and (2) the proportion of unverified requests that are accurate versus inaccurate. Define ρ as the proportion of total requests that are unverified and λ as the proportion of unverified requests that are accurate. Suppose unverified requests exists ($\rho > 0$). Then, the relative performance of NEW should be better than TRAD when most of these requests are accurate ($\lambda \approx 1$), while the relative performance of TRAD should be better than NEW when most of these are inaccurate ($\lambda \approx 0$). These differences should be more pronounced when ρ is large, with unverified requests representing a larger proportion of total requests. It is not immediately apparent which approach is best for all combinations of ρ and λ. In Section 2.5, a methodology to assess the performance of TRAD and NEW across various uniformly distributed demand scenarios is presented.

2.5 Methodology

In this research we demonstrate models that can be used to assess the tradeoff associated with using social media information in tour planning efforts by comparing the response times when NEW, TRAD and HIND are used to solve DRTP for a simple set of case study instances. Specifically, instances with uniformly distributed request locations are considered. Competitive ratio analysis, using HIND as an optimal offline algorithm for DRTP, is performed. Let $R^H(I)$ be the response time when algorithm H is used to solve instance I of DRTP. We analyze the ratios $\frac{R^{TRAD}(I)}{R^{HIND}(I)}$ and $\frac{R^{NEW}(I)}{R^{HIND}(I)}$ for a variety of instances that differ with respect to the proportion of request locations that are unverified and the proportion of unverified request locations that are accurate.

Two methods are used to make this comparison. First, an analytical study is conducted, in which competitive ratios for TRAD and NEW may be approximated for various instance styles, without detailed specifications of request locations and associated routes in each instance. The analytical study considers variations in unverified and accurate request proportions, but does not consider variations in service region size or total volume of requests because the competitive ratios do not directly depend on these parameters. Second, a computational study is executed, in which instances of DRTP are solved exactly using TRAD, NEW and HIND. The computational study, in addition to considering variations in proportions of unverified and accurate requests, considers variations in region size and request volume. These studies are presented in Sections 2.5.1 and 2.5.2, respectively.

2.5.1 Analytical study

Methods for approximating optimal TSP tour lengths without detailed specifications of the points to be visited are found in the literature (e.g., [2, 4, 7, 12, 13, 17, 29, 35]). These approaches can be useful in strategic and tactical planning, when some information about the cost of operational routes is needed but an exact specification of the sequence of points to be visited is not required. For example, Kwon et al. (1995) develop regression and neural network models to estimate tours for uniformly distributed nodes in rectangular regions [29]. Additionally, routing approximations for different location distributions and geographies have been developed. For example, Daganzo (1984) provides a routing approximation for uniformly distributed nodes in zones of irregular shapes and then the analysis is extended for non-uniformly distributed nodes in a later paper [12, 13]. Blumenfeld and Beckman (1985) generalize tour length approximations according to varying customer location densities and clusters [4]. Both Ong and Huang (1989) and Chien (1992) present computational experiments to validate various tour length estimators [7, 35]. Figliozzi (2009) provides tour length approximations for the Vehicle Routing Problem with Time Windows [17].

Let node set N define an instance of the TSP on a complete undirected graph. A well-known study in the traveling salesman problem approximation literature expresses the Euclidean distance of the shortest tour visiting N randomly and uniformly distributed customer locations in an area of size A as [2]:

$$L^*(N) = k\sqrt{NA},\ as\ N \to \infty \tag{2.1}$$

where k is a constant, estimated as 0.7124 [27]. This estimator is selected for this study due to its simplicity and its applicability for the case study presented, which considers only uniformly distributed nodes. It should be noted that approximations produced by this TSP length estimator are most accurate when N is large [4, 7, 13].

This estimator is applied to approximate response time for solutions to DRTP developed by NEW, TRAD and HIND approaches. In this analysis, it is assumed the request locations in subsets $\mathcal{V}$, $\mathcal{U}$, $\mathcal{U}_A$ and $\mathcal{U}_I$ are uniformly distributed in a regularly shaped region of size A. Let N denote the total number of request locations in an instance of DRTP. The number of points in each subset are $|\mathcal{V}| = (1-\rho)N$, $|\mathcal{U}| = \rho N$, $|\mathcal{U}_\mathcal{A}| = \lambda\rho N$, and $|\mathcal{U}_\mathcal{I}| = (1-\lambda)\rho N$. Then, response times $R^H(I)$ associated with solutions developed for the decision approaches under consideration can be approximated as:

$$R^{TRAD}(I) = L^*(\mathcal{V}) + L^*(\mathcal{U}_A) = k\sqrt{(1-\rho)NA} + k\sqrt{\lambda\rho NA}, \tag{2.2}$$

$$R^{NEW}(I) = L^*(\mathcal{V} \cup \mathcal{U}) = k\sqrt{NA}, \tag{2.3}$$

$$R^{HIND}(I) = L^*(\mathcal{V} \cup \mathcal{U}_A) = k\sqrt{[(1-\rho)N + \lambda\rho N]\, A}. \tag{2.4}$$

As mentioned previously, the approximations in Equations (2.2)-(2.4) are most accurate when the number of points in each subset is large. For some combinations of parameters (N,ρ,λ) this may not hold. For example, even with a relatively large $N = 1000$, the number of accurate unverified points in $\mathcal{U}_\mathcal{A}$ is only 10 when $\rho = \lambda = 0.1$. Thus, the first term in Equation (2.2) estimates tour length for 990 points in this example, while the second term estimates tour length for only 10 points. The potential impact of this on the results we present is discussed along with the results in Section 2.6.

Using these approximations, the competitive ratios for TRAD and NEW can be computed as follows:

$$\frac{R^{TRAD}}{R^{HIND}} = \frac{\sqrt{(1-\rho)N} + \sqrt{\lambda\rho N}}{\sqrt{(1-\rho)N + \lambda\rho N}} = \frac{\sqrt{1-\rho} + \sqrt{\lambda\rho}}{\sqrt{1-\rho+\lambda\rho}}, \tag{2.5}$$

$$\frac{R^{NEW}}{R^{HIND}} = \frac{\sqrt{N}}{\sqrt{(1-\rho)N + \lambda\rho N}} = \frac{1}{\sqrt{1-\rho+\lambda\rho}}. \tag{2.6}$$

Note that these ratios are independent of N and defined for every combination of parameter values $\rho \in [0,1]$, $\lambda \in [0,1]$, except the following:

- $\rho = 0$: There are no unverified requests, so λ, the proportion of unverified requests that are accurate, is not defined.
- $\rho = 1$ and $\lambda = 0$: All requests are unverified ($\rho = 1$) and none are accurate ($\lambda = 0$). Because there are no accurate points to be visited, there is no planning problem to solve.

The three observations presented in Section 2.4.1 hold for the ratios described in Equations (2.5)-(2.6). The first observation states when there are no unverified requests, the performance of NEW, TRAD and HIND are equivalent. Equations (2.5) and (2.6) are not defined in this case. The second observation states when there are both verified and unverified requests but all are accurate, the performance of HIND and NEW are equivalent. When $\lambda = 1$ and $\rho \in (0,1)$ the competitive ratio described in Equation (2.6) is 1, giving equivalence of NEW and HIND. The last observation states when there are both verified and unverified requests but all unverified requests are inaccurate, the performance of HIND and TRAD are equivalent and strictly better than NEW. When $\rho \in (0,1)$ and $\lambda = 0$, the competitive ratio described in Equation (2.5) is 1, giving equivalence of NEW and TRAD. Furthermore, when $\lambda = 0$ Equation (2.6) is $\frac{1}{\sqrt{1-\rho}}$ which is strictly greater than 1 when $\rho \in (0,1)$, demonstrating TRAD and HIND perform strictly better than NEW. A final observation is that when all requests are unverified ($\rho = 1$) and all unverified requests are accurate ($\lambda = 1$), all three decision approaches are equivalent, with Equations (2.5) and (2.6) both reducing to 1.

2.5.2 Computational study

A computational study is now presented that requires the detailed specification of all requests, their locations, and their classification as verified or unverified and accurate or inaccurate. Using this information, Concorde software is used to find the optimal TSP tours through the set of request locations considered by each alternative decision approach [9]. Response times for tours are calculated exactly and competitive ratios are computed. This study is conducted for two primary reasons. First, the tour length approximation employed in the analytical study is known to be less reliable when the number of points to be visited is small. The computational study does not have this limitation, as tour lengths are computed exactly. Therefore, the computational study is employed for instances with small numbers of points to complement the analytical study. Second, the computational study replicates the actual planning process that follows a disaster event once demand information becomes available. That is, a decision approach (e.g., TRAD or NEW) must be applied to develop detailed route specifications for a set of specific requests under consideration.

2.6 Experiments and Results

The performance of NEW and TRAD are assessed using the analytical and computational study methods described in Sections 2.5.1 and 2.5.2. Only two parameters are required in the analytical study, ρ and λ. Competitive ratios for TRAD and NEW are approximated using Equations (2.5) and (2.6) for all combinations of $\rho \in [0.1, 0.2, \ldots, 0.9]$ and $\lambda \in [0, 0.1, 0.2, \ldots, 1.0]$. The parameter value $\rho = 0$ is excluded because DRTP is not defined when all requests are verified. The parameter value $\rho = 1.0$ is excluded because when all requests are unverified, the only type of information to consider is social media information.

In the computational study, actual request locations are required so test instances are selected from the TSP literature. The instances selected have uniformly distributed customer locations and represent a variety of region sizes and number of total requests. Two region sizes are represented, with regions less than 2500 square kilometers begin classified as small regions (S) and regions between 2500 and 5000 square kilometers being classified as large (L). While the sizes of these regions are rather large for practical purposes (the state of Rhode Island is approximately 1500 square kilometers, for example), travel times between pairs of locations can be scaled down without loss of generality in these results. These region sizes were selected due to their availability in the literature and in order to demonstrate the impact of changing the average separation between request locations (in other words, the density of request locations). Three levels of number of total requests are considered, with low (L) indicating less than 25 requests, medium (M) indicating 25 to 44 requests, and high (H) indicating 45 or more requests. We choose instances from the literature with these numbers of requests in order to complement the analytical study results, which may be less reliable when the number of points to be visited is small. The instances taken from the literature are summarized in Table 2.2. Each is denoted using two letters XX, where the first letter denotes region size and the second denotes request volume. For each instance, we do not consider the amount of goods requested as we are interested in modeling travel time in this initial study. Thus, zero service time at request locations and unlimited vehicle capacity are assumed.

Instance	n	Name in lit.	Reference
SL	21	n20w200.005	[20]
SM	41	n40w80.004	[44]
SH	61	n60w80.001	[44]
LL	22	E-n22-k4	[8]
LM	40	P-n40-k5	[1]
LH	60	P-n60-k10	[1]

Table 2.2: Instances

A total of 99 variants of each instance selected from the literature are included in the computational study; one for each combination of $\rho \in [0.1, 0.2, \dots 0.9]$ and $\lambda \in [0, 0.1, 0.2, \dots 1.0]$. These are denoted $XX_{\rho\lambda}$. Each instance variant is created in the following manner. Each of the request locations included in the original dataset XX is randomly designated as unverified according to a Bernoulli distribution with parameter ρ. Each resulting unverified request location is then randomly designated as accurate according to a Bernoulli distribution with parameter λ. Ten random replicates of each of the 99 variants for each of the 6 instances selected from the literature are generated. For instance XX, the same request locations will comprise each replicate of each variant $XX_{\rho\lambda}$, but the designation of specific request locations as verified or unverified and accurate or inaccurate may change. A study of density is an indirect consideration by varying the instance parameters such as region size, request volume, and (ρ, λ). For example, instances with small regions have higher request density than their large region counterparts. Similarly, instances with small values of (ρ, λ) have lower unverified and accurate request density than their counterparts with higher values of these parameters.

Results from the analytical and computational studies are presented in Sections 2.6.1 and 2.6.2, respectively.

2.6.1 Analytical results

The results of the analytical study described in Section 2.5.1 are illustrated in Figure 2.5. The parameters ρ and λ are found along the two horizontal axes and the competitive ratio is found along the vertical axis. The proportion of total requests that are unverified and the proportion of unverified requests that are accurate increase from the center of the figure moving outwards. The dark gray plane represents NEW and the light gray plane represents TRAD.

For most of the (ρ, λ) combinations included in the analytical study, NEW has a lower competitive ratio than TRAD, indicating better performance. For example, when 10 percent of requests are unverified and 40 percent of those are accurate, the competitive ratio for NEW (illustrated by the dark gray plane) is 1.03 and the competitive ratio for TRAD is 1.18 (light gray plane). Because these precise values are difficult to discern from the three dimensional graph, they are included in the Appendix in Table 2.5. In general, as the accuracy of

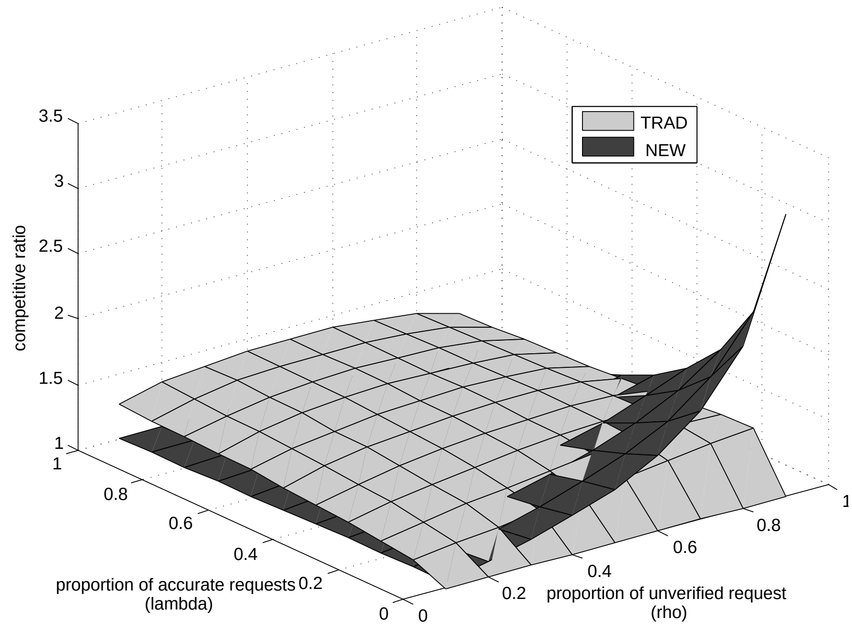


Figure 2.5: Competitive ratios approximated by analytical study

unverified requests, λ, increases, the performance of NEW improves. This result is intuitive. All unverified request locations are included in tours produced by NEW. As more of these become accurate, the tour produced by NEW will more closely resemble the tour produced by the optimal offline approach HIND.

When the number of unverified requests is high and many are inaccurate, TRAD is preferred over NEW. This occurs, for example, when 90 percent of requests are unverified and only 20 percent of those are accurate, resulting in competitive ratios for TRAD and NEW of 1.40 and 1.89, respectively. This result is also intuitive. When NEW includes large numbers of unverified requests in planned tours and many of those unverified requests are inaccurate, excess travel is incurred to inaccurate request locations that are not included in tours produced by HIND and TRAD.

The results in Figure 5 should be interpreted with caution due to potential inaccuracy of the route length approximations for tours that visit small numbers of points. As stated previously, the approximation in Equation 2.1 is most accurate when N, the number of points to be visited, is large. However, there is no commonly agreed upon standard in the literature regarding how large N should be. Consider then, for sake of exposition, a "relatively big" value of N=100. Furthermore, let N also denote the total number of requests in an instance of DRTP. The approximation for NEW is for a tour visiting all 100 points regardless of the parameter values ρ and λ. Therefore, there is no concern over interpreting the NEW approximations as long as N is "big enough." However, the approximation for TRAD requires taking the summation of estimated lengths for two separate tours. One of these tours visits a small number of points when both ρ and λ are small, and both tours visit a small number of points when ρ is large and λ is small. For example, if $(\rho, \lambda) = (0.2, 0.2)$, then the tours visit 80 points and 4 points, respectively, and if $(\rho, \lambda) = (0.9, 0.2)$, then the tours visit 10 and 18 points, respectively. Therefore, the regions in Figure 5 with low λ values should be interpreted with caution. However, it is interesting to note the referred-to region is the only one in which a preference for TRAD is indicated. Additionally, computational study results, which are not limited by an assumption regarding the size of N, also support a preference for TRAD in these regions (see Figure 2.7 and the discussion in Section 2.6.2). Of course, all regions of Figure 5 are called into question if N is not "big enough", such that even NEW approximations visiting all N points are unreliable.

Table 2.3 and Figure 2.6 provide summary information for the performance of NEW and TRAD according to the analytical study. Table 2.3 indicates NEW was the preferred approach in 76 out of the 99 combinations of (ρ, λ) tested and TRAD was the preferred approach in the remaining 23. On average, NEW and TRAD produce solutions with response times that are 22% and 30% higher than response times of solutions produced by the reference approach HIND. Figure 2.6 presents a box plot of competitive ratios associated with solutions produced by NEW and TRAD. The range of competitive ratios associated with solutions produced by TRAD is 1 to 1.5 while the range is 1 to 3.2 for NEW. This indicates the worst-case performance of NEW is worse than TRAD, producing tours requiring up to triple the response time of the reference offline approach HIND. An emergency manager wishing to minimize the worst-case performance may prefer TRAD, which will only produce tours that require up to 50% greater response time than the reference offline approach in the worst case.

Decision approach	Avg % deviation from HIND	Num. scenarios in which approach is preferred
NEW	22.22%	76
TRAD	30.41%	23

Table 2.3: Analytical results summary

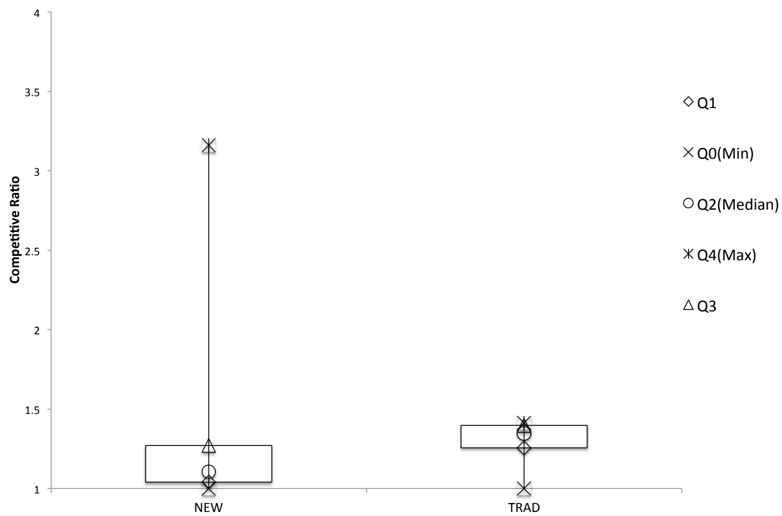


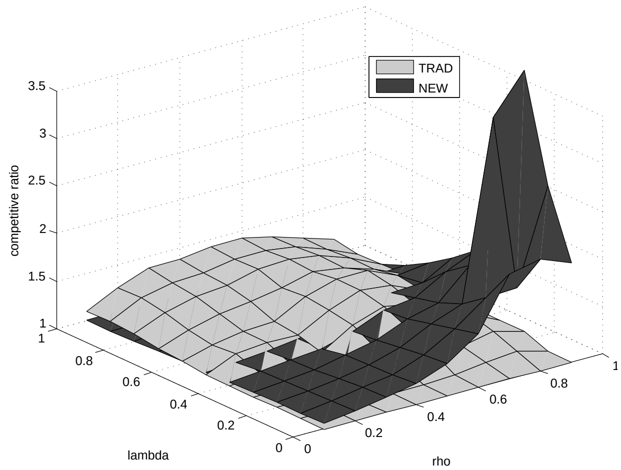
Figure 2.6: Analytical results box plot

In contrast, an emergency manager wishing to select the approach that will perform better for the majority of scenarios may prefer NEW. Seventy-five percent of the solutions produced by NEW had competitive ratios less than 1.3, while only twenty-five percent of the solutions produced by TRAD had competitive ratios of 1.3 or lower. Similar caution expressed with respect to Figure 2.5 for small values of N should also exhibited when interpreting the results in Table 2.3 and Figure 2.6.

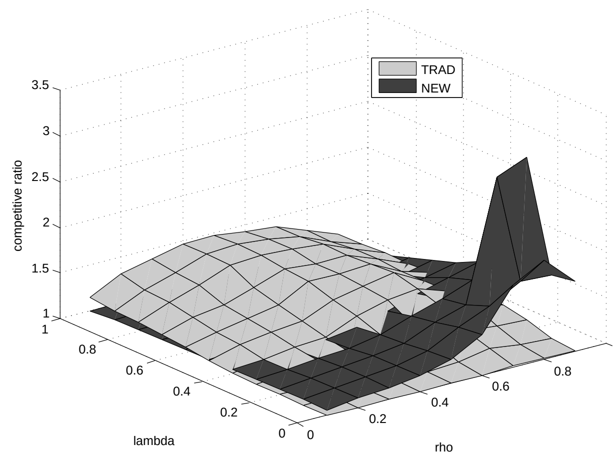
2.6.2 Computational results

Results of the computational study described in Section 2.5.2 are summarized in Figure 2.7. Subfigures 2.7a through 2.7f present results for each instance taken from the literature. Each point on each plane represents the average competitive ratio of the indicated approach over 10 replicates of the given instance variant $XX_{\rho\lambda}$.

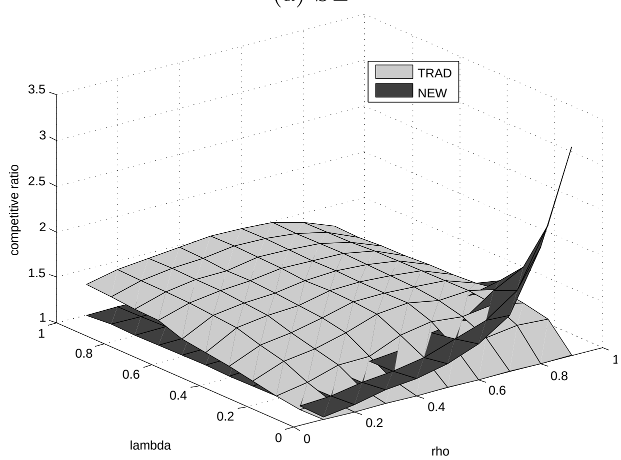
As seen in the analytical study, the performance of NEW generally improves as λ in-



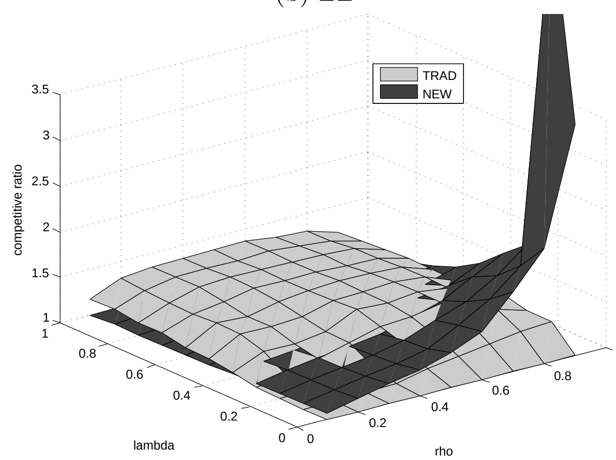
(a) SL



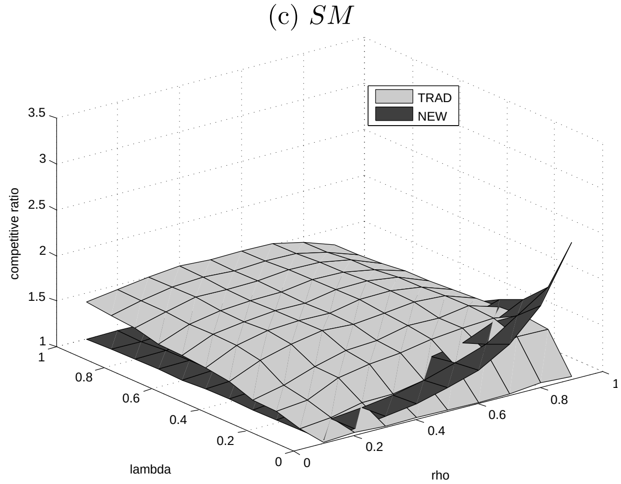
(b) LL



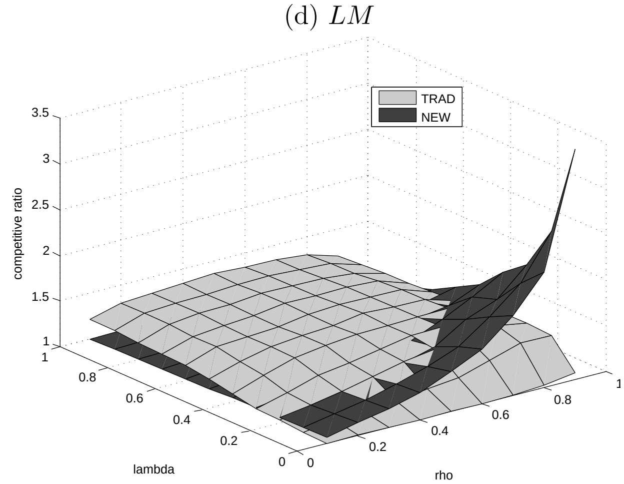
(c) SM



(d) LM



(e) SH



(f) LH

Figure 2.7: Competitive ratios approximated by computational study

creases. The impact of region size and request volume on decision approach performance can also be assessed. Moving from left to right within a row of Figure 2.7, the impact of increasing the region size can be seen. Incidentally, moving from left to right within a row also demonstrates the impact of decreasing request density, as the same number of requests is distributed across a larger region. In general, TRAD is preferred in a larger number of instance variants when the region size increases from small to large. For example, when request volume is high and region size is small, TRAD is preferred for all variants $SH_{\rho\lambda}$ where $\lambda = 0.1$ but not for all variants where $\lambda = 0.2$ (Figure 2.7e). When region size increases to large, TRAD is preferred for all variants of $LH_{\rho\lambda}$ where $\lambda \leq 0.2$, and for some values of ρ when $\lambda = 0.3$ or 0.4 (Figure 2.7f). This may be because as region size increases, average distance between points increases, and thus the relative penalty associated with visiting inaccurate unverified locations as prescribed by NEW also increases.

The impact of increasing the request volume (and density) can be seen moving from top to bottom within columns of Figure 2.7. In general, NEW is preferred in a larger number of instance variants when request volume increases from low to high. For example, when region size is large and request volume is low, NEW is preferred only when $\lambda \geq 0.4$ for most values of ρ (Figure 2.7b). When request volume increases to high, NEW is preferred when $\lambda \geq 0.2$ for most values of ρ. Therefore, a lower threshold of accuracy is required for NEW to be preferred when request volume is higher. Note the images corresponding to high request volumes, Figures 2.7e and 2.7f, are very similar in appearance to Figure 2.5. A possible explanation is as request volume increases, computational study results converge asymptotically to analytical study estimates, because the approximation in Equation (2.1) on which the analytical study is based is most accurate when the number of points is large.

Summary results for the computational study are presented in Table 2.4. The right-most column presents the number of replicates in which each approach was preferred. For each instance, 990 replicates were considered - 10 replicates of each of 99 (ρ, λ) combination variants. However, the two numbers in the right-most column for a given instance will not always sum to 990 because for some replicates, TRAD and NEW achieve equal competitive ratios. The majority preferred approach in each instance is indicated in bold. When request volume is low, the number of replicates in which TRAD and NEW are preferred are approximately equal, with NEW being preferred in 416 out of 990 replicates of SL and in 469 out of 990 replicates of LL. For these instances, the average deviation of response times associated with solutions produced by TRAD from the reference offline approach is better than that of NEW, with TRAD exhibiting 18.1% and 20.4% deviation from the offline approach while NEW exhibits 27.8% and 21.5% deviation. However, as request volume increases, the average deviation of response times associated with solutions produced by NEW from the reference offline approach is better than that of TRAD. The number of replicates in which NEW is preferred also increases. For the medium and high request volume instances, NEW is preferred in 2764 out of 3960 replicates. The preference for NEW over TRAD is most distinct when region size is small and request volume is high. In this case, NEW is preferred in 791 out of 990 replicates and exhibits an average deviation from HIND of 16.2% while TRAD exhibits a deviation of 39.0%. A preferred approach is not indicated in the table for instance LM because TRAD has better performance according to average deviation from the offline approach while NEW has better performance according to the number of replicates in which the approach is preferred.

Figure 2.8 presents box plots for competitive ratios associated with solutions produced by NEW and TRAD for the six instances considered. For any given instance, the range of competitive ratios associated with solutions produced by NEW is always wider than that of TRAD, indicating that TRAD has better worst-case performance. For example, the range associated with NEW for instance SM in Figure 2.8c is 1 to 3.4, while the range associated with TRAD is 1 to 1.6. However, as seen in the analytical study and in Table 2.4, the average-case performance is usually better for NEW. Take for example instance SM in Figure 2.8c, where 75% of solutions produced by NEW had competitive ratios less than 1.2, while only 25% of the solutions produced by TRAD had competitive ratios of 1.35 or lower. Trends associated with region size and request volume can be observed in the box plots of Figure 2.8 as well. Moving from top to bottom within a column as request volume

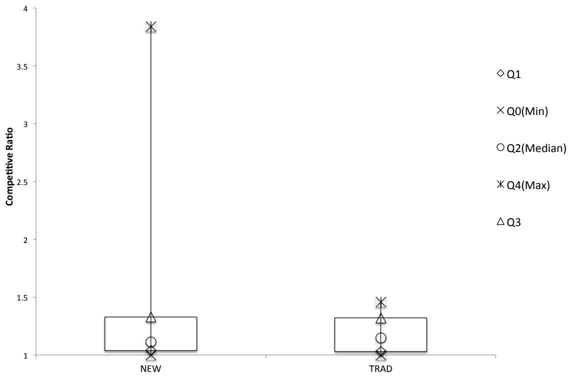
Instance	Approach	Avg % deviation from HIND	Num. replicates in which approach is preferred
SL	NEW	27.8%	416
	TRAD	**18.1%**	**530**
SM	**NEW**	**17.1%**	**717**
	TRAD	33.0%	269
SH	**NEW**	**16.2%**	**791**
	TRAD	39.0%	198
LL	NEW	21.5%	469
	TRAD	**20.4%**	**486**
LM	NEW	27.6%	599
	TRAD	24.3%	390
LH	**NEW**	**25.3%**	**657**
	TRAD	26.9%	330

Table 2.4: Computational results summary

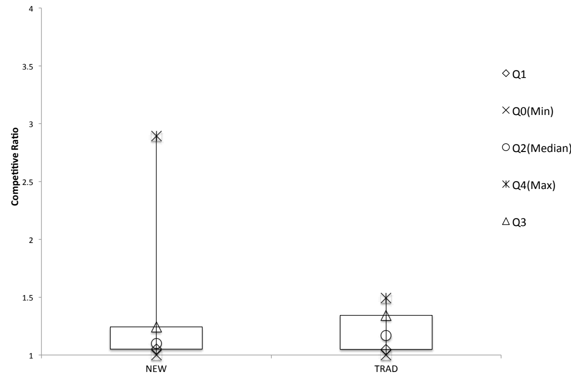
increases, the preference for NEW becomes more apparent as the box representing the middle 50% of solutions produced by NEW moves lower on the y-axis (competitive ratio) than the analogous box representing the middle 50% of solutions produced by TRAD. Moving from left to right within a row as region size increases, the relative performance of TRAD improves as the box representing the middle 50% of solutions produced by TRAD moves lower with respect to the y-axis.

2.7 Conclusions

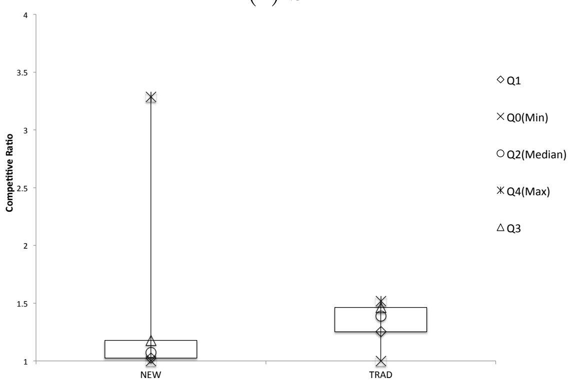
When developing disaster relief response plans, emergency managers have historically considered information communicated through traditional means, such as on the ground assessments. In this paper, we have studied whether relief requests communicated through social media should also be considered when developing disaster relief routing plans. Two alternative decision approaches are analyzed — one that considers social media information



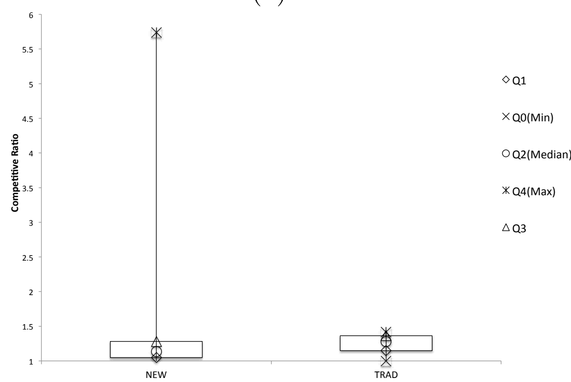
(a) SL



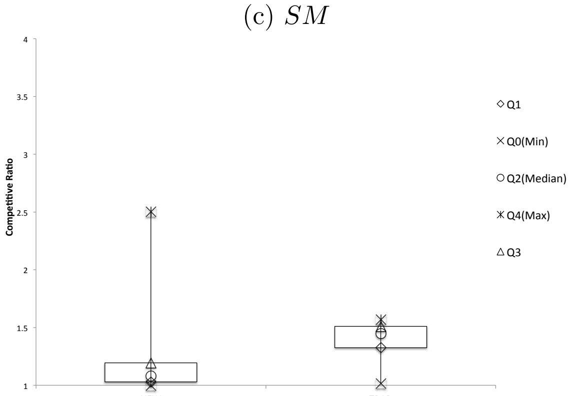
(b) LL



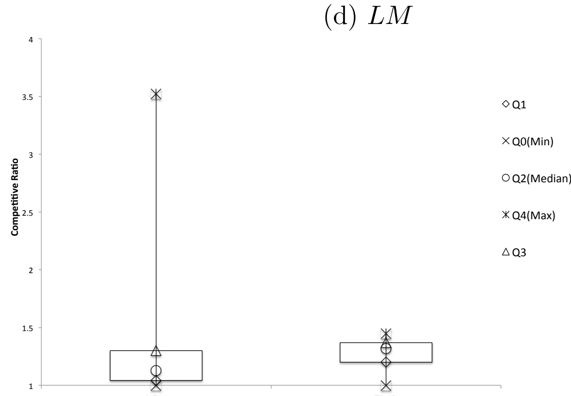
(c) SM



(d) LM



(e) SH



(f) LH

Figure 2.8: Computational results box plot

(NEW) and one that does not (TRAD) — and compared to an offline reference approach that benefits from knowing in advance the accuracy of social media requests (HIND).

The alternative decision approaches are assessed for a set of case study instances with uniformly distributed request locations using an analytical study and a computational study. In the analytical study, information regarding request volume and region size are not required. Instead, competitive ratios for the alternative decision approaches are approximated using only information about the proportion of total requests that are unverified (ρ) and the proportion of unverified requests that are accurate (λ). In the computational study, specific request locations and their classification as verified or unverified and accurate or inaccurate are required. Then, competitive ratios for the alternative decision approaches are computed exactly. For the case study instances with uniformly distributed request locations, it was demonstrated NEW had superior average-case performance while TRAD had superior worst-case performance in both the analytical and computational studies. That is, on average across all instance variants considered, the time to service requests using the NEW approach was closer to the reference offline approach than in the routes generated by TRAD. However, the deviation of NEW from the offline reference approach had greater magnitude in the worst-case than did TRAD. Therefore, NEW may be preferred by an emergency manager wishing to focus on average or expected performance, while TRAD may be preferred by an emergency manager wishing to minimize the worst-case performance.

Specific information regarding the nature of the demand scenario at hand may influence the choice of decision approach. When faced with a disaster relief routing planning problem, an emergency manager will have information regarding the size of the geographic region across which demand is distributed, total request volume, proportion of total requests that are unverified, and possibly some assumptions regarding the accuracy of unverified requests. The results presented in this paper can be useful for this emergency manager. Suppose for example an emergency manager has received information regarding 40 verified requests and 10 unverified requests both uniformly distributed in an impacted region. Then, the results corresponding to medium request volume instances with $\rho = 0.20$ can be considered. While the emergency manager will not know the proportion of unverified requests that are accurate with certainty, they can study the relative performance of TRAD and NEW as λ varies. In this case, the emergency manager would find as long as at least 10% of the unverified information were accurate, NEW would be the preferred approach.

The results presented in this paper rely on several assumptions. First, uniformly distributed request locations are required. Furthermore, each subset of requests (unverified versus verified, accurate versus inaccurate) are also assumed to be uniformly distributed. However, the computational study could be replicated for non-uniformly distributed instances in order to gain insight into decision approach performance for those cases. Second, the TRAD decision approach assumes the accuracy of unverified requests can be assessed using alternate resources while an initial tour to verified locations is executed. This assumption is most valid when the proportion of unverified requests to verified requests is small, so sufficient time is available during the tour visiting verified requests to verify the remaining information. However, regardless of the size of the unverified request set, the combined duration of the two tours developed according to TRAD is still a valid lower bound on the total time required to visit all accurate requests using this approach. For example, an initial tour could include all verified requests, a second tour could include a subset of unverified requests newly classified as accurate, a third tour could include the next subset of newly classified requests, and so on.

A third assumption used in this paper is negligible service time at customer locations. If service time is not negligible, the penalty associated with visiting inaccurate locations increases. However, it is still anticipated that the time required to verify a location as inaccurate upon arrival should be small when compared with the time required to unload relief goods at accurate demand locations. An area for future work is to consider service time that depends on the magnitude of goods being delivered. Fourth, we have only considered inaccuracy that results from demand not being present at an unverified location. To consider demand that is present but with a different magnitude than expected, the models presented here must be extended. Finally, we have considered only a single vehicle with unlimited capacity, saving variations of constraints on the resource set for future work. These simplifying assumptions were made in order to introduce a formal method for quantitatively assessing the impact of including unverified information in disaster relief planning and demonstrate its usefulness. The insights this framework is able to provide will become more valuable as the methodology is extended to incorporate realistic problem characteristics such as multiple vehicles with limited capacity, request time windows, non-uniform request location distributions, and demand-dependent service times.

An additional area for future study is to incorporate other objectives that may be more important in some disaster relief settings than traditional efficiency objectives of minimizing response time. For example, maximizing the amount of demand served and minimizing total and latest arrival times are reasonable objectives in disaster response. Considering the magnitude of demand at each request location could lead to policies that would, for example, prioritize serving an unverified location with a large amount of demand earlier than an verified location with a small amount of demand. Equity objectives may be important in disaster response also. Because the population using social media to communicate their needs may differ demographically from the population not doing so, it may be important to balance the consideration given to each type of information.

2.8 Appendix

Table 2.5: Table of analytical results

ρ	λ	TRAD	NEW	ρ	λ	TRAD	NEW	ρ	λ	TRAD	NEW
0.1	0.0	1.00	1.05	0.4	0.0	1.00	1.29	0.7	0.0	1.00	1.83
0.1	0.1	1.10	1.05	0.4	0.1	1.22	1.25	0.7	0.1	1.34	1.64
0.1	0.2	1.14	1.04	0.4	0.2	1.28	1.21	0.7	0.2	1.39	1.51
0.1	0.3	1.16	1.04	0.4	0.3	1.32	1.18	0.7	0.3	1.41	1.40
0.1	0.4	1.18	1.03	0.4	0.4	1.35	1.15	0.7	0.4	1.41	1.31
0.1	0.5	1.20	1.03	0.4	0.5	1.37	1.12	0.7	0.5	1.41	1.24
0.1	0.6	1.22	1.02	0.4	0.6	1.38	1.09	0.7	0.6	1.41	1.18
0.1	0.7	1.23	1.02	0.4	0.7	1.39	1.07	0.7	0.7	1.40	1.13
0.1	0.8	1.24	1.01	0.4	0.8	1.40	1.04	0.7	0.8	1.40	1.08
0.1	0.9	1.25	1.01	0.4	0.9	1.40	1.02	0.7	0.9	1.39	1.04
0.1	1.0	1.26	1.00	0.4	1.0	1.41	1.00	0.7	1.0	1.38	1.00
0.2	0.0	1.00	1.12	0.5	0.0	1.00	1.41	0.8	0.0	1.00	2.24
0.2	0.1	1.14	1.10	0.5	0.1	1.25	1.35	0.8	0.1	1.38	1.89
0.2	0.2	1.19	1.09	0.5	0.2	1.32	1.29	0.8	0.2	1.41	1.67
0.2	0.3	1.23	1.08	0.5	0.3	1.36	1.24	0.8	0.3	1.41	1.51
0.2	0.4	1.25	1.07	0.5	0.4	1.38	1.20	0.8	0.4	1.40	1.39
0.2	0.5	1.28	1.05	0.5	0.5	1.39	1.15	0.8	0.5	1.39	1.29
0.2	0.6	1.29	1.04	0.5	0.6	1.40	1.12	0.8	0.6	1.38	1.21
0.2	0.7	1.31	1.03	0.5	0.7	1.41	1.08	0.8	0.7	1.37	1.15
0.2	0.8	1.32	1.02	0.5	0.8	1.41	1.05	0.8	0.8	1.36	1.09
0.2	0.9	1.33	1.01	0.5	0.9	1.41	1.03	0.8	0.9	1.35	1.04
0.2	1.0	1.34	1.00	0.5	1.0	1.41	1.00	0.8	1.0	1.34	1.00
0.3	0.0	1.00	1.20	0.6	0.0	1.00	1.58	0.9	0.0	1.00	3.16
0.3	0.1	1.18	1.17	0.6	0.1	1.29	1.47	0.9	0.1	1.41	2.29
0.3	0.2	1.24	1.15	0.6	0.2	1.36	1.39	0.9	0.2	1.40	1.89
0.3	0.3	1.28	1.13	0.6	0.3	1.39	1.31	0.9	0.3	1.37	1.64
0.3	0.4	1.31	1.10	0.6	0.4	1.40	1.25	0.9	0.4	1.35	1.47
0.3	0.5	1.33	1.08	0.6	0.5	1.41	1.20	0.9	0.5	1.33	1.35
0.3	0.6	1.34	1.07	0.6	0.6	1.41	1.15	0.9	0.6	1.31	1.25
0.3	0.7	1.36	1.05	0.6	0.7	1.41	1.10	0.9	0.7	1.30	1.17
0.3	0.8	1.37	1.03	0.6	0.8	1.41	1.07	0.9	0.8	1.29	1.10
0.3	0.9	1.38	1.02	0.6	0.9	1.41	1.03	0.9	0.9	1.27	1.05
0.3	1.0	1.38	1.00	0.6	1.0	1.41	1.00	0.9	1.0	1.26	1.00

Bibliography

[1] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberan, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Research report 949-M, Universite Joseph Fourier, Grenoble, France, 1995.

[2] J. Beardwood, J. Halton, and J. Hammersley. The shortest path through many points. Proceedings of the Cambridge Philosophical Society, 55:299–327, 1959.

[3] Dimistris J. Bertsimas. A vehicle routing problem with stochastic demand. Operations Research, 40(3):574–585, 1991.

[4] Dennis E. Blumenfeld and Martin J. Beckmann. Use of continuous space modeling to estimate freight distribution costs. Transportation Research Part A: General, 19(2):173–187, 1985.

[5] Ann Melissa Campbell and Barrett W. Thomas. Challenges and Advances in A Priori Routing, volume 43. Springer US, Operations Research/Computer Science Interfaces, 2008.

[6] Pew Research Center. Twitter a key source of news for many during Hurricane Sandy. Available at `http://www.pewresearch.org/daily-number/twitter-a-key-source-of-news-for-many-during-hurricane-sandy/`, 2012.

[7] T. William Chien. Operational estimators for the length of a traveling salesman tour. Computers & Operations Research, 19:469–478, 1992.

[8] N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. Operational Research Quarterly, 20:309–318, 1969.

[9] William J. Cook. Concorde TSP solver. Available at `http://www.tsp.gatech.edu`, 2009.

[10] American Red Cross. Disaster services program review: fiscal year 2011. Available at `www.redcross.org/images/MEDIA_CustomProductCatalog/m8540111_Disaster-Relief-Program-Review-2011.pdf`, 2011.

[11] American Red Cross. More Americans using mobile apps in emergencies. Available at `http://www.redcross.org/news/press-release/More-Americans-Using-Mobile-Apps-in-Emergencies`, 2012.

[12] Carlos F. Daganzo. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. Transportation Science, 18(4):331–350, 1984.

[13] Carlos F. Daganzo. The length of tours in zones of different shapes. Transportation Research Part B: Methodological, 18:135–145, 1984.

[14] Luis E. de la Torre, Irina S. Dolinskaya, and Karen R. Smilowitz. Disaster relief routing: Integrating research and practice. Socio-Economic Planning Sciences, 46:88–97, 2012.

[15] Project Epic. Tweak the tweet. Available at http://epic.cs.colorado.edu/?page_id=11. Accessed September 2012.

[16] Elizabeth Ferris and Daniel Petz. The year that shook the rich: a review of natural disasters in 2011. Technical report, The Brookings-LSE Project, Washington, DC, March 1 2012. Available at http://www.brookings.edu/research/reports/2012/03/natural-disaster-review-ferris.

[17] Miguel Andres Figliozzi. Planning approximations to the average length of vehicle routing problems with time window constraints. Transportation Research Part B: Methodological, 43(4):438–447, 2009.

[18] Heather Ford. Can Ushahidi rely on crowdsourced verifications? Idea Lab, November 28 2011. Available at http://www.pbs.org/idealab/2011/11/can-ushahidi-rely-on-crowdsourced-verifications325.html.

[19] Craig Fugate. Understanding the power of social media as a communication tool in the aftermath of disasters, Testimony before the Homeland Security and Governmental Affairs Committee, Subcommittee on Disaster Recovery and Intergovernmental Affairs. Available at http://www.dhs.gov/news/2011/05/04/written-statement-craig-fugate-administrator-federal-emergency-management-agency, 2011.

[20] Michel Gendreau, Alain Hertz, Gilbert Laporte, and Mihnea Stan. A generalized insertion heuristic for the traveling salesman problem with time windows. Operational Research, 3:330–335, 1998.

[21] Michel Gendreau, Gilbert Laporte, and Rene Seguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. Operations Research, 44(3):469–477, 1996.

[22] Joe Hagan. Tweet science. New York Magazine, Published October 2, 2011. Available at nymag.com/news/media/twitter-2011-10/.

[23] Amanda Lee Hughes and Leysia Palen. Twitter adoption and use in mass convergence and emergency events. Proceedings of the 6th International ISCRAM Conference, Goteborg, Sweden, May 2009.

[24] Harvard Humanitarian Initiative. Disaster relief 2.0: the future of information sharing in humanitarian emergencies. Technical report, UN Foundation and Vodafone Foundation Technology Partnership, Berkshire, UK and Washington, D.C., 2011.

[25] Clarence Wardell III and Yee San Su. 2011 social media + emergency management camp: transforming the response enterprise. Technical report, CNA Analysis & Solutions, 4825 Mark Center Drive, Alexandria, VA, 22311, 2011. Available at http://www.wilsoncenter.org/sites/default/files/SMEM_Report.pdf.

[26] Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Operations Research, 36(6):929–936, 1988.

[27] D.S. Johnson, L.A. McGeoch, and E.E. Rothberg. Asymptotic experimental analysis for the held-karp traveling salesman bound. Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pages 341–350, 1996.

[28] Mark Kramer, Roger Costello, and John Griffith. Asymptotic experimental analysis for the held-karp traveling salesman bound. Mind and Society: Cognitive Studies in Economics and Social Sciences, 8:209–221, 2009.

[29] Ohseok Kwon, Bruce Golden, and Edward Wasil. Estimating the length of the optimal tsp tour: an empirical study using regression and neural networks. Computers & operations research, 22(10):1039–1046, 1995.

[30] Gilbert Laporte, Franois V. Louveaux, and Helene Mercure. A priori optimization of the probabilistic traveling salesman problem. Operations Research, 42(3):543–549, 1994.

[31] Bruce R. Lindsay. Social media and disasters: current uses , future options, and policy considerations. Technical report, Congressional Research Service, 7-5700 `www.crs.gov`, R41987, 2011.

[32] Susannah McClendon and Anthony C. Robinson. Leveraging geospatially-oriented social media communications in disaster response. Proceedings of the 9th International ISCRAM Conference, Vancouver, Canada, April 2012.

[33] Patrick Meier. How the UN used social media in response to Typhoon Pablo. Available at `http://irevolution.net/2012/12/08/digital-response-typhoon-pablo/`, 2012.

[34] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: can we trust what we RT? Proceedings of the First Workshop on Social Media Analytics, Washington, D.C., pages 71–79, 2010.

[35] H.L. Ong and H.C. Huang. Asymptotic expected performance of some {TSP} heuristics: An empirical evaluation. European Journal of Operational Research, 43(2):231–238, 1989.

[36] Will Oremus. Building a better truth machine: could machine-learning algorithms help debunk Twitter rumors before they spread? Available at `http://www.slate.com/articles/technology/future_tense/2012/12/social_media_hoaxes_could_machine_learning_debunk_false_twitter_rumors_before.html`, 2012.

[37] Teun Terpstra, A. de Vries, R. Stronkman, and G.L. Paradies. Towards a realtime twitter analysis during crises for operational crisis management. Proceedings of the 9th International ISCRAM Conference, Vancouver, Canada, April 2012.

[38] Robert Thomson, Naoya Ito, Hinako Suda, Fangyu Lin, Yafei Liu, Ryo Hayasaka, Ryuzo Isochi, and Zian Wang. Trusting tweets: The fukushima disaster and information source credibility on twitter. Proceedings of the 9th International ISCRAM Conference, Vancouver, Canada, April 2012.

[39] Tim Tinker and Elaine Vaughan. Risk and crisis communications: best practices for government agencies and non-profit organizations. Technical report, Booz Allen Hamilton, 2010. Available at `http://www.boozallen.com/insights/insight-detail/riskandcrisis`.

[40] UN-SPIDER. Lessons from haiti. Coordinates, 6:27–31, 2010.

[41] Ushahidi. Haiti dataset. Available at `http://community.ushahidi.com/research/datasets/`. Accessed June 2012.

[42] Ushahidi. The SwiftRiver Platform. Available at `http://ushahidi.com/products/swiftriver-platform`. Accessed December 2012.

[43] Sarah E. Vieweg. Situational awareness in mass emergency: a behavioral and linguistic analysis of microblogged communications. Available at `http://works.bepress.com/vieweg/15`, 2012.

[44] Eric Gelinas Yvan Dumas, Jacques Desrosiers and Marius M. Solomon. An optimal algorithm for the traveling salesman problem with time windows. Operational Research, 43:367–371, 1995.

3. A CASE STUDY ON THE TEAM ORIENTEERING PROBLEM WITH THE INCORPORATION OF UNCERTAIN SOCIAL DATA IN DISASTER RELIEF TOUR PLANNING

3.1 Introduction

As in Chapter 2, in this chapter a response planning problem with two distinct classes of information available at the time of planning is considered: (i) unverified social data describing demand that is not known with certainty, and (ii) verified data describing demand known with certainty which is obtained by trusted traditional sources. The models presented in Chapter 2 are extended here for additional practical considerations such as multiple vehicles, request specific demand magnitudes, and demand-dependent service times. The goal of the problem introduced in this study is to deliver products and essential services to as many victims as possible in order to minimize human suffering and saves lives.

Specifically, this chapter investigates whether it is worthwhile to consider and to act on social data requests prior to their absolute verification in the context of planning disaster relief routes. The problem under consideration is a Team Orienteering Problem (TOP) variant with a single depot and a homogenous fleet of uncapacitated vehicles. A case study motivated by the 2010 Haiti earthquake is developed using real social data to demonstrate the usefulness of the framework. Possible scenarios are also analyzed to identify under which circumstances social data integration improves response efficacy. An Adaptive Large Neighborhood Search (ALNS) algorithm is developed to solve the case problem instances and to analyze the performance of alternative decision strategies for the delivery of relief supplies.

The contributions of this work are three-fold. First, this chapter extends the modeling of the single-vehicle variant of disaster relief tour planning (DRTP) to a maximization objective variant with a homogeneous fleet of vehicles. Second, a set of practical social data strategies is introduced to represent and model how emergency managers use social data in response planning. Third, the alternative strategies are tested for comparative effectiveness in a case study, which is developed using real social data from the Haiti earthquake. It is anticipated that the case study instances will become benchmark instances for future researchers.

This chapter is organized as follows. Section 3.2 provides a review of existing literature regarding team orienteering problems. In Section 3.3, a formal problem statement and alternative social strategies are presented. Section 3.4 presents the solution method in detail. The experimental settings are presented in Section 3.5. Case study development is discussed in Section 3.6. Finally, Section 3.7 summarizes the work and presents the conclusions.

3.2 Literature review

The problem being described is a Team Orienteering Problem (TOP) variant and falls into the general category of the vehicle routing problem under uncertainty. Because a detailed review in the area of vehicle routing with the presence of uncertainty is provided in Chapter 2, in this section, only works that focus on solving the TOP variants are discussed briefly. The aim of the TOP introduced in Chao et al. [1] is to maximize the total scores collected within a specified timeframe by visiting a set of nodes, each with a given score. The number of vehicles is limited, and each node can be visited once at most. Several variants of the Orienteering Problem (OP), a special case of the TOP with only a single vehicle, can be found in the literature, such as the Orienteering Problem with Time Windows (OPTW) and the Team Orienteering Problem with Time Windows (TOPTW). For a comprehensive review of OP variants and applications, the reader is referred to Vansteenwegen et al. [2].

Butt and Cavalier [3] first apply the TOP to athlete recruiting from high schools. A scout has a specified number of days during which they can visit schools. A score is assigned to each school based on its recruiting potential, and the scout tries to maximize recruiting potential. A simple heuristic algorithm is presented in which a school is selected based on the ratio of its score and distance. Chao et al. [1] present a heuristic approach for the TOP and introduce benchmark instances. In the proposed heuristic algorithm, a seed node, which is farthest from the starting and ending nodes, is selected, and remaining nodes are inserted into the routes using the cheapest insertion method. Tang and Miller-Hooks [4] describe a TOP application of routing technicians, who can only work a certain number of hours, to service customers. A tabu search approach with adaptive memory is proposed. Their approach generates better solutions than Chao et al. [1] but uses more computational time. Archetti et al. [5] implement two variants of tabu search and a variable neighborhood search (VNS) for solving the TOP. The VNS outperforms Chao et al. [1] and Archetti et al. [5]. Ke et al. [6] propose an Ant Colony Optimization (ACO) approach, which finds the best known solutions for 359 instances out of 387 benchmark instances. Vansteenwegen et al. [7] present a guided local search method which requires less computational time but the solution quality is not better than previous algorithms. Dang et al. [8] implement a particle swarm optimization inspired algorithm to solve the TOP instances. They found all best known solutions except one. Kim et al. [9] propose an augmented large neighborhood search method and obtain the same set of results as Dang et al. [8] in less computational time. Boussier et al. [10] present an exact solution approach for solving both the TOP and the selective vehicle routing problem with time windows (SVRPTW) with additional side constraints. They can solve the TOP instances exactly for up to 100 customers. In this study, we consider a TOP variant where there is uncertainty in demand. An objective of maximizing amount of demand served is considered. The proposed routing model can provide decision support for emergency response teams that focus on taking supplies and services to the people in need.

3.3 Problem statement

The routing problem under consideration includes two types of information at the time of planning: (i) unverified social data $\mathcal{S}$ describing demand that is not known with certainty and (ii) verified traditional data $\mathcal{T}$ describing demand known with certainty. The set of all requests is denoted $\mathcal{N}$, and thus $\mathcal{N} = \mathcal{T} \cup \mathcal{S}$. Each request in N specifies a location, demand magnitude, and type (e.g. food, shelter, medical). A request is accurate if the demand exists in the stated magnitude and type at the specified location; otherwise, the request is inaccurate. Therefore, social data requests are classified as unverified requests until they can be investigated and re-classified as either accurate $\mathcal{SA}$ or inaccurate $\mathcal{SI}$; note that $\mathcal{S} = \mathcal{SA} \cup \mathcal{SI}$. We assume that a fleet of vehicles with unlimited capacity serves request locations within a specified time interval. The goal is to develop vehicle routes that visit as many accurate demand locations ($i \in \mathcal{T} \cup \mathcal{SA}$) as possible exactly once during response operations.

This can be modeled as a Team Orienteering Problem variant that is defined as follows. Assume that $G = (\mathcal{N}, \mathcal{A})$ is a directed network graph with set of $n+1$ nodes, where $\mathcal{N} = \{0, 1, \ldots, n\}$ is the set of nodes, and $\mathcal{A} = \{(i,j) : i \neq j \in \mathcal{N}\}$ is the set of arcs connecting nodes in $\mathcal{N}$. The node set $\mathcal{N}$ includes a starting node (i= 0) and a set of demand nodes $\{i = 1, \ldots, n\}$. Therefore, $\mathcal{N} = \mathcal{T} \cup \mathcal{S} \cup \{0\}$. The travel time on arc (i,j), denoted t_{ij}, is known. Let $c_i(j,k)$ be the cost of inserting node i between nodes j and k, where $c_i(j,k) = t_{ji} + t_{ik} - t_{jk}$. Each location $i \in \mathcal{T} \cup S$ can be visited at most once, and the amount of demand $d_i > 0$ will be served if there is a visit to location i within the specified time limit $\mathcal{T}_{max}$. The magnitude of demand for each location, d_i, is known, and no demand occurs at the depot, $d_0 = 0$. It is assumed a homogeneous fleet of m uncapacitated vehicles is based at the depot to deliver relief supplies.

Let $R_k = \{0, i_1, i_2, \ldots, i_n, 0\}$ be a vehicle route k and let i_j denote the customer index in position j in the route. Service time s_a denotes the time required for unloading relief supplies at accurate request locations ($\mathcal{T} \cup \mathcal{SA}$). When a route R_k includes a vehicle visit to an inaccurate social data location i, the time required to confirm a location as inaccurate ($\mathcal{SI}$) upon a vehicle arriving there is s_i; no supplies will be unloaded. Thus, the total accurate demand served on a vehicle route k is denoted $d(R_k) = \sum_{\{i \in R_k : i \in (\mathcal{T} \cup \mathcal{SA})\}} d_i$. Due to the limited number of vehicles and time budget constraints, visiting all locations ($\mathcal{N}$) may not be feasible. Consequently, the objective is to identify a set of feasible vehicle routes that maximizes the total amount of accurate demand served, $max \sum_{k \in \mathcal{V}} d(R_k)$.

3.3.1 Social data strategies

Several decision strategies that a decision maker could adopt in practice for planning vehicle routes are considered. The decision strategies included are intended to reflect three distinct perspectives on the current and future value of social data. These perspectives are one that never acts on social data elements unless they are verified, one that always act on social data elements with no attempt at verification, and one that acts on social data elements by considering various alternative scenarios. Each of these perspectives is motivated from a 2012 survey conducted by CNA and NEMA [11]. A total of five social data decision strategies are described in order to represent the broad range of emergency manager preferences for solving the proposed problem. The first two decision strategies are as follows:

- **Only Verified Requests** ($OnlyVerified$): This approach does not include social data requests ($\mathcal{S} \in \mathcal{N}$) in vehicle tours. Therefore, only traditional requests in $\mathcal{T}$ can appear in the set of routes developed. This strategy represents more than three-quarters of agencies participating in the survey that would not act on unverified information obtained via social media.

- **All Requests** (All): This method always acts on social data requests in addition to traditional requests without attempting to verify them. Therefore, both social data ($\mathcal{S}$) requests and traditional requests ($\mathcal{T}$) can appear in developed routes. This strategy represents one out of ten agencies surveyed that would not hesitate to act on social data.

Several additional decision strategies represent more than two-thirds of agencies that expect social data to have at least a moderate impact on future response efforts.

- **Verified Requests First, Unverified Requests Last** ($FirstVerified$): This strategy only allows for visiting social data requests after all verified requests have been visited. In this strategy, the social data requests will always be visited at the end of the tour.

- **Prioritize Verified Requests** ($PrioritizeVerified$): This strategy prioritizes traditional requests over social requests by assigning higher weight to them. This allows inserting social data requests into vehicle tours if time capacity (T_{max}) allows. Decision makers consider weighted-demand, $w_i \times d_i$, in the decision-making process, where w_i is the assigned weight to location i and d_i is the demand at that location. This strategy would allow a social data location to appear anywhere within a tour.

- **Unverified Requests Located Close to Verified Requests** ($CloseProximity$): This method allows acting on social data locations if they are located close to a traditional request that is already on the tour. An emergency manager considers only traditional data first when making vehicle tour decisions. Then the strategy allows considering social data locations ($\mathcal{S}$) by finding a feasible insertion place in a tour. More precisely, this method inserts a social location i between nodes j and $j+1$ such that $c_i(j, j+1)$ is minimal. With this strategy, a decision maker is willing to visit an unverified social request location that is in close proximity to a verified traditional location and it doesn't exceed the time limit after a best possible tour plan has been developed for traditional locations.

This research focuses on quantifying the tradeoff between the timeliness and usefulness of social data by comparing the performance of vehicle routes developed using these emergency decision strategies.

3.4 Methodology

Adaptive large neighborhood search (ALNS) is modified to solve the routing problems induced by each decision strategy in this study. ALNS is a local search algorithm that was first developed by Ropke and Psinger [12] for the pickup and delivery problem. The same methodology was later successfully applied to several variants of the vehicle routing problem, such as the capacitated vehicle routing problem, multi-depot vehicle routing problem, vehicle routing problem with multiple routes, pollution-routing problem, and inventory routing problem [13, 14, 15, 16, 17]. ALNS is an extension of the large neighborhood search (LNS) framework presented by Shaw [18]. In LNS, a solution is partially destroyed and repaired through the application of several operators. The difference between ALNS and LNS is that the operators are selected in a probabilistic and adaptive fashion in ALNS, according to their past performance. Several ruin and recreate operators compete to improve the current solution [19]. A performance score is assigned to each operator, and the score is increased whenever a new solution is produced by the operator.

An overview of the ALNS scheme we use is provided in Algorithm 1. The algorithm begins generating an initial feasible solution s_{init} using a simple constructive heuristic. After all request locations associated with a chosen decision strategy are initially placed in the unvisited location list u, a request location i is randomly removed from u in order to find a feasible insertion place in a vehicle route r. This process is repeated until there are no request locations to visit (u is empty) or there are no feasible insertion places. Starting from the initial solution, ALNS explores the solution space by successively destroying a part of the current solution and then reconstructing it in a different way. This neighborhood search process is performed using a pair of pre-determined removal and insertion operators. A pair of operators ψ is probabilistically chosen based on their historical performance. At the end of each iteration, a new solution s_{new} is obtained by applying the pair of removal and insertion operators to the current solution $s_{current}$. Then s_{new} is accepted or rejected based on simulated annealing acceptance criteria described later. Then the performance associated with the applied pair of operators ψ is updated. The algorithm stops when a fixed number of ALNS iterations have passed or after there has been no improvement in the objective function for η% of total iterations. Then the best solution s_{best} is returned. In the following, each component of this algorithm will be explained in detail.

Algorithm 1 A general adaptive large neighborhood search algorithm with simulated annealing

generate initial solution s_{init}
$s_{current} \leftarrow s_{best} \leftarrow s_{init}$
initialize probabilities of all pairs of destroy and repair operators
while stopping criterion is not met **do**
    select a removal/insertion pair ψ based on their past performance
    apply ψ on the solution $s_{current}$ to obtain its neighbor solution s_{new}
    if (s_{new} is better than $s_{current}$) **then**
        update the current solution $s_{current} \leftarrow s_{new}$
        update performance of pair ψ
    else if (rand[0, 1] $\leq e^{-(\Delta d/temperature)}$)
        update the current solution $s_{current} \leftarrow s_{new}$
        update performance of pair ψ
    end if
    if ($s_{current}$ is better than s_{best}) **then**
        update the best solution $s_{best} \leftarrow s_{current}$
        update performance of pair ψ
    end if
    update $temperature \leftarrow temperature * \alpha$
end while
return s_{best}

3.4.1 Acceptance and stopping criteria

A simulated annealing based acceptance criterion is used to decide whether to accept or reject a new solution s_{new}. The new solution is accepted over $s_{current}$ if s_{new} has a larger demand magnitude than $s_{current}$. Otherwise, s_{new} is accepted with probability $e^{-(\Delta d/temperature)}$, where the temperature is greater than zero and Δd is the difference between the objective values of s_{new} and $s_{current}$. Starting from an initial temperature value, the temperature is decreased every iteration by a factor of α, where $0 < \alpha < 1$ is the cooling rate.

3.4.2 Adaptive mechanism

The search is divided into two segments of consecutive iterations. In the first segment, all pairs of operators have the same score and are selected with equal probability . Statistics associated with the performance for operators are collected during this segment. Since the removal and insertion operators are chosen as a pair, their performance is evaluated pairs. After the first segment, pairs of operators that have successfully found new improvement solutions have a higher score and thus a higher probability of being chosen. Let φ_ψ be a measure of how well pair ψ has performed in past iterations. Then, the probability of selecting pair ψ is $p_\psi = \varphi_\psi / \sum_{j=1}^{h} \varphi_j$, where h is the total number of pairs. The score φ_ψ of a pair ψ is increased by score adjustment parameters σ_1, σ_2, and σ_3. The score φ_ψ=0 is set to zero at the beginning and rises incrementally when a pair ψ results in a new solution. More precisely, if a pair of removal and insertion operators finds a new best solution, the score is increased by σ_1, if it finds a solution better than the current solution but worse than the current best, its score is increased by σ_2, and if it finds a solution worse than the current solution which is accepted according to the simulated annealing criteria, the score is increased by σ_3.

3.4.3 Removal and reinsertion operators

Nine removal operators and three insertion operators are considered. Removal operators are paired with insertion operators; thus, there are twenty-seven pairs of operators. A new vehicle plan is obtained by first using a removal operator to remove r request locations from the vehicle plan and place them in the unvisited request list u. Note that other locations may already be in the list u also, if they were not accommodated in the plan in a previous iteration. After removal, the insertion operator places some, but maybe not all, of the request locations in u into vehicle routes.

3.4.3.1 Removal operators

The set of removal operators used is described below. Ropke and Pisinger [12] explain that the number of requests removed from the current routing plan has a significant impact on the performance of ALNS. A too small or too large part of the solution should not be removed [12, 13]. Therefore, the number of request locations to remove r from the current vehicle plans is determined by the percentage range $[a, b]$ to remove.

1. **Random removal** removes randomly chosen the r request locations from the plan. This simple approach is intended to diversify the search space.

2. **Worst distance removal** successively removes the r request locations which bring the maximum distance savings.

3. **Smallest demand removal** successively removes the r locations that have minimum demand. The idea is to open potential insertion request locations within tours for unvisited requests with higher demand.

4. **Proximity based removal** attempts to remove visits to request locations geographically near each other. This operator is a special case of the Shaw removal proposed in [18]. The goal of the Shaw removal is to remove a set of nodes that are related in a predefined way. We choose to base the relatedness measure on proximity. First, a seed request c is randomly selected from the unvisited location list u. Then the r request locations nearest to c are removed from vehicle tours in which they appear. The idea is to create openings in routes that are likely to accommodate requests that are currently unvisited and located near seed c.

5. **Smallest demand per distance removal** removes the request location i which brings the smallest demand per distance, $d_i/c_i(predecessor(i), successor(i))$, where the successor of request i is denoted by successor(i), and the predecessor of request (i) is denoted by predecessor(i). The motivation is to remove requests for which the anticipated payoff (demand served) is relatively low compared to the distance the vehicle must travel to reach the location.

6. **Worst demand per distance removal** removes a location from a route which is extreme with respect to the current average demand per distance in the route. The operator selects request i that brings the maximum demand per distance $\{d(R_k -$

$\{i\})/t(R_k - \{i\})\}$ where $d(R_k - \{i\})$ represents the total demand served on a vehicle route k excluding i, and $t(R_k - \{i\})$ represents the total distance of the vehicle route k without i. This operator tries to improve a shortcoming of the smallest demand per distance removal by considering the impact to an entire route when a request is removed, as opposed to the impact only on the request's neighbor.

Proposition. *Smallest demand per distance and worst demand per distance removal operators do not necessarily choose the same node for removal.*

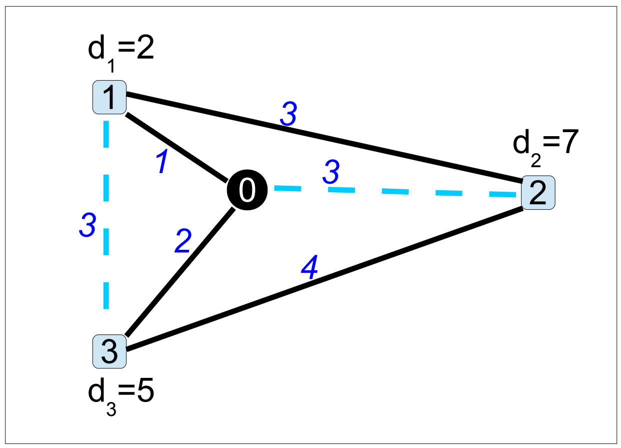


Figure 3.1: A vehicle route

Consider for example the given network in Figure 3.1. Assume a vehicle route is 0-1-2-3-0, where 0 represents the depot and all other nodes have demand d_i. The node to remove according to each operator is given below.

Worst demand per distance:

$$\text{For node 1: } \left(\frac{d_2 + d_3}{t_{02} + t_{23} + t_{30}}\right) = \left(\frac{7+5}{3+4+2}\right) = 1.33$$

$$\text{For node 2: } \left(\frac{d_1 + d_3}{t_{01} + t_{13} + t_{30}}\right) = \left(\frac{2+5}{1+3+2}\right) = 1.16$$

$$\text{For node 3: } \left(\frac{d_1 + d_2}{t_{01} + t_{12} + t_{20}}\right) = \left(\frac{2+7}{1+3+3}\right) = 1.28$$

Node 1 is chosen to remove, because it has the highest value.

Smallest demand per distance:

$$\text{For node 1: } \frac{d_1}{t_{01} + t_{12} - t_{02}} = \frac{2}{1+3-3} = 2$$

$$\text{For node 2: } \frac{d_2}{t_{12} + t_{23} - t_{13}} = \frac{7}{3+4-3} = 1.75$$

$$\text{For node 3: } \frac{d_3}{t_{23} + t_{30} - t_{02}} = \frac{5}{4+2-3} = 1.66$$

Node 3 is chosen to remove, because it has the lowest value.

These two operators may choose different nodes to remove from a vehicle route, and thus, they are not identical.

7. **Historical knowledge removal** keeps a record of the best position cost of every request location to decide which request location to remove. The best position cost represents the minimum insertion cost for a request location for all locations it has

appeared in throughout the algorithm. A position cost of request location i is calculated as $\kappa(i) = c_i(predecessor(i), successor(i))$. The best position cost $\kappa(i^*)$ of request i is updated throughout the algorithm. A request i' is chosen with the maximum deviation from its best position cost, $\{\kappa(i) - \kappa(i^*)\}$. The motivation is that considering historical information may improve the performance of local search (as in some other local search algorithms, long-term memory is in the form of a tabu list, for example). This operator is similar to the neighborhood graph removal used in Ropke and Pisinger [20] and historical knowledge node removal used in Demir et al. [16]. To receive good performance from this operator, it should be excluded from selection in the early iterations of the neighborhood search. Otherwise, the best position cost will be based on a history of bad solutions.

8. **Rectangular removal** removes a set of request locations in a predefined area in the coordinate system. This removal operator is similar to the zone removal used in Demir et al. [16]. First, the corner point of the entire study area is computed. Then, the study area is partitioned into nine smaller regions. A region is randomly chosen, and all its request locations that appear in vehicle routes are removed from those routes. A new region is selected if the chosen area does not contain any locations. Note that this operator does not necessary remove exactly r request locations from the route plan.

9. **Circle removal** removes request locations from a predetermined circular area. More precisely, the operator first computes the average distance among all locations $t_{avg} = \sum_{i=0}^{n} t(i, i+1)/n$. Then, a request location is selected randomly as the center of a circle with radius t_{avg}. All request locations located within this circle are removed. This operator is inspired from the rectangular removal. Like the previous removal operator, less or more than r request locations might be removed.

3.4.3.2 Insertion operators

After the application of a removal operator, a partial solution exists. A partial solution describes a routing plan for which some of its request locations have been removed. The goal of insertion operations is to repair the partial solution by finding feasible insertion locations for some of the requests in the unvisited list u. Note that u contains both requests that have been removed, and any requests that were already in the unvisited list before removal began. The insertion operations used are described below.

1. **Greedy insertion** inserts the unvisited request with maximum demand into its cheapest feasible insertion location. This operator focuses on both demand and travel time when inserting requests. The demand criterion is used to determine which request location to insert while the distance criterion is used to determine where to insert the selected request in the plan. Therefore, the unvisited requests in u are sorted with respect to the magnitude of demand, and the request i with the highest demand is the first attempt to be inserted in the position where it yields the lowest increase in the length of the tour between nodes j and $j+1$ such that $c_i(j, j+1)$ is minimal. This operator repeatedly chooses the next highest demand location and calculates the insertion costs after each iteration and inserts the request into the cheapest possible position of the routes.

2. **Regret-2 insertion** tries to improve the shortsighted approach of greedy insertion by considering a regret measure [12]. For a request location, the regret value is the difference between the insertion costs associated with the best and second best positions for a customer i. More precisely, Regret-2 operator chooses to insert request i among u such that $\{\Delta f_i^2 - \Delta f_i^1\}$ is maximal, where Δf_i^h is the cost of inserting a request i at the h^{th} cheapest position.

3. **Demand per distance insertion** improves a shortcoming of the other two insertion operators by taking demand and distance into consideration simultaneously. Let $D_v^h(i)$

be the demand per distance associated with request i in the vehicle route v where h represents the insertion place. The operator can be summarized as follows: insert a request location i into a position h of the vehicle route v which brings the largest demand per distance $D_v^h(i)$ such that $\{d_i/c_i(j,k)\}$ is maximal where position h is between nodes j and k. The heuristic first sorts the requests according to $D_v^h(i)$, chooses the largest one for which to attempt insertion, and then recalculates $D_v^h(i)$ for the remaining requests if the selected request is inserted. The process is repeated until no feasible insertion place remains.

3.5 Experimental settings

In this section, we first introduce the benchmark instances used in this chapter to validate the performance of the proposed ALNS algorithm. Then in Section 3.5.1, we describe how the parameters of ALNS are tuned. In Section 3.5.2, we present initial results of the computational experiments used to assess the performance of ALNS. Finally, Section 3.6 introduces a case study developed using real data from the Haiti earthquake.

The ALNS is implemented in Java and executed on an Intel Core i7 CPU at 3.33GHz with 24GB of RAM. The algorithm validation is performed on a set of 387 benchmark instances from Chao et al. [1], in which the number of nodes varies between 21 and 102 with two, three, and four vehicles. All benchmark instances are available from `http://www.mech.kuleuven.be/en/cib/op`.

3.5.1 Parameter tuning

In order to achieve good performance, the parameters of ALNS must be tuned. We follow the methodology presented by Ropke and Pisinger [12]; one parameter at a time is tuned while the others remain fixed. The best settings are chosen to minimize average deviation from the best-known solutions. We have chosen eight difficult instances (p1.3.l, p3.2.j, p4.2.i, p4.4.p, p5.4.r, p6.2.d, p7.2.o, p7.3.n) among the TOP benchmark instances based on our preliminary test results in order to set the main ALNS parameters. Five random replicates of each of the eight instances are performed.

Table 3.1: Values for ALNS parameters after the tuning phase

Parameters	Tuning Values
Removal Parameters $[a, b]$	
Lower limit of percentage of destruction (a)	5
Upper limit of percentage of destruction (b)	40
Simulated Annealing Parameters	
Temperature	1000
Cooling Rate (α)	0.90
Iteration Parameters	
Number of warm up iterations	2000
Percent of stopping iterations (if no improvement)	60
Number of ALNS iterations	15000
Scoring Parameters	
$(\sigma_1, \sigma_2, \sigma_3)$	(50, 20, 0)

The proposed ALNS approach contains four groups of parameters as shown in Table 3.1: (1) removal parameters: what percent of request locations to remove in each iteration is randomly chosen in the range $[a, b]$; (2) simulated annealing parameters control the acceptance criteria using two parameters, *temperature* and *cooling rate*, α; (3) ALNS iteration parameters: the number of warm-up iterations in the first segment to collect statistics about the operators, and the number of stopping iterations if no improvement can be found; (4) scoring parameters: the scores show how well the removal and insertion pairs have performed in the past iterations. Table 3.1 illustrates the best values found for all parameters. This setting has been used for the following experiments in this study.

3.5.2 Heuristic validation

In this subsection, we discuss the results of experiments carried out to investigate the performance of ALNS. For each benchmark problem, we compare the results obtained with ALNS to best-known solutions given in Dang et al. [8]. Table 3.2 summarizes the average performance of ALNS for 7 instance classes by computing the average percentage gap from best-known solution values. The results represent the average performances of ALNS over five replications. The first column presents the data set names p1 - p7, the second column shows the number of problem instances, the third column shows the number of best-known solutions missed, the third and fourth columns present the maximum and average gaps from best-known solutions, respectively, and the last column gives the average CPU time in seconds for each problem set. ALNS finds most of the best-known solutions (327 out of 387) and provides a competitive average gap (0.11%).

Table 3.2: Average performance of the ALNS

TOP - Data Set	# Instances	# Best Known Missed	Average ALNS Gap		Average CPU
			Max	Avg	
p1	54	0	0.00%	0.03%	4.1
p2	33	0	0.00%	0.00%	1.4
p3	60	0	0.00%	0.00%	6.8
p4	60	34	0.43%	0.73%	247.6
p5	78	7	0.14%	0.29%	50.2
p6	42	0	0.00%	0.00%	29.0
p7	60	19	0.23%	0.45%	176.7
Average			0.11%	0.22%	73.7

In order to further investigate the quality of the proposed algorithm, we compare our algorithm with the following 10 state-of-art algorithms in the literature:

- TMH: tabu search algorithm of Tang and Miller-Hooks [4]
- TSF: tabu search algorithm of Archetti et al. [5]
- SVNS: variable neighborhood search of Archetti et al. [5]
- ACO: ant colony optimization of Ke et al. [6]

- MA: memetic algorithm of Bouly et al. [21]
- SPR: slow path relinking of Souffriau et al. [22]
- DPSO: discrete particle swarm optimization (PSO) of Muthuswamy and Lam [23]
- PSOMA: PSO-based memetic algorithm of Dang et al. [24]
- PSOiA: effective PSO-inspired algorithm of Dang et al. [8]
- AugLNS: augmented large neighborhood search algorithm of Kim et al. [9]

Table 3.3: Summary of the best-performing TOP algorithms

Method	Year	Avg Gap	# Best Known	Avg CPU for all instances
TMH	2005	1.32%	34	336.6
TSF	2007	0.20%	94	531.5
SVNS	2007	0.04%	134	156.1
ACO	2008	0.09%	128	16.6
MA	2010	0.04%	129	36.9
SPR	2010	0.05%	126	21.2
DPSO	2011	1.24%	39	n/a
PSOMA	2011	0.02%	146	22.3
PSOiA	2012	0.00%	156	59.8
AugLNS	2013	0.00%	156	85.1
ALNS	2015	0.31%	98	73.7

Table 3.3 summarizes the performance of the state-of-art algorithms. This comparison is based on 157 benchmark instances of sets p4, p5, p6, and p7 as in Souffriau et al. [22]. The last column presents the average CPU time in seconds for all 387 instances. As shown in this table, PSOiA and AugLNS outperform the other methods in the literature in terms of average gap. The results indicate that the proposed algorithm provides promising results by finding best solutions for 98 out of 157 benchmark instances and having an average gap of 0.31%. The ALNS algorithm outperforms TMH, TSF, and DPSO in terms of the number of best-known solutions found. On average, the performance of ALNS is better than TMH, TSF, SVNS and AugLNS in terms of computational time.

Since the number of customers in this case study instances is much larger (more than 300 customer locations on average) than the TOP benchmark instances and also developing

Table 3.4: Average performance of the fast ALNS

TOP - Data Set	# Instances	# Best Known Missed	Average ALNS Gap Max	Average ALNS Gap Avg	Average CPU
p1	54	0	0.00%	0.34%	0.7
p2	33	0	0.00%	0.11%	0.2
p3	60	0	0.00%	0.22%	0.9
p4	60	34	0.50%	1.39%	30.9
p5	78	14	0.21%	0.87%	6.9
p6	42	0	0.00%	0.08%	4.6
p7	60	43	0.29%	0.78%	21.8
Average			0.14%	0.54%	9.4

routing plans quickly in real disaster situations can be critical in saving lives, we reduce the number of iterations from 15,000 to 2,000. After this significant reduction in the number of iterations, we assess the performance of the fast ALNS algorithm, shown in Table 3.4, by comparing the average gap from best-known solution values before using it for the case study. Tables 3.2 and 3.4 show that, on average, the runtime of ALNS is decreased from 73.7 seconds to 9.4 seconds while the gap between the best-known solutions and ALNS increases slightly from 0.11% to 0.14%. The results indicate that the computational time of ALNS is significantly decreased without sacrificing much in the solution quality, and thus, ALNS with the reduced number of iterations can be implemented for solving the case study instances.

3.6 Case study development

To develop a disaster tour plan for a case study, the model requires two sources of information: (i) actionable social data posted to social media platforms and (ii) verified data collected in traditional ways (e.g. on-the-ground assessment teams). The case study developed from the 2010 Haiti earthquake is used to test the alternative decision strategies because of the availability of both types of data.

3.6.1 Social data

The social data posts are available from the Ushahidi platform – a citizen event reporting platform – for the Haiti earthquake [25]. A dataset containing 3,593 user-generated posts collected during a 97-day period after the disaster were initially filtered by Ushahidi. We chose to work with only the first 15 days of the posts (01/12/2010 - 01/21/2010), which represent 71% of the data. The data is further filtered by only considering the capital city, Port-au-Prince, which is the most populated city in Haiti, and near (approximately 16 miles) the epicenter of the earthquake. The social data include requests for help, comments, opinions, and offers of help. Even though the posts contain relevant data, actionable information, which describes a specific need, quantity, and location, is needed in the response-planning process.

LATITUDE	LONGITUDE	DATE	CATEGORY	MESSAGE
19.762503	-72.20571	01/16/2010	informative	Hospital Justina in Cap-Haitien is open.
18.514162	-72.283988	01/16/2010	actionable	Marilou Roy has 110 people at her house who need water, food, medicine
18.523603	-72.292914	01/16/2010	partially actionable	Hotel Montana in urgent need of water.
18.583333	-72.266667	01/16/2010	informative	Universal super market is open on tabarre
18.383333	-73.1	01/16/2010	partially actionable	Deita Erasme and a few others need rescue/water at her home
18.529598	-72.408061	01/16/2010	actionable	Orphanage in carrefour in dire need! 48 kids will wake up tomorrow to find no food
18.525678	-72.338448	01/16/2010	partially actionable	People trapped under the house. Screaming was heard.

Figure 3.2: Snapshot of social data

In most cases, social media data are unstructured and high-quality data are not usually available. Therefore, the dataset is extracted and actionable requests are identified by manually approving or denying each message as shown in Figure 3.2 where we populated the "category" column ourselves. However, all attributes of a request – location, type, and quantity – may not always be obtained in order for a request to be classified as actionable, and thus they are classified as partially-actionable. Table 3.5 shows the number of actionable and partially-actionable social media requests in Port-au-Prince after manually filtering the data. For example, day 10 includes a total of 50 actionable and 90 partially-actionable requests.

In the dataset, partially-actionable data points usually do not specify the magnitude of

Table 3.5: Number of filtered social data locations in Port-au-Prince

Day	**Dates**	**#Actionable**	**#Partially-actionable**	**Total #Actionable**
1	01/12/2010	1	1	2
2	01/13/2010	5	0	5
3	01/14/2010	3	3	6
4	01/15/2010	4	5	9
5	01/16/2010	20	16	36
6	01/17/2010	24	26	50
7	01/18/2010	26	26	52
8	01/19/2010	13	48	61
9	01/20/2010	25	48	73
10	01/21/2010	50	90	140
11	01/22/2010	38	89	127
12	01/23/2010	58	112	170
13	01/24/2010	24	48	72
14	01/25/2010	12	20	32
15	01/26/2010	12	15	27
Total		315	547	862

Table 3.6: Partially-actionable data

Group	Demand Size
Single family group or smaller family, house, people, babies, children, several people	5
Mid-size and/or multi-family groups families, church, orphans, students, clinic, nursing home, center	20
Large groups hotel, hospital, school, neighborhood	100

their requests (e.g. number of people in need at the specified location). These values tend to vary highly. Therefore, a specific structure is needed in order to estimate the demand on the partially-actionable locations. In order to estimate demand at these locations, partially-actionable requests are categorized under three major groups (small, medium and large) based on the anticipated group size determined from the text in the post, as shown in Table 3.6. According to a survey conducted in 2008 in Port-au-Prince, the average household size was 4.5 individuals [26]. Therefore, we assume that a single family or smaller group has the average demand size of 5, while mid-size family and larger groups have demand sizes of 20 and 100, respectively. For example, in Figure 3.2, a data point describes a family; "Deita Erasme and a few others need rescue/water at her home." Since the number of people in need was not specified, we can classify this request as a single family and assume there are 5 people in need at this location. Once the demand on the partially-actionable locations has manually been estimated, they are re-classified as actionable, as seen in Table 3.5.

3.6.2 Traditional data

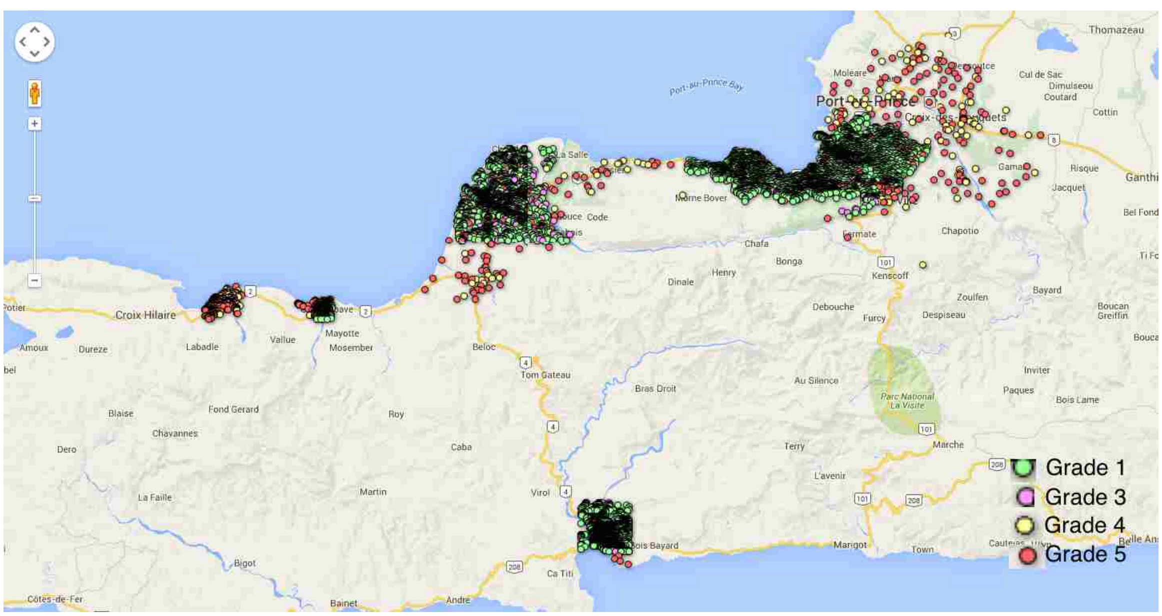


Figure 3.3: Haiti - Damaged buildings - Colors represent different grade levels

In order to represent data from trusted traditional sources, we use building damage assessment datasets led and carried out by the Government of Haiti following the Haiti earthquake [27]. The damage assessment was performed based on satellite imagery and aerial photos. Figure 3.3 shows this data set includes maps of 294,170 buildings. Each building was classified on a scale of 1 to 5, where the damage level increases from 1 to 5. For example, Grade 1 (green points) represents negligible to slight damage, while Grade 5 (red points) represents total destruction. Note that none of the buildings were classified as Grade 2.

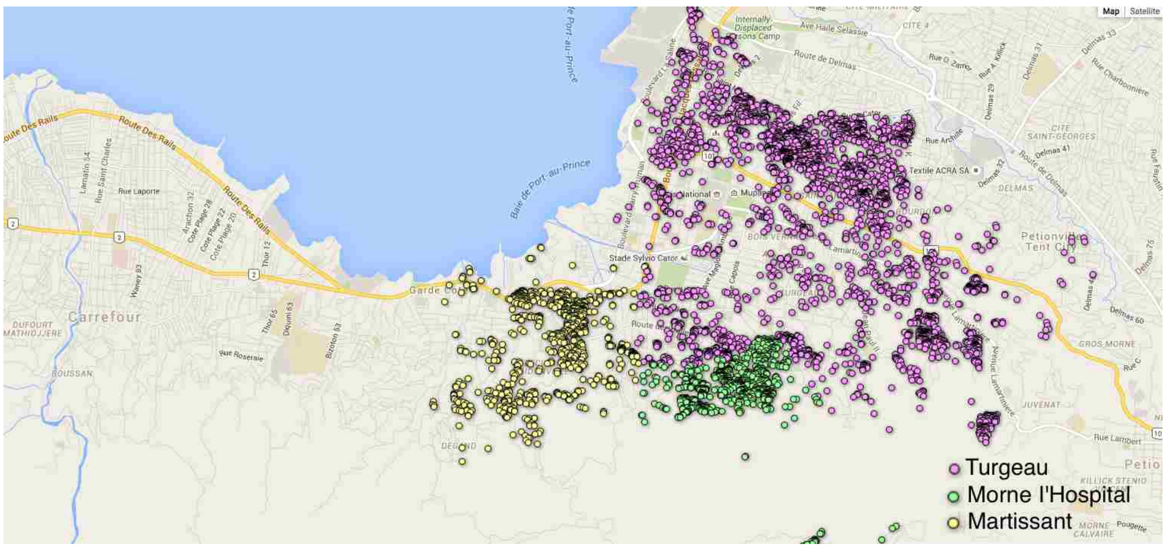


Figure 3.4: Port-au-Prince - Grade 3 damaged buildings - Colors represent different regions

In this study, we only focus on Grade 3 with 15,737 buildings because of the potential for survivors. We assume it is unlikely for emergency organizations to receive a request from very heavily damaged buildings (Grade 4 and Grade 5). Similar to the social data, this dataset is also further filtered by only considering Port-au-Prince. Therefore, there are 5,230 buildings left after filtering, as shown in Figure 3.4. Demand locations and magnitudes are estimated from these Grade 3 buildings to represent traditional data by considering the extent of building damage as follows.

Estimating the number of traditional data locations: To estimate the number of traditional locations, a simple assumption is used to make population (number of locations) estimates because traditional data was not readily available for this case study. However, immediate resource requirements could be found after Hurricane Katrina in 2005 [28]. Here we assume that traditional data in Port-au-Prince follows the same temporal demand distribution as Hurricane Katrina. According to a report that analyzed requests for resources after Hurricane Katrina, the temporal distribution for a total of 630 requests for the first 65 days is as shown in Figure 3.5 [28]. Therefore, damage assessment datasets are divided into 65 days similar to the Hurricane Katrina data, and the number of damaged buildings each day is estimated based on the pattern of demand observed after Hurricane Katrina. The probability of demand on a day can be calculated as n_{day}/n_{total} according to resource requirements following Hurricane Katrina where n_{day} and n_{total} represent the number of requests on a day and the total number of requests for the first 65 days in Hurricane Katrina. The demand seeking population each day is computed from those probabilities. For example, as seen in Figure 3.5, 18 demand requests are available on day 2. Since the probability of demand on day 2 is 18/630 and a total of 5230 buildings is available for the traditional data, the number of requests can be approximated on day 2 for the Haiti earthquake as 149 ($18/630 \times 5230 = 149.3$). There are 3,746 damaged buildings in the first 15 days after approximating the number of requests per day. The first two columns in Table 3.7 represent the days after the disaster and the number of social requests, respectively. The third column in Table 3.7 shows the number of request locations estimated using this method for traditional datasets. For example, as seen in Figure 3.5, demand requests peaked on day 5 and began to decrease over time. Similarly in Table 3.7, the expected number of traditional requests peaks on day 5 with 465 requests and then follows a downward trend. The fourth column in the table provides the total number of requests that contain social and traditional data for each day. The last column gives the proportion of requests that are social versus traditional. Letting ρ represent the proportion of requests that are social, ρ varies between 0.03 and 0.61 in this study. On days 2, 3, and 4, most of the requests are traditional requests because $\rho < 0.05$, while day 12 includes a large number of social requests with $\rho = 0.61$.

Designating traditional demand locations: After estimating the required number

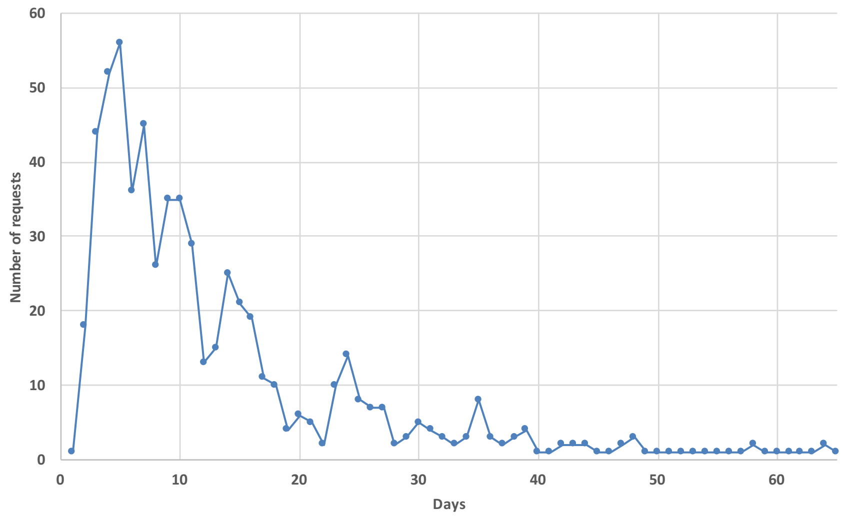


Figure 3.5: Number of requests per day in Hurricane Katrina

Table 3.7: Number of traditional and social requests

Day	# of social requests	Expected # of traditional requests	Total # of requests	Proportion of social requests (ρ)
1	2	8	10	0.20
2	5	149	154	0.03
3	6	365	371	0.02
4	9	432	441	0.02
5	36	465	501	0.07
6	50	299	349	0.14
7	52	374	426	0.12
8	61	216	277	0.22
9	73	291	364	0.20
10	140	291	431	0.32
11	127	241	368	0.35
12	170	108	278	0.61
13	72	125	197	0.37
14	32	208	240	0.13
15	27	174	201	0.13

of traditional locations for 15 days, a set of distinct traditional request locations needs to be selected for each day among the 5,230 buildings. Two different selection methods for the traditional locations are considered: (1) Random – each traditional location is chosen at random, and (2) Uniform-Clustered – some traditional locations are clustered visually by considering the density of the locations. More precisely in the Uniform-Clustered method, at first, relatively dense request locations (small clusters) are identified visually. Next, the number of clusters are reduced by combining three or four small cluster groups together until the clusters are relatively very far apart from one another. This results in five clusters consisting of two medium and three high density clusters as shown in Figure 3.6. The rest of the locations that are outside of these clusters forms un-clustered request locations with low density.

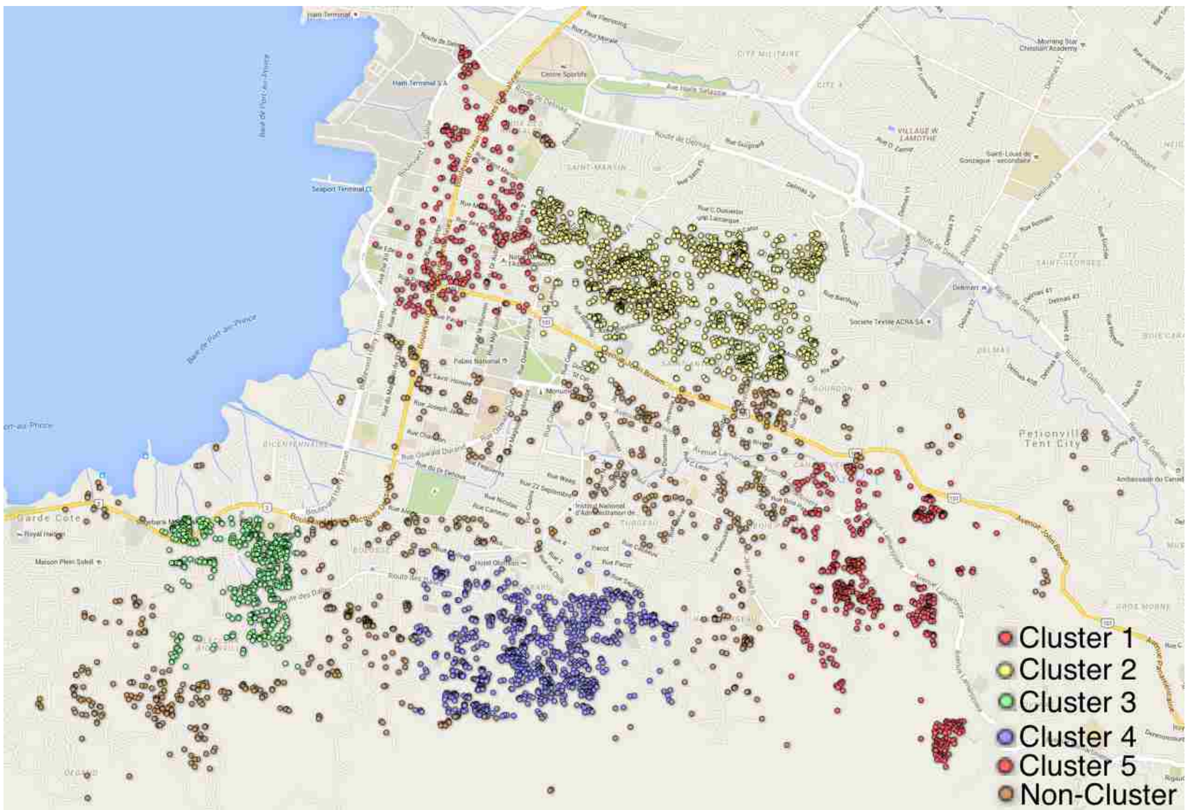


Figure 3.6: Clustered Traditional Locations - Colors represent different clusters

Table 3.8 illustrates the cluster and individual request location probabilities designated based on the density found in each cluster. The first column represents the assigned name of the cluster groups, while the second column gives the number of locations in that cluster. The density of clusters is calculated based on the area of a cluster and the number of requests in this cluster, and each cluster is classified as either low, medium or high density in the third column. Clusters with small regions have a higher request density than their large

Table 3.8: Probability of selecting a request location from a cluster

Cluster Group	# of locations	Density	Location prob.	Cluster prob.
1	1575	high	1/4500	0.35
2	521	high	1/4500	0.12
3	1094	high	1/4500	0.24
4	338	medium	1/6500	0.05
5	692	medium	1/6500	0.11
non-cluster	1010	low	1/7614	0.13
Total	5230			1.00

region counterparts. The "Location prob." column represents a probability of selecting a request location from a cluster group. In the last two columns, the probabilities for each request location are selected in such a way that the sum of the cluster probabilities equals 1. Let p_H, p_M, and p_L be the individual request probabilities of high, medium, and low density clusters. Location probabilities are estimated as follows.

$$\frac{p_L \times (1575 + 521 + 1094) + p_M \times (1094 + 335) + p_L \times (1010)}{5230} = 1, \tag{3.1}$$

where $p_H > p_M > p_L$. Thus, p_H, p_M, and p_L are estimated as 1/4500, 1/6500, and 1/7614, respectively. Selecting a request location from a cluster depends on the number of locations and density in that cluster. For example, because of its high-density cluster probability of 0.35 (1575 $\times$ 1/4500), selecting a request location from Cluster 1 as a traditional data point is more likely than other clusters.

Estimating magnitude of traditional demand locations: Demand magnitudes are known for social data request locations, but they are uncertain for traditional request locations. In order to estimate the amount of demand at each traditional location, a simplifying assumption is to consider the size of buildings and population density. However, the building dataset does not contain the shape files [27]. To estimate uncertain demand magnitudes, we used probability distributions at each traditional request location. One of the most commonly used probability distributions is the triangular distribution, which requires three inputs: the minimum, most likely, and maximum values. These parameters can be determined quantitatively.

Table 3.9: Region populations of Port-au-Prince in 2009

Section of Port-au-Prince	Population	Total # Damaged Buildings	Number of people per building
Turgeau	478,244	55,244	8.66
Morne I'Hospital	152,105	10,173	14.95
Martissant	267,510	27,323	9.79

As seen in Figure 3.4, there are three regions in Port-au-Prince; Martissant (represented by green), Morne L'hopital (represented by yellow), and Turgeau (represented by purple). We can estimate a demand magnitude associated with each building based on the population density of the specific region where the building is located. Table 3.9 shows the population, total number of buildings given in the damage assessment dataset and the population density per building in a region. We update triangular distribution parameters for each region to capture the realistic demand scenarios. Table 3.10 presents the estimated parameters for each traditional request location in a region. We can estimate the lower demand as a single unit of demand for every request location. Demand values for the most likely are assumed to be equal to the population density per building in a given region. Therefore, demand values of 9, 15, and 10 would be more likely to occur for Turgeau, Morne I'Hospital, and Martissant, respectively. Since Port-au-Prince has very few high-rise buildings, we assume that the resident apartments are a maximum five stories. Thus, the upper demand magnitude would be approximately five times the number of people per building as shown in Table 3.10.

Table 3.10: Parameters of Triangular Distribution for each region

	Lower Limit (min)	Mode (most likely)	Upper Limit (max)
Turgeau	1	9	45
Morne I'Hospital	1	15	75
Martissant	1	10	50

3.6.3 Case study instances

A construction effort is needed to create a set of instances for case study scenarios resembling real-world situations. The performance of alternative social data strategies is assessed using various disaster scenarios generated for the case study using the methods described in Section 3.6. Table 3.11 summarizes these methods. In the experimental design, there are four main factors with some sub-factors that vary across scenario instances. Each factor has a number of levels. For example, to designate traditional request locations in a test instance, two alternative factors are considered; random or uniform-clustered. The first three factors are related to traditional data generation while the last factor is used to modify social data. The performance of alternative strategies depends on what proportion of requests are accurate. Let λ denote the proportion of social data locations that are accurate. Since the accuracy of a request communicated through social media is not available in the dataset, we need to assess the impact of varying this parameter, therefore we consider $\lambda \in \{0.33, 0.67\}$.

During the first week of the response in the Haiti earthquake, there was a lack of fuel for vehicles in Port-au-Prince, and the relief workers on the ground didn't have enough trucks to fulfill demand [29]. Therefore, in this study, we consider two different vehicle fleet sizes: one that assumes trucks are scarce and only two vehicles are available, and the other one that assumes more trucks, five vehicles, are available to deliver lifesaving supplies to the affected population.

In this study, each instance variant is created in the following manner. Each scenario is denoted using four letters $W.X.Y.Z$, where the first letter W denotes the scenario day and can vary from d1 up to d15 (e.g. d1 represents day 1). The second letter X denotes the number of vehicles m and can either be m2 for two vehicles or m5 for five vehicles. The third letter Y denotes the proportion of social locations that are accurate and can be either $l0.33$ for $\lambda = 0.33$ and $l0.67$ for $\lambda = 0.67$. The last letter Z denotes the method used to designate traditional locations and can be either random (rnd) or clustered (cl). For example, scenario $d12.m5.l0.33.rnd$ represents an instance in which on day 12, five vehicles

Table 3.11: Experiment development

Factor	# of Levels
Estimating Number of Traditional Locations	1
(1) Expected distribution of resource requirement	
Designating Traditional Request Locations	2
(1) Random	
(2) Uniform - Clustered	
Estimating Traditional Demand	1
(1) Triangular (min, most likely, max)	
Accuracy of Social Data	2
(1)-(2) Proportion of social requests that are accurate	λ={0.33, 0.67}

are available to execute the tours, 33% of social locations are accurate, and traditional locations are placed randomly. From each of these scenarios, eight resulting combinations are considered. Additionally, the first 15 days of the Haiti earthquake are studied for each factor combination, yielding a total of 120 scenario instances (8×15).

The performance of alternative strategies is further analyzed in the computational study by grouping scenarios according to request volume. Three levels of volume of total requests are considered, with low indicating fewer than 250 requests, medium indicating 250 to 400 requests, and high indicating more than 400 requests. Two scenario days for each request volume are chosen by considering the proportion of requests that are social data with parameter $\rho \approx 0.13$ and $\rho \approx 0.34$. Note that days with $\rho < 0.10$ are not considered because there are relatively fewer number of requests that are social. Also, since only a single day (day 12) has $\rho > 0.40$ and cannot be grouped with other days, it is ignored. Table 3.12 illustrates the selected 6 out of 15 days by grouping them according to levels of number of requests where day 13 with $\rho = 0.37$ and day 15 with $\rho = 0.13$ are classified as low, day 6 with $\rho = 0.14$ and day 11 with $\rho = 0.35$ are classified as medium and day 7 with $\rho = 0.12$ and day 10 with $\rho = 0.32$ are classified as high.

Table 3.12: Request volume

Request Volume	day	# of locations	# of social	# of traditional	ρ
Low	13	197	72	125	0.37
	15	201	27	174	0.13
Medium	6	349	50	299	0.14
	11	368	127	241	0.35
High	7	426	52	374	0.12
	10	431	140	291	0.32

In this case study, a single depot location is available to distribute relief supplies to the affected population. During the disaster, a large proportion of buildings had heavy damage which creates a constraint to find a distribution center. To solve this issue, an open space warehouse was set up in the presidential palace gardens in Port-au-Prince after the earthquake [29]. Therefore, we assume that a single depot is available in the presidential residence. The number of vehicles is fixed with unlimited capacity, and each location can be visited at most once. Each route begins and ends at a depot within a time limit. Since we assume that the service time of requests can be a maximum of 25 minutes, the service rate is assumed to be 5 min/demand, but the service time of a request location is assumed to be no more than 25 minutes. Also, the vehicle speed is assumed to be 25 km per hour because Port-au-Prince is a highly populated city with narrow streets and poor road infrastructure, so traffic is chaotic and congested after the earthquake [29]. We further assume that each request location needs to be served within a 12-hour period such as 8:00 am to 8:00 pm in order to minimize human suffering.

Distance matrix: In this case study, the driving distances are calculated using ArcGIS from ESRI, a geographic mapping software, to create a matrix of distances between pairs of request locations and the depot. ArcGIS is a complete system for designing and managing solutions through the application of geographic knowledge [30]. The road network of the case study is created in ArcGIS using OpenStreetMap data, which is an open and freely available database of geographic data, for Haiti and the Dominican Republic [31]. This road network contains all roads, highways, and avenues that accurately represent the network requirements in the case study. The distance matrix for a set of social and traditional locations is generated using the origin-destination (OD) cost matrix analysis, which calculates the least-cost network paths from origins to destinations, in the ArcGIS Network Analyst [32]. Latitude and longitude coordinates of a total of 6092 locations (862 social and 5230 traditional locations) in Port-au-Prince are uploaded to ArcGIS to create a distance table. Note that the distances between the locations are calculated in kilometers. A total of 35,010,889 pairs (5917×5917) of distances is calculated by the OD cost matrix analysis where driving distances for 175 traditional request locations are ignored because of the missing route network connectivity. Therefore, those locations are not included in the case study instances.

3.7 Computational results

The discussion of results will focus on the summary performance of the social data decision strategies. The results of the computational study are summarized in Table 3.13. Traditional locations are designated by either random or cluster selection methods; the number of vehicles is $m = \{2, 5\}$, and the proportion of social requests that are accurate is $\lambda = \{0.33, 0.67\}$. Table 3.13 reports the average percentage gap between the demand served for a decision strategy and the total accurate demand in each 15 days scenario. Also, the table provides the number of days in which a decision strategy is preferred. In Table 3.13, the #days column shows the number of days in a scenario set, and the Gap column illustrates the percentage of unsatisfied demand. The #PS column presents the number of days in which each decision strategy is preferred. However, the #PS column for a given scenario will not always sum to 15 because, for some instances, alternative strategies serve an equal amount of accurate demand, and therefore, any of them can be preferred.

Among the decision approaches, *All* has a lower average gap (40.3%) than the other strategies, indicating better performance. $OnlyVerified$ provides an average gap of 60.2%

Table 3.13: Summary results

Placement of Traditional Data	λ	m	#days	*All*		*OnlyVerified*		*FirstVerified*		*PrioritizeVerified*		*CloseProximity*	
				Gap	#PS	Gap	#PS	Gap	#PS	Gap	#PS	Gap	#PS
Random	0.33	2	15	55.3%	12	71.0%	3	71.0%	3	71.0%	2	71.0%	3
	0.33	5	15	28.3%	11	45.8%	2	41.1%	2	39.0%	3	41.0%	2
	0.67	2	15	46.6%	15	76.1%	0	76.1%	0	76.2%	1	76.1%	1
	0.67	5	15	24.3%	14	55.4%	0	48.2%	1	44.9%	2	50.2%	1
	summary		60	38.6%	52	62.1%	5	59.1%	6	57.8%	8	59.6%	7
Clustered	0.33	2	15	61.8%	11	67.3%	1	67.3%	2	64.2%	4	67.3%	2
	0.33	5	15	31.4%	12	38.7%	0	38.6%	1	37.7%	4	38.6%	2
	0.67	2	15	49.2%	15	74.7%	0	74.7%	1	74.7%	1	74.7%	1
	0.67	5	15	25.5%	14	52.6%	0	48.6%	1	44.3%	2	47.9%	1
	summary		60	41.9%	52	58.3%	1	57.3%	5	55.2%	11	57.1%	6
	Overall		120	40.3%	104	60.2%	6	58.2%	11	56.5%	19	58.4%	13

while the other strategies, *FirstVerified*, *PrioritizeVerified*, and *CloseProximity*, outperform *OnlyVerfied* with 58.2%, 56.5%, and 58.4% average gaps, respectively. This result is not surprising because solutions produced by *OnlyVerified* do not include social data in vehicle tours. Therefore, satisfied demands are fewer than those produced by other approaches, which consider social data at least in some capacity. This result can also be confirmed based on the number of preferred approaches. It is interesting to note that *All* is preferred in more than three-quarters of the instances in each scenario group considered. Table 3.13 indicates that *All* is the preferred strategy in 104 out of 120 instances considered, while *OnlyVerified*, *FirstVerified*, *PrioritizeVerified*, and *CloseProximity* are the preferred strategies only in 6, 11, 19 and 13, respectively. This indicates a fairly robust preference for *All* across a wide variety of scenarios.

Besides *All*, for the other alternative strategies, the preference for *PrioritizeVerified* is more pronounced because unlike *FirstVerified*, *PrioritizeVerified* allows visiting social data requests that can appear anywhere within a vehicle route. As an emergency manager assigns a demand-dependent weight to traditional request locations in this approach, a social request with high demand can replace a traditional request with low demand. This replacement can result in increasing the amount of demand served. In general, the performance of *CloseProximity* is better than *OnlyVerified* and *FirstVerified*. While all these strategies consider only traditional locations first, unlike the other two strategies, $CloseProximity$ only inserts social data locations in close proximity to the tour. Thus, this use of social data increases the total demand served by $CloseProximity$, but the total response time increases only slightly. Also, it is not surprising that the preference for $FirstVerified$ is more pronounced than for $OnlyVerified$ because the former approach allows visiting social data locations at the end of the tour if all traditional locations have already been visited, while the latter approach always ignores social data requests.

The performance of the alternative strategies are further summarized in Table 3.13 by grouping scenarios as either random or clustered according to the placement of the traditional request locations. The preference of *All* seems to be robust in both cases as *All* is preferred in 52 out of 60 scenario instances. However, the performance of *All* tends to worsen as the placement of the traditional requests are changed from random to clustered, while the performance of other strategies tends to improve. On the other hand, the tours produced by other strategies get shorter because traditional locations are located near each other and a higher priority is given to those locations when developing routing plans which can result in decreasing the percent of unvisited customers. For example, the percentage of unsatisfied demand of a solution produced for *All* increases from 38.6% to 41.9% when the traditional locations are changed from random to clustered. In contrast, the percentage of unvisited customers of a solution produced for $OnlyVerified$ decreases from 62.1% to 58.3%. However, it needs to be noted that the relative performance of *All* is better than the other alternative strategies when the placement of traditional locations are either random or clustered. When traditional locations are changed from random to clustered, the preference for *All* stays the same (52 instances) while the preferences for $OnlyVerified$, $FirstVerified$ and $CloseProximity$ are less pronounced (they are preferred in 1, 5, and 6 instances, respectively), despite the fact that the percentage of unserved demand for those approaches decreases. However, the preference for $PrioritizeVerified$ improves from 8 instances to 11 instances when traditional locations are changed from random to clustered.

This is because the distance traveled between traditional locations decreases which allows visiting more social locations with high demand.

As seen in Table 3.13, the performance of *All* generally improves as the proportion of social data that are accurate increases, whereas the performance of other strategies tends to worsen. Since all request locations are intended to be included in routes produced by *All*, as more social data locations become accurate, the total amount of demand served by solutions produced by *All* increases. However, as the total accurate demand in a given scenario becomes higher, the percentage of unsatisfied accurate demand of a solution produced by other strategies increases, because social data either is not considered or is only considered and acted on in a limited way. For example, when the placement of traditional data is random, only two vehicles are available and λ increases, the percent of unsatisfied demand decreases from 55.3% to 46.6% for *All* and increases from 71.0% to 76.1% for $OnlyVerified$.

As expected, when more vehicles become available to deliver relief supplies to the affected population, the amount of unsatisfied demand decreases. In general, this reduction is most pronounced for $PrioritizeVerified$, because more social data requests with a high amount of demand can be inserted into their vehicle tours. However, this reduction is less pronounced for $OnlyVerified$, because all traditional locations may be visited by a fewer number of vehicles while other vehicles remain idle. For example, when the traditional locations are random and $\lambda = 0.33$, the percentage of unserved demand of a solution produced by $PrioritizeVerified$ decreases from 71.0% to 39.0% whereas this reduction is only from 71.0% to 45.8% for $OnlyVerified$.

Table 3.14 further summarizes the performance of decision strategies by grouping scenarios (only 6 out of 15 days) according to request volume where scenario days 15 and 13 are classified as low, scenario days 6 and 11 are classified as medium, and scenario days 7 and 10 are classified as high, as shown in the first and second columns. The ρ column illustrates the proportion of total requests that are social data is $\rho \approx \{0.13, 0.34\}$. The λ column illustrates the proportion of social data requests that are accurate is $\lambda = \{0.33, 0.67\}$. The #instances

Table 3.14: Performance of decision strategies according to request volume

Request Volume	day	ρ	λ	#instances	*All*		*OnlyVerified*		*FirstVerified*		*PrioritizeVerified*		*CloseProximity*	
					Gap	#PS	Gap	#PS	Gap	#PS	Gap	#PS	Gap	#PS
Low	15	0.13	0.33	4	27.37%	2	38.91%	1	38.91%	1	22.32%	1	38.85%	0
	15	0.13	0.67	4	24.86%	4	46.68%	0	46.68%	0	46.72%	0	46.63%	0
	13	0.37	0.33	4	22.12%	2	24.76%	1	24.76%	1	21.75%	1	23.52%	1
	13	0.37	0.67	4	22.51%	4	34.64%	0	34.64%	0	27.19%	0	32.48%	0
	summary			16	24.21%	12	36.25%	2	36.25%	2	29.49%	2	35.37%	1
Medium	6	0.14	0.33	4	36.48%	2	65.77%	0	65.77%	0	65.76%	2	65.81%	0
	6	0.14	0.67	4	24.01%	4	77.95%	0	77.95%	0	77.93%	0	77.97%	0
	11	0.35	0.33	4	42.39%	4	47.67%	0	47.67%	0	47.71%	0	47.60%	0
	11	0.35	0.67	4	40.46%	4	53.31%	0	53.31%	0	53.33%	0	53.23%	0
	summary			16	35.83%	14	61.17%	0	61.17%	0	61.18%	2	61.15%	0
High	7	0.12	0.33	4	57.55%	4	60.08%	0	60.08%	0	60.10%	0	60.11%	0
	7	0.12	0.67	4	42.58%	4	63.24%	0	63.24%	0	63.25%	0	63.26%	0
	10	0.32	0.33	4	50.44%	4	65.53%	0	65.53%	0	65.54%	0	65.55%	0
	10	0.32	0.67	4	32.97%	4	73.18%	0	73.18%	0	73.19%	0	73.19%	0
	summary			16	45.88%	16	65.51%	0	65.51%	0	65.52%	0	65.53%	0
	Overall			48	35.31%	42	54.31%	2	54.31%	2	52.07%	4	54.02%	1

column presents the number of instances in each scenario. The Gap column shows the percentage of unsatisfied demand, and the #PS column reports the number of instances in which a strategy is preferred. Note that more than one decision strategy may be preferred for a given scenario instance. As seen in Table 3.14, the performance of decision strategies generally decreases as request volume increases. For example, the percent of unserved demand of solutions produced by *All* is 24.21%, 35.83 %, and 45.88% on average for low, medium, and high request volumes, respectively. *All* outperforms other alternative strategies in terms of both the percentage of unsatisfied demand and the number of preferred strategies when the region size is either low, medium or high. Table 3.14 indicates that *All* is preferred in 42 out of 48 such instances, while $OnlyVerified$, $FirstVerified$, $PrioritizeVerified$, and $CloseProximity$ are the preferred strategies only in 2, 2, 4 and 1 instances, respectively. The preference of *All* is even more distinct (*All* is preferred in 16/16 instances) when request volume is high. It is interesting to note that $OnlyVerified$ and $FirstVerified$ have the same performance in all selected scenarios. This may be because all traditional requests have not been visited by the routes developed for $FirstVerified$ in order to begin serving social requests. Therefore, a similar solution is produced by both $OnlyVerified$

and $FirstVerified$. The second best strategies in terms of the percentage of unsatisfied demand are $PrioritizeVerified$ with 29.49% when request volume is low, $CloseProximity$ with 61.15% when request volume is medium, and $OnlyVerified$ and $FirstVerified$ with 65.51% when request volume is high. As request volume increases, the tours developed by $OnlyVerified$, $FirstVerified$, $PrioritizeVerified$, and $CloseProximity$ more closely resemble each other. This may be because as request volume increases, more traditional requests are included in tours by these strategies, whereas *All* may include social and traditional requests evenly in tours.

The impact of increasing the proportion of social data requests (ρ) can also be seen in Table 3.14. In general, when the number of social requests increases from $\rho = 0.13$ to $\rho = 0.37$ and when request volume is either low or medium, the percentage of unsatisfied demand for all decision strategies decreases. For example, the percentage of unsatisfied demand for $PrioritizeVerified$ decreases from 46.72% to 27.19% when request volume is low, $\lambda = 0.67$, and ρ increases from 0.13 to 0.37. This may be because as ρ increases for low request volume, the potential amount of inaccurate demand also increases which may result in a smaller total amount of accurate demand for a given instance. However, when request volume is high, the percentage of unsatisfied demand for the decision strategies increases as ρ increases except for All. For example, when the scenarios have high request volume and $\lambda = 0.33$, ρ increases 0.12 to 0.32, the percentage of unsatisfied demand for All decreases from 57.55% to 32.97%. In contrast, the percentage increases for other strategies. This may be because the potential penalty associated with not traveling to social locations (in which social requests are either ignored or considered in a limited capacity) increases as the proportion of social data requests and the request volume increases.

Detailed results over five replications for the case study instances are given in Tables 3.15 and 3.16 for each alternative decision strategy. In these tables, the first column presents the name of the scenario instance. The second column reports the total amount of accurate demand of the scenario instance. The Max Demand Served column presents the maximum accurate demand served over five runs, and the Avg CPU reports the average computational time in seconds.

Table 3.15: Placement of traditional locations is random

Instance	Total Accurate Demand	All		OnlyVerified		FirstVerified		PrioritizeVerified		CloseProximity	
		Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU
d1.m2.l0.33.rnd	190	190	0.1	190	0.1	190	0.1	190	0.1	190	0.1
d2.m2.l0.33.rnd	3261	1984	109.9	1728	108.9	1728	116.2	1724	118.7	1714	104.6
d3.m2.l0.33.rnd	7643	2303	1499.6	2171	1547.3	2171	1608.8	2170	1658.4	2169	1612.0
d4.m2.l0.33.rnd	9349	2273	2965.8	2233	2937.5	2233	3167.5	2223	3136.3	2234	3168.3
d5.m2.l0.33.rnd	10334	2554	4186.8	2455	4150.5	2455	4929.3	2454	4916.2	2451	4769.0
d6.m2.l0.33.rnd	11727	7514	1183.5	2015	1114.5	2015	1178.0	1999	1262.7	2000	1183.6
d7.m2.l0.33.rnd	8114	2237	2425.4	2157	2340.7	2157	2551.0	2158	2601.0	2157	2543.0
d8.m2.l0.33.rnd	4466	1917	458.2	1966	478.1	1966	481.9	1966	608.7	1966	517.0
d9.m2.l0.33.rnd	7052	2459	1353.8	2142	1196.6	2142	1290.1	2143	1441.9	2143	1356.4
d10.m2.l0.33.rnd	9471	4852	2022.2	1982	2175.2	1982	2311.6	1982	2559.2	1981	2319.6
d11.m2.l0.33.rnd	5741	2248	1317.2	2032	1105.6	2032	1200.2	2041	1429.1	2050	1272.0
d12.m2.l0.33.rnd	8891	7629	523.4	1638	336.4	1638	363.7	1639	593.1	1638	370.0
d13.m2.l0.33.rnd	2856	1667	200.9	1633	122.2	1633	128.7	1626	232.7	1634	147.2
d14.m2.l0.33.rnd	4836	1998	316.8	2116	324.9	2116	347.8	2112	390.1	2131	344.3
d15.m2.l0.33.rnd	3783	1876	222.5	1909	181.4	1909	190.3	1907	245.4	1908	213.7
d1.m5.l0.33.rnd	190	190	0.1	190	0.1	190	0.1	190	0.1	190	0.0
d2.m5.l0.33.rnd	3261	3170	266.3	2887	259.4	2887	205.5	2886	194.1	2886	153.6
d3.m5.l0.33.rnd	7643	4582	2407.2	4427	2472.8	4427	2617.9	4410	2690.8	4416	2472.3
d4.m5.l0.33.rnd	9349	4751	4305.5	4658	4068.7	4658	4274.7	4674	4754.5	4656	4439.2
d5.m5.l0.33.rnd	10334	5166	7683.6	4994	7560.5	4994	6317.3	4993	6483.0	4992	5979.4
d6.m5.l0.33.rnd	11727	9588	1883.5	3927	1769.7	3927	1895.9	3929	2023.3	3927	1803.9
d7.m5.l0.33.rnd	8114	4529	3620.5	4313	3170.3	4313	3302.3	4305	3553.2	4303	3384.1
d8.m5.l0.33.rnd	4466	3477	1143.5	3515	781.0	3515	832.8	3516	1208.0	3518	823.3
d9.m5.l0.33.rnd	7052	4706	2194.9	4231	1738.1	4231	1846.7	4232	2347.8	4250	1903.1
d10.m5.l0.33.rnd	9471	6987	3716.2	3916	2659.1	3916	2811.3	3913	3850.5	3911	2820.2
d11.m5.l0.33.rnd	5741	4187	2238.3	3869	1457.0	3869	1512.8	3859	2371.7	3868	1583.7
d12.m5.l0.33.rnd	8891	8629	1069.7	2270	293.4	6838	837.5	8642	947.6	6823	361.8
d13.m5.l0.33.rnd	2856	2760	445.9	2514	95.0	2514	172.7	2766	418.7	2629	110.2
d14.m5.l0.33.rnd	4836	3874	858.1	3939	585.7	3939	616.5	3935	781.8	3938	570.5
d15.m5.l0.33.rnd	3783	3423	482.5	3338	298.8	3338	314.6	3340	531.9	3347	319.4
d1.m2.l0.67.rnd	195	195	0.1	190	0.0	190	0.1	195	0.1	195	0.1
d2.m2.l0.67.rnd	3262	1984	107.5	1728	107.9	1728	130.3	1724	136.7	1715	104.3
d3.m2.l0.67.rnd	7649	2303	1508.1	2171	1539.3	2171	1626.9	2170	1675.1	2169	1601.4
d4.m2.l0.67.rnd	9474	2373	2935.6	2233	2972.6	2233	3154.2	2223	3166.6	2234	3167.0
d5.m2.l0.67.rnd	11822	3980	5745.2	2455	5652.8	2455	4827.3	2454	4777.4	2452	4754.7
d6.m2.l0.67.rnd	14212	9934	1185.0	2015	1104.3	2015	1198.6	1999	1290.1	2000	1192.5
d7.m2.l0.67.rnd	9570	3622	2412.9	2157	2363.1	2157	2526.3	2158	2582.2	2157	2525.1
d8.m2.l0.67.rnd	4658	1993	463.1	1966	468.6	1966	507.3	1966	627.5	1966	515.8
d9.m2.l0.67.rnd	12600	7861	1339.8	2142	1199.3	2142	1278.4	2143	1412.0	2143	1349.1
d10.m2.l0.67.rnd	10706	5892	2172.5	1982	2178.2	1982	2335.7	1982	2547.5	1981	2351.4
d11.m2.l0.67.rnd	5966	2313	1320.6	2032	1119.8	2032	1182.2	2041	1446.3	2050	1272.2
d12.m2.l0.67.rnd	14869	13341	526.2	1638	334.3	1638	360.9	1639	587.8	1638	372.9
d13.m2.l0.67.rnd	3082	1767	199.4	1633	122.2	1633	129.4	1626	229.3	1634	146.9
d14.m2.l0.67.rnd	6206	3298	351.5	2116	325.0	2116	347.5	2112	385.5	2131	343.7
d15.m2.l0.67.rnd	4630	2676	222.1	1909	180.4	1909	192.9	1907	247.0	1908	214.0
d1.m5.l0.67.rnd	195	195	0.1	190	0.1	195	0.1	195	0.0	195	0.0
d2.m5.l0.67.rnd	3262	3171	269.7	2887	260.6	2887	205.5	2886	197.1	2887	153.6
d3.m5.l0.67.rnd	7649	4583	2427.4	4427	2490.6	4427	2606.7	4410	2694.0	4416	2519.7
d4.m5.l0.67.rnd	9474	4851	4273.5	4658	4104.8	4658	4239.6	4674	4720.0	4656	4546.8
d5.m5.l0.67.rnd	11822	6592	7061.1	4994	7096.4	4994	6189.6	4994	6285.0	4992	5856.0
d6.m5.l0.67.rnd	14212	12008	1894.9	3927	1782.5	3927	1887.7	3929	2030.3	3927	1810.1
d7.m5.l0.67.rnd	9570	5917	3554.8	4313	3110.6	4313	3247.4	4305	3496.2	4306	3383.0
d8.m5.l0.67.rnd	4658	3594	1145.9	3515	784.3	3515	838.6	3516	1202.4	3518	822.8
d9.m5.l0.67.rnd	12600	10108	2182.4	4231	1721.0	4231	1786.2	4232	2334.3	4250	1901.9
d10.m5.l0.67.rnd	10706	8028	3733.7	3916	2670.6	3916	2800.1	3913	3841.3	3912	2961.9
d11.m5.l0.67.rnd	5966	4254	2237.0	3869	1468.6	3869	1556.7	3860	2456.8	3868	1580.4
d12.m5.l0.67.rnd	14869	14411	1095.6	2270	289.2	10902	833.4	14415	941.8	8343	360.4
d13.m5.l0.67.rnd	3082	2884	443.2	2514	93.7	2514	174.8	2866	419.5	2648	110.2
d14.m5.l0.67.rnd	6206	5195	863.9	3939	586.4	3939	619.2	3935	791.6	3938	572.5
d15.m5.l0.67.rnd	4630	4224	479.4	3338	302.1	3338	313.0	3340	541.2	3347	319.1

Table 3.16: Placement of traditional locations is clustered

Instance	Total Accurate Demand	All		OnlyVerified		FirstVerified		PrioritizeVerified		CloseProximity	
		Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU	Max Demand Served	Avg CPU
d1.m2.l0.33.cl	187	187	0.1	182	0.1	187	0.1	187	0.2	187	0.2
d2.m2.l0.33.cl	3490	1985	98.1	1943	108.2	1943	134.6	1943	150.6	1943	124.9
d3.m2.l0.33.cl	7561	2226	1382.8	2195	1371.0	2195	1615.5	2195	1860.1	2195	1466.5
d4.m2.l0.33.cl	8807	2090	2705.0	2121	3101.5	2121	2960.7	2122	3382.6	2121	2808.8
d5.m2.l0.33.cl	10990	3330	4160.5	2255	3767.3	2255	5005.2	2256	5266.4	2256	4432.8
d6.m2.l0.33.cl	6595	1990	992.4	2248	1025.7	2248	1227.4	2256	1407.7	2248	1111.0
d7.m2.l0.33.cl	8857	2557	2209.0	2350	2638.0	2350	2147.0	2349	2736.8	2350	2343.0
d8.m2.l0.33.cl	4410	1893	445.8	1836	418.8	1836	455.3	1836	596.3	1836	449.0
d9.m2.l0.33.cl	7176	2898	1579.7	1967	1491.2	1967	1120.9	1967	1598.5	1967	1245.5
d10.m2.l0.33.cl	7162	2522	2048.4	2135	2240.4	2135	2215.2	2132	2518.1	2135	2215.7
d11.m2.l0.33.cl	5698	2410	1828.6	2130	1374.6	2130	1019.2	2130	1484.4	2130	1115.9
d12.m2.l0.33.cl	3380	2012	587.2	1717	372.1	1717	296.6	1712	686.9	1717	339.1
d13.m2.l0.33.cl	2759	1638	248.5	1724	162.0	1724	128.3	1720	196.5	1724	144.5
d14.m2.l0.33.cl	5726	2949	355.5	1976	334.9	1976	297.8	1991	377.6	1976	302.0
d15.m2.l0.33.cl	4535	2656	315.6	1768	189.6	1768	153.9	1769	196.6	1769	174.2
d1.m5.l0.33.cl	187	187	0.1	182	0.0	187	0.0	187	0.1	187	0.0
d2.m5.l0.33.cl	3490	3382	238.8	3294	195.9	3294	211.5	3300	213.8	3295	168.8
d3.m5.l0.33.cl	7561	4445	3229.3	4408	2726.2	4408	2136.0	4392	2687.9	4408	2450.2
d4.m5.l0.33.cl	8807	4440	5269.6	4435	4348.9	4435	3384.2	4434	3425.3	4435	3983.0
d5.m5.l0.33.cl	10990	5957	7978.4	4798	6298.9	4798	6929.5	4800	6434.2	4798	5758.7
d6.m5.l0.33.cl	6595	4186	2165.3	4355	1921.9	4355	1513.1	4368	2563.8	4355	1795.2
d7.m5.l0.33.cl	8857	5086	3721.6	4729	3307.3	4729	2538.5	4733	4105.5	4729	3232.0
d8.m5.l0.33.cl	4410	3440	1184.9	3447	811.9	3447	715.4	3449	1099.5	3449	724.6
d9.m5.l0.33.cl	7176	4938	2606.8	3865	1809.5	3865	1464.3	3858	2278.3	3865	1833.3
d10.m5.l0.33.cl	7162	4740	3979.7	4196	3306.7	4196	2597.8	4198	3294.6	4196	2753.8
d11.m5.l0.33.cl	5698	4336	2494.4	3940	1507.4	3940	1230.0	3933	2063.6	3940	1573.3
d12.m5.l0.33.cl	3380	3168	1115.4	2410	309.1	2456	641.6	3179	909.2	2482	346.4
d13.m5.l0.33.cl	2759	2681	489.0	2578	95.0	2578	146.5	2680	352.2	2602	105.4
d14.m5.l0.33.cl	5726	4831	781.0	3773	524.7	3773	507.5	3783	646.5	3773	528.1
d15.m5.l0.33.cl	4535	4128	628.2	3148	356.8	3148	312.4	3141	378.1	3149	379.0
d1.m2.l0.67.cl	190	190	0.1	182	0.1	190	0.1	190	0.1	190	0.1
d2.m2.l0.67.cl	3493	1985	97.5	1943	108.3	1943	137.7	1943	160.3	1943	127.6
d3.m2.l0.67.cl	7587	2226	1574.6	2195	1523.9	2195	1401.5	2196	1868.0	2196	1458.6
d4.m2.l0.67.cl	9213	2490	3526.7	2121	3156.4	2121	2514.0	2122	3009.4	2122	2818.0
d5.m2.l0.67.cl	11215	3441	5595.1	2255	4665.6	2255	5449.1	2256	5183.3	2256	4336.5
d6.m2.l0.67.cl	14230	9540	999.2	2248	1041.2	2248	1106.2	2256	1357.8	2248	1094.8
d7.m2.l0.67.cl	10086	3707	2532.0	2350	2603.4	2350	2044.0	2349	3029.0	2350	2333.5
d8.m2.l0.67.cl	4643	1968	448.9	1836	419.4	1836	456.7	1836	588.5	1836	456.1
d9.m2.l0.67.cl	7627	3160	1856.4	1967	1433.8	1967	1095.6	1967	1488.9	1967	1269.9
d10.m2.l0.67.cl	12091	7198	2217.9	2135	2528.7	2135	2220.2	2132	2421.3	2135	2220.1
d11.m2.l0.67.cl	6853	3375	1837.1	2130	1437.0	2130	1009.4	2130	1470.8	2131	1119.9
d12.m2.l0.67.cl	11596	9989	585.6	1717	374.9	1717	292.6	1713	656.5	1717	346.3
d13.m2.l0.67.cl	3381	2155	245.1	1724	163.2	1724	126.6	1720	199.1	1724	143.2
d14.m2.l0.67.cl	5874	3049	399.7	1976	327.2	1976	299.3	1991	372.5	1976	301.7
d15.m2.l0.67.cl	4901	2976	306.8	1768	189.7	1768	151.9	1769	196.3	1769	175.2
d1.m5.l0.67.cl	190	190	0.1	182	0.0	190	0.1	190	0.1	190	0.0
d2.m5.l0.67.cl	3493	3383	241.7	3294	183.0	3294	210.9	3302	237.4	3296	172.3
d3.m5.l0.67.cl	7587	4445	3221.4	4408	2705.2	4408	1985.6	4392	2521.8	4408	2454.7
d4.m5.l0.67.cl	9213	4840	5113.2	4435	4171.4	4435	3212.5	4434	3245.5	4435	3984.3
d5.m5.l0.67.cl	11215	6068	7185.2	4798	6038.9	4798	5386.2	4801	5182.7	4798	5824.0
d6.m5.l0.67.cl	14230	11742	2148.2	4355	1909.0	4355	1474.6	4368	2348.0	4355	1791.7
d7.m5.l0.67.cl	10086	6237	3520.6	4729	3322.4	4729	2519.2	4734	3309.3	4730	3251.4
d8.m5.l0.67.cl	4643	3583	1175.3	3447	775.2	3447	700.0	3449	1094.4	3450	720.2
d9.m5.l0.67.cl	7627	5200	2691.5	3865	1950.9	3865	1462.6	3859	2069.8	3866	1849.3
d10.m5.l0.67.cl	12091	9445	3970.7	4196	3056.9	4196	2585.2	4198	3259.8	4196	2734.2
d11.m5.l0.67.cl	6853	5322	2087.8	3940	1415.5	3940	1219.7	3933	2034.1	3941	1564.9
d12.m5.l0.67.cl	11596	11178	1055.2	2410	304.8	6965	640.5	11189	882.3	7530	348.7
d13.m5.l0.67.cl	3381	3211	527.2	2578	95.6	2578	146.3	3200	352.4	2721	106.0
d14.m5.l0.67.cl	5874	4933	759.8	3773	530.9	3773	504.8	3784	646.3	3773	530.3
d15.m5.l0.67.cl	4901	4448	630.1	3148	381.3	3148	321.2	3141	376.7	3149	378.6

3.8 Conclusions

The rapid delivery of life-saving supplies (e.g. water, food, and medical products) to affected populations is critical in minimizing human suffering and saving lives. However, providing the quick support in an efficient manner is quite challenging because information regarding the locations of impacted groups is needed. On-the-ground search and rescue teams identify this information by visiting disaster areas in person, but this effort is time-consuming, and disaster logistics activities (e.g. disaster relief tour planning) need to be done in the critical hours following the incident. For that reason, the emergency management community has begun to adapt by using information posted on social media platforms in order to identify a large amount of potentially life-saving information quickly. However, verifying the accuracy of social data is challenging, and decisions need to be made based on a limited information during the planning period. The trustworthiness and reliability of social data are significant concerns. Therefore, this study investigates whether there is value in considering and acting on social data prior to its verification in the context of planning disaster relief tours.

A case study motivated by the 2010 Haiti earthquake is used to test the developed models, where the real social data posts are available online whereas traditionally collected (e.g. on-the-ground assessment) data is not available. Thus, traditional data is generated using a damage assessment dataset of Haiti. In this case study, the objective of maximizing the amount of demand served is considered, because the main goal of the disaster response is to deliver emergency supplies to as many victims as possible, and the traditional efficiency objective of minimizing tour duration does not often adequately represent all considerations in disaster relief response planning. Five alternative social data decision strategies that an emergency manager could adapt in practice are presented: one that does not consider social ($OnlyVerified$), one that considers and acts on all social data (All), one that considers social data if all traditional requests have been served ($FirstVerified$), one that considers and acts on social data but higher priority is given to traditional data ($PrioritizeVerified$)

and one that allows acting on social data if a social data location is close to a traditional location on the route.

The problem under consideration is a vehicle routing problem variant called Team Orienteering Problem (TOP) with a single depot and a fleet of vehicles. To solve this problem, an Adaptive Large Neighborhood Search (ALNS) is employed. The developed model is validated using the TOP benchmark instances. The computational results indicate that ALNS finds 327 out of 387 best-known solutions with an average gap of 0.11%. These results confirm that ALNS is suitable for this case study application and can be adapted for alternative strategies. The computational results provide managerial insight associated with incorporating social data in disaster relief tour planning. An emergency manager wishing to select a strategy that will perform better for the majority of scenarios may prefer *All*. Besides *All*, the preference of $PrioritizeVerified$ is more pronounced than other alternative strategies. An emergency manager who is less optimistic regarding the accuracy of social data may prefer this approach. The performance of *All* tends to worsen as the placement of the traditional requests are changed from random to clustered, but still *All* has a superior average performance on the percentage of unsatisfied demand, while the performance of other strategies tends to improve. When request volume is high, the preference of *All* is even more distinct. The preference for $OnlyVerified$ is less pronounced when the proportion of social data increases over the proportion of traditional data.

The case study presented in this paper relies on several limitations. First, traditionally obtained verified data is not readily available and is generated based on damage assessment dataset in this study. Furthermore, we assume that traditional data follows the same demand distribution as Hurricane Katrina. However, the quality of insights will increase as real traditional data for the Haiti earthquake become available. Second, the performance of alternative emergency manager strategies depends on the proportion of social data locations that are accurate (λ), because the emergency manager will not know with certainty the proportion of social requests that are accurate. Therefore, varying this proportion can provide insight into the impact of considering social data in tour planning. However, a limited number of variations is considered in this study when $\lambda = 0.33$ and $\lambda = 0.67$. The computational study could be expanded by varying $\lambda \in [0, 0.1, 0.2, \ldots, 1.0]$. Third, we have considered only a fleet size of 2 and 5 vehicles with unlimited capacity in order to represent small and medium size relief organizations, respectively. However, the number of vehicles could be varied for quantitatively assessing the impact of including social data for various resource availability in disaster relief tour planning. Fourth, we assume that the available time is limited to 12 hours per day to serve all request locations. Instead of a general time budget, need-dependent time windows at each request location can be considered. Finally, we have only considered a social data demand location that can either be accurate or inaccurate. However, in a real-world case, a specified location may be correct, but the quantity of goods may be specified incorrectly. Thus, the demand is present but at a different magnitude than expected. This request can be classified as partially-accurate. Despite these limitations, this case study addresses questions around the usefulness of information provided through citizen event reporting.

Bibliography

[1] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," European journal of operational research, vol. 88, no. 3, pp. 464–474, 1996.

[2] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," European Journal of Operational Research, vol. 209, no. 1, pp. 1–10, 2011.

[3] S. E. Butt and T. M. Cavalier, "A heuristic for the multiple tour maximum collection problem," Computers & Operations Research, vol. 21, no. 1, pp. 101–111, 1994.

[4] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," Computers & Operations Research, vol. 32, no. 6, pp. 1379–1407, 2005.

[5] C. Archetti, A. Hertz, and M. G. Speranza, "Metaheuristics for the team orienteering problem," Journal of Heuristics, vol. 13, no. 1, pp. 49–76, 2007.

[6] L. Ke, C. Archetti, and Z. Feng, "Ants can solve the team orienteering problem," Computers & Industrial Engineering, vol. 54, no. 3, pp. 648–665, 2008.

[7] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. Van Oudheusden, "A guided local search metaheuristic for the team orienteering problem," European Journal of Operational Research, vol. 196, no. 1, pp. 118–127, 2009.

[8] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "An effective pso-inspired algorithm for the team orienteering problem," European Journal of Operational Research, vol. 229, no. 2, pp. 332–344, 2013.

[9] B.-I. Kim, H. Li, and A. L. Johnson, "An augmented large neighborhood search method for solving the team orienteering problem," Expert Systems with Applications, vol. 40, no. 8, pp. 3065–3072, 2013.

[10] S. Boussier, D. Feillet, and M. Gendreau, "An exact algorithm for the team orienteering problem," 4OR, vol. 5, pp. 211–230, 2007.

[11] Y. S. Su, C. W. III, and Z. Thorkildsen, "Social media in the emergency management field," tech. rep., Report, CNA, 2010. Available at `http://www.cna.org/sites/default/files/research/SocialMedia_EmergencyManagement.pdf`.

[12] R. S and P. D., "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows.," Transportation Science, vol. 40, pp. 455–472, 2006.

[13] G. M. Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," Computers & Operations Research, vol. 39, pp. 728–735, 2012.

[14] V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," Computers & Operations Research, vol. 39, no. 12, pp. 3215–3228, 2012.

[15] N. Azi, M. Gendreau, and J.-Y. Potvin, "An adaptive large neighborhood search for a vehicle routing problem with multiple routes," Computers & Operations Research, vol. 41, pp. 167–173, 2014.

[16] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," European Journal of Operational Research, vol. 223, no. 2, pp. 346–359, 2012.

[17] Y. Adulyasak, J.-F. Cordeau, and R. Jans, "Optimization-based adaptive large neighborhood search for the production routing problem," Transportation Science, vol. 48, no. 1, pp. 20–45, 2012.

[18] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," Principles and Practice of Constraint Programming. Springer Berlin Heidelberg, 1998.

[19] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," Computers & operations research, vol. 34, no. 8, pp. 2403–2435, 2007.

[20] R. S and P. D., "A unified heuristic for a large class of vehicle routing problems with backhauls," European Journal of Operational Research, vol. 171, pp. 750–775, 2006.

[21] H. Bouly, D.-C. Dang, and A. Moukrim, "A memetic algorithm for the team orienteering problem," 4OR, vol. 8, no. 1, pp. 49–70, 2010.

[22] W. Souffriau, P. Vansteenwegen, G. V. Berghe, and D. Van Oudheusden, "A path relinking approach for the team orienteering problem," Computers & Operations Research, vol. 37, no. 11, pp. 1853–1859, 2010.

[23] S. Muthuswamy and S. S. Lam, "Discrete particle swarm optimization for the team orienteering problem," Memetic Computing, vol. 3, no. 4, pp. 287–303, 2011.

[24] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "A pso-based memetic algorithm for the team orienteering problem," in Applications of Evolutionary Computation, pp. 471–480, Springer, 2011.

[25] Ushahidi, "Haiti dataset." Available at `http://community.ushahidi.com/research/datasets/`. Accessed June 2012.

[26] A. R. Kolbe and R. A. Hutson, "Human rights abuse and other criminal violations in port-au-prince, haiti: a random survey of households," The Lancet, vol. 368, no. 9538, pp. 864–873, 2006.

[27] R. sensing damage assessment:, "Haiti earthquake 2010: Remote sensing based building damage assessment data," tech. rep., UNITAR/UNOSAT, EC JRC and World Bank, 2010. Available at `http://www.unitar.org/unosat/haiti-earthquake-2010-remote-sensing-based-building-damage-assessment-data`.

[28] J. Holguín-Veras and M. Jaller, "Immediate resource requirements after hurricane katrina," Natural Hazards Review, vol. 13, no. 2, pp. 117–131, 2011.

[29] A. Martinez and L. Wassenhove, "The haiti relief operation: Bottlenecks to aid delivery during the first week of response," tech. rep., Social Innovation Centre. Accessed December 2015. Available at `http://centres.insead.edu/humanitarian-research-group/research-projects/documents/HaitibottlenecksFINAL.pdf`.

[30] ESRI, "Arcgis." Available at `http://www.arcgis.com/features/index.html`. Accessed June 2016.

[31] Geofabrik, "Haiti and dominican republic." Available at `http://download.geofabrik.de/central-america/haiti-and-domrep.html`. Accessed June 2016.

[32] ESRI, "Arcgis network analyst." Available at `http://www.esri.com/software/arcgis/extensions/networkanalyst`. Accessed June 2016.

4. THE DYNAMIC TEAM ORIENTEERING PROBLEM

4.1 Introduction

During the last two decades, the orienteering problem (OP) has received considerable attention in the literature. Many real-world applications such as humanitarian relief logistics and tourist and school bus routing problems have been modeled as OP variants [1, 2, 3]. In the OP, there is a set of locations and each location is associated with a profit (score). The problem is to develop a single tour, subject to duration limit T_{max}, that visits each location at most once. The objective is to maximize the collected profit. The extension of the OP to multiple routes is introduced by Chao et al. [4] under the name of the team orienteering problem (TOP). In the TOP, the profit of a location can be collected by at most one vehicle and the objective is to maximize the total profit collected by all vehicles. For a comprehensive review of the OP and its variants, the reader is referred to Vansteenwegen et al. [5].

In this paper we introduce a dynamic variant of the TOP called the dynamic team orienteering problem (DTOP). The DTOP differs from the TOP in that instead of being known *a priori*, profit locations are revealed over time. The goal is to determine a set of vehicle routes, each constrained by maximum duration T_{max}, that maximizes the total profit collected over a planning horizon. To the best of our knowledge, despite numerous variants of static OP and TOP and their real-world applications, no other studies in the literature have considered this problem. The most closely related papers are as follows. Lau at el. (2012) consider a dynamic and stochastic OP (single-vehicle) in which demand is deterministic but travel times are random variables with distributions that depend on arrival times at customer locations [6]. They develop a local search algorithm that combines a greedy insertion heuristic with a hybrid variable neighborhood search and simulated annealing approach for its solution. Zhang et al. (2015) study a stochastic OP, where demand is deterministic and customer wait times are stochastic [7]. The problem is proposed in the context of routing and scheduling a textbook salesperson, where the salesperson may need to wait in a queue of unknown length to meet the customer. They model the problem as a Markov decision process and use an approximate dynamic programming approach to solve it. Li (2012) describes a TOP variant where travel times are dynamic and vary with time, while all other problem elements such as scores at nodes are deterministic [8]. A mixed integer programming model is presented, and a dynamic node labelling algorithm is designed based on the idea of network planning and dynamic programming. In summary, only a few dynamic OP studies are available in the literature at the time of this writing, and none of these consider demand locations which are revealed dynamically over time.

The problem we introduce, DTOP, is in the general category of dynamic deterministic routing problems, where part or all of the input is unknown at the beginning of the planning horizon and stochastic information describing the uncertainty is not available [9]. Routes must be redefined in an ongoing fashion as new demand information becomes available. Many dynamic vehicle routing problem (VRP) variants appear in the literature; a comprehensive review is provided in Pillac et al. [9]. The most common objective among dynamic VRPs in the literature is the minimization of travel time [10, 11, 12, 13, 14]. To the best of our knowledge, the objective of maximizing total profit (or maximizing total demand served) over a planning horizon has not been considered. A related objective, maximizing the number of customer locations visited over a planning horizon, has been studied by Bent and Van Hentenryck [15], but the problem makes no distinction between high and low demand customers. Thus, the dynamic variant of the problem they study can be viewed as a special case of the DTOP we introduce, where the profit is the same at each customer location.

To solve the DTOP, we adapt the multiple plan approach (MPA) presented by Bent and Van Hentenryck (2004) for the dynamic vehicle routing problem with time windows (VRPTW) [15]. Using MPA, a pool of alternative routing plans is maintained throughout a planning horizon, where routing plans may differ in the subsets of customers they serve and also in the orders in which they serve those customers. Routing plans are updated each time a new customer request is revealed. To determine the actions to execute over time (i.e., which customer to visit next), a distinguished plan is selected from among the alternative plans in the pool each time a vehicle is ready to leave its current location. In Bent and Van Hentenryck (2004), the distinguished plan is selected using a consensus function that determines which actions are taken the most frequently across all plans in the pool. In our adaptation of MPA for DTOP, we employ the consensus function idea and also test a new method for selecting a distinguished plan. Specifically, we pick the plan that maximizes total demand served throughout the planning horizon. Another primary difference between the MPA algorithm presented by Bent and Van Hentenryck and ours lies in the local search method used to improve each routing plan in the pool. They use a two-stage hybrid local search for VRPTW whereas we use an adaptive large neighborhood search for TOP.

We assess the performance of the MPA algorithm for DTOP using two methods. First, we compare the multiple-plan approach to a single-plan approach in which the single plan is developed using a sophisticated greedy algorithm as proposed in [15] and [14]. Both the MPA and single-plan approaches are online algorithms. Second, we compare the performance of the online MPA algorithm to the solution of the offline problem in which all input data is known a priori. To obtain the solution to the offline problem, an exact solution algorithm is used. Specifically, the constraint programming (CP) model developed by Gedik et al. [16] for solving the static Team Orienteering Problem with Time Windows (TOPTW) is used. To use this model, we treat the customer disclosure time as the beginning of the time window for a customer and the tour duration limit T_{max} as the end of the time window. Then percentage deviation analysis is utilized to compare the online and offline solutions.

The contributions of this work are three-fold. First, we introduce and define a new problem: the dynamic team orienteering problem (DTOP). This study extends the modeling of TOP by considering dynamic customer arrivals. Second, a multiple plan approach (MPA) for DTOP is presented, and a new local search and a method for selecting a distinguished plan (maximizing total demand served) are evaluated. Lastly, 1161 new benchmark instances for DTOP are introduced. They are derived by modifying the TOP benchmark instances

[17]. We provide optimal offline solutions for 925 of the instances using our CP approach. This addresses the need for benchmark instances of dynamic routing problems emphasized in Pillac et al. [9].

This paper is organized as follows. In Section 3.2, a formal problem statement is presented. Section 3.3 provides a review of existing literature regarding solution algorithms and evaluation methods for related problems. Section 3.4 presents the solution method in detail. Section 3.5 presents and discusses the results of the computational experiments and future research. The last section gives concluding remarks.

4.2 Problem statement

The DTOP studied in this chapter is defined on a complete graph $G = (\mathcal{N}, \mathcal{A})$ with a set of $n+2$ nodes, where $\mathcal{N} = \{0, \ldots, n, n+1\}$ is the set of nodes, and $\mathcal{A} = \{(i,j) : i \neq j \in \mathcal{N}\}$ is the set of arcs connecting nodes in $\mathcal{N}$. The node set $\mathcal{N}$ includes a starting node ($i = 0$) and an ending node ($i = n+1$) while other nodes are labeled 1 to n, each with a profit (demand) $p_i > 0$, where $p_0 = p_{n+1} = 0$. The starting node may be in the same location as the ending node, but this is not a requirement. There is a set of nodes $\mathcal{K} \subseteq \mathcal{N}$ known in advance and a set of nodes $\mathcal{D} \subseteq \mathcal{N}$ revealed throughout the planning horizon. We denote the disclosure time of a node $i \in \mathcal{D}$ as d_i, where $d_i \in (0, T_{max}]$. Note that $\mathcal{N} = \mathcal{K} \cup \mathcal{D} \cup \{0, n+1\}$. The travel time on arc (i,j), denoted c_{ij}, is known. A fixed number of vehicles (m) is available, and each vehicle $k \in \mathcal{M}$ must depart from node 0 at time 0 and arrive at the ending node $n+1$ no later than T_{max}. It is possible for a vehicle to wait for a time w_i at its current location i before departing for the next node. Because of the limited number of vehicles and the duration constraint, visiting all nodes may not be feasible. Therefore, the objective is to find a set of m vehicle routes that maximizes the total profit ($\max \sum_{k=1}^{m} \sum_{(i \in r^k)} p_i$) collected by visiting nodes throughout a planning horizon.

4.3 Related literature

This section provides a brief review of existing research regarding solution strategies to tackle dynamic and deterministic routing problems and summarizes evaluation methods used for assessing alternative solution methods for dynamic routing problems.

4.3.1 Solution methods

A broad range of algorithms has been developed to accommodate the dynamic nature of routing problems. Pillac et al. [9] classify dynamic routing problems in two categories: deterministic and stochastic. In both categories, the inputs to the problem become available to the decision maker over time or change over time. In stochastic problems, the stochastic information on the dynamically revealed inputs is described using known probability distributions. For deterministic problems, such probability distributions are not available; optimization is performed only on the known inputs. Because the stochastic information is not available in the problem presented in this paper, DTOP is in the dynamic and deterministic category.

Solution methodologies for dynamic and deterministic routing problems can be divided into two categories: single-plan optimization and multiple-plan optimization. Single-plan optimization methods solve a static routing problem each time a new request is received and maintain a single solution (i.e, one set of vehicle routes in the case of DTOP). Traditional algorithms such as tabu search [18, 19], ant colony algorithm [20, 21], large neighborhood search[22], genetic algorithm [23], and integer programming [24, 25] can be used to develop the single plan. Multiple-plan optimization methods are based on an adaptive memory that maintains and stores alternative solutions (i.e., multiple sets of vehicle routes in the case of DTOP) throughout a planning horizon. Each plan in the set of alternative solutions is generated by solving a static problem for the known customer locations which have been revealed by a particular point in time,. The adaptive memory is used to make a decision at discrete points in time, such as when a new customer is disclosed or when a vehicle is ready to depart its current location (to decide what customer to visit next). Gendreau et al. [10] introduce a parallel tabu search algorithm with adaptive memory for the dynamic VRPTW. In their study, a set of initial solutions is constructed and stored in the adaptive memory. New solutions are created by combining routes from different solutions from the memory. This combination process is similar to the crossover operation performed in genetic algorithms in which a new solution is obtained by a decomposition/reconstruction method and stored in the memory. The tabu search is applied to each subset of routes produced by the decomposition method. The adaptive memory replaces the worst solution if the memory is full. Bent and Van Hentenryck [15] generalize this algorithm by making solutions in adaptive memory independent of a specific local search and by introducing the Multiple Plan Approach (MPA) to solve the Dynamic VRPTW. Pillac et al. [14] propose two multi-plan optimization methods: a fast re-optimization approach based on a parallel Adaptive Large Neighborhood Search (pALNS) and a MPA. Both algorithms maintain a pool of solutions. The difference is that MPA continuously optimizes the pool of solutions while pALNS only generates a new solution whenever a new customer appears. In this study, the MPA presented by Bent and Van Hentenryck [15] is adapted to tackle the DTOP, because this algorithm is flexible enough to account for specific aspects of DTOP.That is, any static local search algorithm can be selected to implement within an event-driven framework to optimize each plan in the pool.

4.3.2 Assessment of solution methods

Unlike static routing problems in which measuring the performance of a method is relatively easy using reference benchmark instances, dynamic routing problems may require additional efforts to assess the performance of a proposed algorithm. Bent and Van Hentenryck [15] present a multiple scenario approach (MSA) that continuously generates routing plans for scenarios using known and future requests in order to solve a stochastic VRP with time windows. In the problem, stochastic information on the dynamically revealed locations based on known probability distributions is available. To assess the performance of the MSA, they introduce an additional two algorithms; one that does not consider stochastic information called MPA and one that is a sophisticated greedy algorithm. The MSA approach is developed for solving a dynamic and stochastic problem, while the MPA and the greedy algorithm are proposed for solving a dynamic (not stochastic) problem. In their study, comparing the MSA to the MPA evaluates the value of stochastic information, while comparing the MPA to the greedy approach assesses the value of maintaining multiple plans for a number of future scenarios. Hvattum et al. [12] propose a dynamic stochastic hedging heuristic (DSHH) to solve the dynamic and stochastic vehicle routing problem in which the goal is to minimize the number of vehicles used and the total time travelled. In the DSHH, sample scenarios are used to build a plan for each time interval. The algorithm generates scenarios based on deterministic and stochastic future customers. They describe two alternative solution methods that are used for comparison purposes. One is a myopic dynamic heuristic in which stochastic information is ignored, and the other is based on a deterministic model where a solution is developed in hindsight using a local search heuristic. By comparing the myopic dynamic heuristic to the offline approach, the authors assess the quality of the myopic algorithm. Pillac et al. [14] study a dynamic deterministic version of the technician routing and scheduling problem in which the objective is to minimize the total working time and the total distance. They propose a parallel adaptive large neighborhood search (pALNS) and MPA to solve this problem. These two proposed algorithms are compared with a greedy heuristic in which a new request is accepted or rejected based on the availability of a feasible insertion place.

In our study, a greedy algorithm which constructs an initial solution using local search provides an online comparison for MPA. Also, a reference offline algorithm is used to assess the performance of MPA. Specifically, a CP model developed for the static TOPTW by Gedik et al. [16] isused. The following section presents the modified MPA framework in detail.

4.4 Problem-solving methodology

In this section, we first introduce a greedy algorithm used for the single-plan optimization and then present an adapted MPA for the DTOP. The parameters associated with the algorithms given in the paper are defined in Table 4.1. Let $r_t^k = \{0, i_1, i_2, \ldots, i_{j-1}, i_j, \ldots, n+1\}$ be a planned route k at time t, where i_g denotes the g^{th} node visited on the route. At any particular time t, the vehicle k executing the route may be at a node or traveling between nodes. Define $pred(k,t)$ as the node most recently visited by vehicle k at time t (the vehicle may still be located there) and define $succ(k,t)$ as the node to be visited next. Therefore, the portion of the route that has been executed at time t, denoted R_t^k, is $\{0, i_1, i_2, \ldots, pred(k,t)\}$. Note that $r_t^k = R_t^k + \{succ(k,t), \ldots, n+1\}$. Because vehicles are allowed to wait at a node i for a time w_i before departing for the next, the arrival time to the g^{th} node in route k is calculated as $a_g = a_{g-1} + w_{g-1} + c_{g-1,j}$ for all $g \in r_t^k$, where $a_0 = 0$ and $a_{n+1} \leq T_{max}$ for all $k \in \mathcal{V}$.

4.4.1 A greedy algorithm for DTOP

The greedy algorithm maintains a single routing plan which is updated throughout a planning horizon. The plan q at time t is denoted σ_t^q, where σ_t^q is the set of all vehicle routes $\{r_t^1, r_t^2, \ldots, r_t^m\}$. In general, greedy algorithms in the vehicle routing literature produce a feasible solution without using a sophisticated local search scheme. However, the greedy algorithm we develop does make use of an iterative local search algorithm as in [15] in order to produce a good initial routing plan σ_0 for the customers known *a priori*. Algorithm 2 outlines the greedy algorithm, beginning with how the initial solution σ_0 is constructed. First, a customer i is randomly selected from the set of known customers $\mathcal{K}_0$ and placed into the first position in a randomly selected vehicle route. That is, because it is the first customer being considered, it will be visited immediately after the depot. Customers selected for

Table 4.1: Parameters used in the algorithms

Parameter	Description
r_t^k	The route planned for vehicle k that exists at time t, $r_t^k = \{0, i_1, i_2, \ldots, n+1\}$
$pred(k,t)$	Most recent customer location visited by vehicle k at time t
$succ(k,t)$	Next customer planned for vehicle k to visit after time t
R_t^k	The portion of r_t^k that has been executed by time t, $R_t^k = \{0, i_1, i_2, \ldots, pred(k,t)\}$
σ_t^q	A routing plan q (set of routes for all vehicles) at time t, $\sigma_t^q = \{r_t^1, r_t^2, \ldots, r_t^m\}$
σ^*	The distinguished plan
Λ_t	Currently executed plan at time t, $\Lambda_t = \{R_t^1, R_t^2, \ldots, R_t^m\}$
$\mathcal{P}_t$	The set of routing plans in the pool at time t, $\mathcal{P}_t = \{\sigma_t^1, \sigma_t^2, \ldots, \sigma_t^z\}$
$\mathcal{K}_t$	The set of known customers at time t, $\mathcal{K}_t \subseteq \mathcal{N}$
$\mathcal{V}_t$	The set of customers that have been visited by time t (those appearing in Λ_t)
$\mathcal{C}_t$	The set of candidate customers at time t, $\mathcal{C}_t = \mathcal{K}_t \setminus \mathcal{V}_t$
$\mathcal{G}(\sigma_t^q)$	The set of customers included in plan q at time t regardless of whether they have been visited yet
i_t^*	New dynamic customer at time t
L_t	A list of possible customers to visit next at time t, $L_t = \{succ(1,t), succ(2,t), \ldots, succ(m,t)\}$
$\gamma(i)$	The evaluation function of the request i
$f(\sigma_t)$	The ranking function for each plan σ_t

insertion in later iterations will be placed at the end of the route, to be visited immediately after the last customer currently in the route. In this way, routes are constructed in "start to finish? fashion. Customers are only inserted into these positions if it is feasible; that is, if the vehicle can accommodate this new customer and still return to the depot on time. Customer insertions continue until all known customers in $\mathcal{K}_0$ have been attempted (lines 2 and 3). Note that routing plan σ_0^q only contains known customers at this point, and does not necessarily contain all of them.

Following this construction phase, the initial solution σ_0^q is improved using an iterative local search that aims to improve the plan by maximizing the total profit collected by routes in the plan (line 4). The type of local search used is the Adaptive Large Neighborhood Search (ALNS) introduced in Chapter 3. Still, only customers in $\mathcal{K}_0$ can be included in the routing plan. During ALNS, the initial solution σ_0^q may be partially or fully destroyed and then repaired through the application of several move operators. Once ALNS terminates, the initial solution will undergo no further changes until new dynamic customers are disclosed over time.

As new dynamic customers are disclosed, a greedy criterion is used to determine whether they can be accommodated in σ_t^q, and if so, where they should be inserted. Here, we assume an insertion cost function $C_i(a,b)$ to represent the cost of inserting a node i between nodes a and b. This requires removing edge (a,b) from the current tour and adding edges (a,i) and (i,b). Thus, the insertion cost is $C_i(a,b) = c_{ai} + c_{ib} - c_{ab}$. For a dynamic customer disclosed at time t, the greedy algorithm searches for the cheapest feasible insertion location (a^*, b^*) in σ_t^q, where $(a^*, b^*) = argmin_{(a,b) \in \sigma_t^q} C_i(a,b)$ (line 8). The only insertion locations that may be considered are those falling within the un-executed part of a route r_t^k, that is, $R_t^k + \{succ(k,t), \ldots, n+1\}$. If no feasible insertion locations are available for the customer, it is rejected and the routing plan remains unchanged. If multiple customers have the same disclosure time, the algorithm considers them sequentially based on their customer indices in increasing order. During route execution, routing plan σ_t^q specifies the next location $succ(k,t)$ a vehicle k should visit once it is ready to depart its current location (line 11). If no such customer exists, then the vehicle waits at the current location until a new customer appears or until it must return to the depot in order to not violate T_{max}; whichever comes first.

Algorithm 2 A general algorithm for the greedy algorithm

1: Initialize: $t = 0$
2: $\mathcal{K}_t \leftarrow$ select known customers $(\mathcal{N})$
3: $\sigma_t^q \leftarrow GenerateRandomSolution(\mathcal{K}_t)$
4: $\sigma_t^q \leftarrow LocalSearch(\sigma_t^q)$
5: **repeat**
6:     **if** (event = new customer i_t^* arrival) **then**
7:         $\mathcal{K}_t \leftarrow i_t^*$
8:         $\sigma_t^q \leftarrow CheapestInsertion(i_t^*)$
9:     **end if**
10:     **if** (event = vehicle k departure) **then**
11:         $R_t^k \leftarrow succ(k,t)$
12:     **end if**
13: **until** $(t = T_{max})$

4.4.2 A multiple plan approach for DTOP

The main idea of the MPA is to generate and maintain a solution pool that is used to determine a distinguished plan at every decision process. Decisions are required each time an event arises (lines 3, 7, and 24). Three different types of events take place during the response planning: (i) customer arrivals occur, whenever a new request is received, (ii) vehicle departures occur, whenever a vehicle completes serving a customer, (iii) timeouts occur, whenever the end of a planning period is reached and all vehicles arrive at the ending depot. The primary mechanisms of the MPA include (i) generating a pool of plans, (ii) maintaining the routing plans in the pool, and (iii) selecting a distinguished plan to determine actions to execute. Maintaining multiple plans tend to lead to a better solution than a single plan, because a set of alternative plans provides more diverse solutions to choose a best plan among them according to a selection criterion, instead of a single plan option. Adaptations to the mechanisms of the multiple-plan approach for the problem under consideration are described within the following sections.

4.4.2.1 Generating a pool of plans

The aim of the plan pool is to populate and maintain a set of diverse routing plans that are used to choose a distinguished plan at every decision process. Each plan σ_t^q in the pool $\mathcal{P}_t$ must have the same executed partial routes R_t^k for all vehicles k but the un-executed portions of the routes may vary from plan to plan. In order to have diverse routing solutions it is necessary to have a randomized structure to generate alternative solutions. The approach used to obtain a set of plans $\mathcal{P}_t$ at every execution step is presented in Algorithm 3.

The algorithm starts by populating a pool of solutions, $\mathcal{P}_0$ at time 0 based on currently known information $\mathcal{K}_0$. A candidate list $\mathcal{C}_t$ holds the current set of unvisited customer locations that can be inserted into the set of routes (lines 2 and 3).Anytime a new customer location is disclosed, it is added to $\mathcal{C}_t$, and anytime a location is visited, it is removed. Each vehicle plan σ_t^q in the pool $\mathcal{P}_t$ is generated via a random constructive heuristic. When the al-

Algorithm 3 Solution pool generation

1: Initialize: $t = 0$, $\mathcal{P}_t = \emptyset$, $\mathcal{C}_t = \emptyset$, $r_t^k = \emptyset$, and $\sigma_t^q = \emptyset \ \forall k, \forall q$
2: $\mathcal{K}_t \leftarrow$ select known customers $(\mathcal{N})$
3: $\mathcal{C}_t \leftarrow i \in \mathcal{K}_t \setminus \mathcal{V}_t$ assign unvisited requests to candidate list
4: **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
5:     **for all** $r_t^k \in \sigma_t^q$ **do**
6:         **repeat**
7:             $r_t^k \leftarrow$ randomly assign request $i \in \mathcal{C}_t$
8:             $G(\sigma_t^q) \leftarrow i$
9:         **until** no more feasible assignments
10:     **end for**
11: **end for**
12: **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
13:     $\sigma_t^q \leftarrow LocalSearch(\sigma_t^q)$
14: **end for**
15: **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
16:     **if** $([\mathcal{K}_t \setminus G(\sigma_t^q)] \neq \emptyset)$ **then**
17:         $r_t^k \leftarrow CheapestInsertion(i \in [\mathcal{K}_t \setminus G(\sigma_t^q)])$
18:     **end if**
19: **end for**

gorithm starts planning at time t, σ_t^q is simply the same plan from the previous period, except perhaps now more customers have moved into the visited set of routes $\Lambda_t = \{R_t^1, R_t^2, \ldots, R_t^m\}$. A customer location i is randomly selected from the candidate list $\mathcal{C}_t$ and inserted into a position on a vehicle route r_t^k that brings the smallest increase in tour duration if it is feasible (line 7). Then a local search procedure, ALNS, introduced in Chapter 3, is used to improve the total collected profit of plans in the pool (line 13). After ALNS, unassigned known customers (regardless of they have been visited) in the plans σ_t^q at time t are then inserted in decreasing order of demand into the improved routing plans in the cheapest feasible insertion location (a^*, b^*) (line 17).

4.4.2.2 Maintaining a routing plan

The algorithm continuously solves static routing problems and adds the new routing plans into the pool. The routing plans in the pool are tentative solutions which are subject to change whenever a new event occurs. Algorithm 4 describes how routing plans in the solution pool can be maintained by the implemented MPA. The event-time-increment simulation technique is used to check the state of the system at each event. In this system, the time is advanced to those instants in time when critical events occur by simply skipping all eventless points in time. When a new customer i_t^* is received (line 3), it is added to the known customer list $\mathcal{K}_t$ at time t, and the algorithm decides whether it can be inserted into already developed plans σ_t^q in the pool or whether it should be rejected (line 4). If the customer is accepted, it is inserted a location that falls within the un-executed part $\{succ(k,t), \dots, n+1\}$ of a route r_t^k using the cheapest insertion method (line 5).

When a vehicle reaches a destination (line 7), it has to be determined whether the vehicle is assigned to the next customer location or whether the vehicle waits until the next event in its current location. If there are not any customer locations to visit next ($\mathcal{C}_t = \emptyset$), the vehicle waits at its current location until the next event occurs. Otherwise, all routing plans in the pool are improved using a local search (line 9), and then a new distinguished plan σ^* is selected to determine the next destination $succ(k,t)$ for the vehicle route k (line 11). Therefore, all routing plans in the pool must include the currently visited customer locations of the performed plan $\Lambda_t = \{R_t^1, R_t^2, \dots, R_t^m\}$.

The distinguished plan σ^* evolves over time as new customers are revealed, and vehicle routes r_t^k in the plans σ_t^q are modified to accommodate them. When the executed portions $[0, \dots, pred(k,t)]$ of the performed route R_t^k and the alternative routes r_t^k are not identical, r_t^k are removed from the plan $\mathcal{P}$ (line 16). The algorithm maintains routing plans by regenerating the deleted route (line 17) that needs to include the currently executed route R_t^k. The unexecuted portion $\{succ(k,t), \dots, n+1\}$ of this route r_t^k, which includes known unvisited customer locations, is populated and improved by the local search (lines 18 and 19). Note that if a vehicle k is already traveling towards the next request location $succ(k,t)$, the vehicle cannot be diverted from this destination. The algorithm stops at the end of the fixed planning horizon and all vehicles arrive at the depot before T_{max} (line 24).

Algorithm 4 A general algorithm for the implemented event-driven MPA

Initialize: $R_t^k = \emptyset \ \forall \ k$, $t = 0$, and $\mathcal{C}_t = \emptyset$
repeat
    if (event = new customer i_t^* arrival) **then**
        $\mathcal{K}_t \leftarrow i_t^*$
        for all $\sigma_t^q \in \mathcal{P}_t$ **do**
            $r_t^k \leftarrow CheapestInsertion(i^*)$
        end for
    end if
    if (event = vehicle k departure) **then**
        for all $\sigma_t^q \in \mathcal{P}_t$ **do**
            $\mathcal{P}_t \leftarrow LocalSearch(\sigma_t^q)$
        end for
        $succ(k,t) \leftarrow distinguishedPlanSelection(\mathcal{P}_t)$
        $R_t^k \leftarrow i_{next}$ and $\mathcal{V}_t \leftarrow succ(k,t)$
        for all $\sigma_t^q \in \mathcal{P}_t$ **do**
            for all $r_t^k \in \sigma_t^q$ **do**
                if (r_t^k is incompatible with R_t^k) **then**
                    $\sigma_t^q \leftarrow \sigma_t^q \setminus \{r_t^k\}$
                    $r_t^k \leftarrow GenerateNewRoute(\mathcal{C}_t, R_t^k)$
                    $\sigma_t^q \leftarrow \sigma_t^q + \{r_t^k\}$
                    $\sigma_t^q \leftarrow LocalSearch(\sigma_t^q)$
                end if
            end for
        end for
    end if
until $(t = T_{max})$

4.4.2.3 Selecting a distinguished plan

The aim is to maintain a pool of plans to select a solution to determine actions to execute for each decision epoch. Therefore, a distinguished plan σ^* is chosen among the alternative solutions in the pool at the current time t whenever a vehicle completes serving a customer and waits for its next assignment. The next destination $succ(k,t)$ of the vehicle route k is assigned according to the selected distinguished plan σ^*. Since the routing plans in the pool are updated over time as more customers are received, a new distinguished plan at each decision epoch needs to be selected continuously to determine actions to execute. In this study, the distinguished plan σ^* is selected from the current pool $\mathcal{P}_t$ based on either (i) a consensus function as described in [15] or (ii) total demand served.

Algorithm 5 The consensus algorithm for the MPA

1: Initialize: $\gamma \leftarrow 0 \; \forall i \in G(\sigma_t^q)$, $\sigma^* = \emptyset$, $f(\sigma_t^q) \leftarrow 0$, and $f^*(\sigma_t^q) \leftarrow 0$
2: **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
3:     **for all** $k \in \mathcal{M}$ **do**
4:         $L_t \leftarrow NextCustomersToVisit(\sigma_t^q, k)$
5:     **end for**
6: **end for**
7: **for all** $i \in L_t$ **do**
8:     **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
9:         **if** $i \in G(\sigma_t^q)$ **then**
10:             $\gamma(i_0) \leftarrow \gamma(i_0) + 1$
11:         **end if**
12:     **end for**
13: **end for**
14: **for all** $\sigma_t^q \in \mathcal{P}_t$ **do**
15:     $f(\sigma_t^q) \leftarrow 0$
16:     **for all** $v \in \sigma_t^q$ **do**
17:         $f(\sigma_t^q) \leftarrow f(\sigma_t^q) + \gamma(i)$
18:     **end for**
19: **end for**
20: $f^*(\sigma_t^q) \leftarrow argmax_q(f(\sigma_t^q))$
21: $\sigma^* \leftarrow \sigma_t^q$
22: **return** σ^*

The consensus function chooses the next customer location $succ(k, t)$ based on which appears with highest frequency across all plans and assigns the customer to the associated vehicle route k. This selection method chooses a distinguished plan σ^* by ranking the plans in the pool based on their similarities. Algorithm 5 presents an outline of the consensus algorithm implemented for the MPA. The algorithm begins by initializing the ranking function $f(\sigma_t^q)$ for each plan. Then the algorithm selects the next customers $succ(k, t)$ to visit for the vehicle(s) and adds them into a list L_t of candidate customers to visit next (line 4). The number of times each customer $i \in L_t$ appears first is counted across all plans $\sigma_t^q \in \mathcal{P}_t$ and added incrementally to the evaluation function γ of a customer i (line 10). The ranking functions $f(\sigma_t^q)$ are computed for each plan in the pool by totaling the evaluations of each customer (line 17). A distinguished plan σ^* is chosen with the highest ranking function

$f^*(\sigma_t^q)$ at each decision during the execution of routes (lines 20-22).

The total demand served is utilized to determine actions to execute by sorting the solution pool in decreasing order of demand. This method chooses a distinguished plan with the highest demand served $\sigma^* = argmax_q(Demand(\sigma_t^q))$ throughout the planning horizon. If there are multiple pool plans with the same objective value that maximize demand, then a second-level objective is used to break ties by considering minimum distance traveled. Thus, a plan with the smallest total route duration is chosen as a distinguished plan. From now on, MPA will be called MPAc and MPAd when the consensus function and demand served are used as ranking functions, respectively.

4.5 Experiments and results

This section presents the computational study and its results. We first describe how we assess the performance of the proposed algorithms. Next, we discuss how we adapt TOP benchmark instances for DTOP and tune parameters for MPA. Finally, results are presented.

4.5.1 Performance evaluation

To evaluate the performance of the MPA and greedy algorithms for DTOP, we compute the percentage deviation of solutions produced by the online algorithms from the solutions produced by an offline algorithm for the same problem. This is similar to the *value of information* proposed by Mitrović-Minić [26]. We use percentage deviation to describe how much an online algorithm's solution deviates from an offline algorithm's solution, whereas the value of information is interpreted as the gap between the solution produced by an online algorithm and the solution produced by the same algorithm when all information is known a priori.

In this discussion, $Z_A(I)$ denotes the total collected profit when algorithm A is used to solve instance I, and $Z_B(I_{off})$ denotes the collected profit returned by algorithm B for the offline instance I_{off}. For a maximization problem, the percentage deviation $Dev_A(I)$ is computed as:

$$Dev_A(I) = \frac{Z_B(I_{off}) - Z_A(I)}{Z_B(I_{off})} \times 100. \tag{4.1}$$

The offline algorithm we use for DTOP is the CP model for TOPTW noted previously. In TOPTW, each customer has a time window $[e_i, l_i]$, such that they can not be visited earlier than e_i or later than l_i. Note that TOPTW is the offline equivalent to DTOP when $e_i = d_i$ and $l_i = T_{max}$ for each customer and $l_i = T_{max}$ for the ending node $n+1$ also. The solution to an instance of the offline (static) TOPTW thus provides an upper bound on the total profit collected in the analogous instance of the online DTOP.

There has been extensive research devoted to heuristic solution techniques to solve TOPTW as in [27, 28, 29, 30, 31, 32, 33]. A comprehensive review of TOPTW and its formulation are provided in Vansteenwegen et al. [5]. To the best of our knowledge, the latest study on the TOPTW is performed by Gedik et al. [16] with the application of an exact solution technique. They propose a constraint programming (CP) model for the TOPTW with 2.25% average gap on TOPTW benchmark instances, while additional new best-known and optimal solutions are also provided. Their results show that the proposed CP model is quite competitive with the-state-of-the-art algorithms and can be a reference method for solving variants of orienteering problems.

This comparison provides an insight into how the dynamic algorithms perform compared to the reference offline problem. The algorithms are implemented in Java and run on an Intel Core i7 CPU at 3.33GHz and 24GB of RAM. Five replications of the MPA and greedy algorithms are executed for each instance, whereas the CP model does not require any replications. The CP model computation time limit is set to 30 minutes for each instance.

4.5.2 Generating test problems

We adapt test problems for DTOP from TOP benchmark instances developed by Chao et al. [17] and available via <`www.mech.kuleuven.be/en/cib/op`>. They provide seven problem sets (p1 - p7) for TOP and the characteristics of those instances are summarized in Table 4.2.

A total of 387 TOP instances are included, each having a different value for duration limit T_{max}. The number of nodes varies between 21 and 102 and the number of vehicles varies between 2 and 4. The duration limit and coordinates and profits for each node are taken directly from the TOP instances provided by Chao et al. [17], but disclosure times must be generated for each node in order to create instances of DTOP. The performance of dynamic algorithms depends on the number of dynamic nodes and their disclosure times, among other factors such as the geographical distribution of nodes [34]. Therefore, we generate disclosure times for nodes such that the effective degree of dynamism varies among DTOP test instances [35]. Letting $\mathcal{N}$ be the set of customers, d_i the disclosure time of customer $i \in \mathcal{N}$, and n_{total} the total number of customers, Larsen [35] defines the effective degree of dynamism (δ^e) as follows:

$$\delta^e = \frac{\sum_{i \in \mathcal{N}} d_i}{n_{total}\ T_{max}}. \tag{4.2}$$

Customers known at the beginning of the planning horizon have a disclosure time of 0 ($d_i = 0$ for $i \in \mathcal{K}_0$), whereas the disclosure time for the dynamic customers in $\mathcal{D}$ can take values in the interval $(0, T_{max}]$. Therefore, the range of δ^e is $[0, 1]$ where $\delta^e = 0$ in static problems. Larsen et al. [11] categorize dynamic routing problems as weakly dynamic if $\delta^e \leq 0.3$, moderately dynamic if $0.3 < \delta^e < 0.8$, and strongly dynamic if $\delta^e \geq 0.8$. Therefore, we generate DTOP benchmark instances so that these three levels of dynamism are represented. For each of the 387 TOP instances, we generate three DTOP instances: weakly, moderately and highly dynamic. For example, for instances in TOP problem set p1, we generate DTOP problem sets p1-weakly, p1-moderately and p1-highly. This yields a total of 1161 DTOP instances.

Disclosure times are generated for each of the 1161 instances as follows. First, ρ% of nodes are randomly selected and assigned a disclosure time of 0 to represent those customers known *a priori*. For weakly and moderately dynamic instances, we use $\rho = 10\%$ and for highly dynamic instances, we use $\rho = 5\%$. All other nodes are dynamic nodes with $d_i > 0$.

Table 4.2: Static TOP benchmark data set

Problem set	Number of nodes	Number of instances	Number of vehicles	T_{max}
p1	32	54	2, 3, 4	2.5 - 42.5
p2	21	33	2, 3, 4	3.8 - 22.5
p3	33	60	2, 3, 4	3.8 - 55.0
p4	100	60	2, 3, 4	12.5 - 120.0
p5	66	78	2, 3, 4	1.2 - 65.0
p6	64	42	2, 3, 4	3.8 - 40.0
p7	102	60	2, 3, 4	50 - 200.0

We randomly generate specific disclosure times for each $i \in \mathcal{D}$ by sampling from a Normal distribution with mean $\delta^e T_{max}$ and standard deviation $T_{max}/4$. This mean is derived by rearranging 4.2 and setting δ^e as a target effective degree of dynamism that takes on values 0.3, 0.5 and 0.9 for weakly, moderately and highly dynamic instances, respectively. The standard deviation is derived by applying the methodology of Hozo [36], specifically, the standard deviation of a sample can be estimated using low (a) and high (b) ends of the range as $(b - a)/4$ for a normally distributed data set. Here, $b = T_{max}$ and $a = 0$ since customer disclosure times can take values in the range $[0, T_{max}]$. Table 4.3 illustrates the average degrees of dynamism for DTOP benchmark instances generated in this manner. These values vary between 0.20 and 0.83.

Table 4.3: Average degrees of dynamism of DTOP benchmarks

Problem Set	**Weakly**	**Moderately**	**Highly**
p1	0.20	0.55	0.82
p2	0.20	0.49	0.83
p3	0.20	0.56	0.81
p4	0.21	0.53	0.81
p5	0.21	0.52	0.81
p6	0.20	0.52	0.81
p7	0.21	0.53	0.80

4.5.3 Parameter tuning

The local search algorithm we employ within MPA is ALNS. Various parameters must be tuned for ALNS, and some of the values are taken directly from the tuned parameter values presented in Chapter 3 of this dissertation, as shown in Table 4.4. The remaining parameter values that must be tuned are the number of iterations and the pool size. The number of iterations needs to be tuned again because MPA has two phases and may require two different numbers of iterations in ALNS; one for an initial solution and one for a subsequent reoptimization. These parameters directly impact the solution quality and computational time of MPA.

To tune the number of iterations in ALNS, we have chosen nine test instances at random, which includes three instances for each level of dynamism: "dw.p1.3.1", "dw.p3.4.k", "dw.p4.2.i", "dm.p5.4.w", "dm.p6.3.l", "dm.p7.2.o", "dh.p3.2.j", "dh.p4.4.p", and "dh.p7.3.n", where dw, dm and dh stand for weak, moderate and high degrees of dynamisms, respectively. The number following "p? indicates the problem set number, the second number is the number of vehicles, and the last letter represents a specific instance of the problem set. For example, dw.p3.4.k is instance k of the problem set p3 with 4 vehicles and it is weakly dynamic.

Table 4.4: Values for ALNS parameters

Parameters	Tuning Values
Removal Parameters $[a, b]$	
Lower limit of percentage of destruction (a)	5
Upper limit of percentage of destruction (b)	60
Simulated Annealing Parameters	
Temperature	1000
Cooling Rate (α)	0.90
Scoring Parameters	
$(\sigma_1, \sigma_2, \sigma_3)$	(50, 20, 0)

We first performed tests to tune parameter $I_{initial}$ and I_{reopt}, which indicate the number of iterations of ALNS for an initial solution and subsequent reoptimization within MPA, respectively. Table 4.5 reports the average percentage deviation from the reference offline algorithm and average computation time in seconds for each $(I_{initial}, I_{reopt})$ combination over the nine chosen instances. The best combination for the number of iterations is (15000, 5000) with an average solution gap of 27.68% and a computation time of 857.60 seconds (CPU). Since the computational time is relatively high for this setting compared to others, we pick the combination of (2000, 500) without sacrificing much from the solution quality while decreasing the average computational time significantly.

Table 4.5: Parameter tuning on the number of iterations of ALNS

Instance	**(2000, 500)**	(2000, 1000)	(5000, 1000)	(5000, 2000)	(10000, 2000)	(10000, 5000)	(15000, 5000)
dw.p1.3.l	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%	10.00%
dw.p3.4.k	17.14%	20.00%	17.14%	17.14%	17.14%	17.14%	17.14%
dw.p4.2.i	2.62%	3.72%	1.93%	1.93%	2.89%	2.20%	1.93%
dm.p5.4.w	30.73%	30.73%	30.73%	30.73%	30.73%	30.73%	30.73%
dm.p6.3.l	24.79%	24.79%	24.79%	24.79%	24.79%	24.79%	24.79%
dm.p7.2.o	15.59%	14.31%	21.46%	18.60%	18.60%	14.31%	14.31%
dh.p3.2.j	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
dh.p4.4.p	52.05%	52.05%	48.54%	48.54%	52.05%	52.05%	48.54%
dh.p7.3.n	51.66%	51.66%	51.66%	51.66%	51.66%	51.66%	51.66%
Avg. Dev.	28.29%	28.58%	28.47%	28.15%	28.65%	28.10%	27.68%
Avg. Time	91.92	173.97	187.34	344.40	349.29	777.55	857.60

The selected parameters are indicated in bold.

The pool size for MPA is tuned using values between 20 and 60. The results shown in Table 4.6 use the iteration combination (2000,500). A pool size of 30 plans is selected based on the minimum percentage gap of 28.29%.

4.5.4 Computational results

For each instance, we compare the results obtained with the dynamic algorithms to the offline results generated with the CP model. Tables 4.7 through 4.10 provide summary comparisons, while Tables 4.11 through 4.18 present the results for each instance produced by each approach in detail. We report the solutions developed by the greedy algorithm, denoted Greedy, and by MPA with both demand served (MPAd) and consensus function

Table 4.6: Parameter tuning on the size of the plan pool

Instance	20	**30**	40	50	60
dw.p1.3.l	10.00%	10.00%	10.00%	10.00%	10.00%
dw.p3.4.k	20.00%	17.14%	14.29%	14.29%	17.14%
dw.p4.2.i	3.58%	2.62%	3.86%	5.10%	3.72%
dm.p5.4.w	30.73%	30.73%	30.73%	30.73%	30.73%
dm.p6.3.l	24.79%	24.79%	24.79%	24.79%	24.79%
dm.p7.2.o	18.60%	15.59%	18.60%	21.46%	20.03%
dh.p3.2.j	50.00%	50.00%	50.00%	50.00%	50.00%
dh.p4.4.p	52.05%	52.05%	57.02%	52.05%	52.05%
dh.p7.3.n	51.66%	51.66%	51.66%	51.66%	51.66%
Avg. Dev.	29.05%	28.29%	28.99%	28.90%	28.90%
Avg. Time	60.12	89.92	110.35	149.09	175.78

The selected parameter is indicated in bold.

(MPAc) as the ranking functions.

Table 4.7 summarizes the results for DTOP benchmark instances based on the level of dynamism. The first column, #INS, provides the total number of test instances grouped in each level of dynamism. The second column, #Best, reports the number of instances for which each algorithm obtains the best solution (best among the three online algorithms). The third column reports the average percentage of the available profit collected by each algorithm, including the offline algorithm, across all instances having each level of dynamism. According to the number of instances where an algorithm obtains the best solution, MPAd is quite effective, finding the best solution in 1017 out of 1161 DTOP benchmark instances. This is compared with 924 instances for MPAc and 637 for Greedy. The superior performance of MPAd over MPAc is likely due to the fact that the ranking function in MPAd uses the profit collected as a decision criterion when selecting a distinguished plan, while MPAc does not. The offline algorithm, CP, collects on average 26.6% of total profits available in a problem instance. Recall that many of the CP model solutions are optimal offline solutions. The reason percent profit is low is because the time budget and disclosure times of customers limit the ability to visit all customers. Among the online algorithms, MPAd produces solutions with the highest average percent of profit collected, indicating better performance according to this metric. However, the distinction between it and MPAc is not great across the levels of dynamism. For example, the average percents of total profit collected for MPAd and MPAc are 34.6% and 34.3% for weakly dynamic instances, respectively, and 20.1% versus 20.0% for moderately dynamic instances. For highly dynamic instances, the average percents of total profit collected for MPAd and MPAc are equal. Both MPAd and MPAc outperform Greedy on this metric across all levels of dynamism. To see the impact of the degree of dynamism in general, note that the percent of total profit collected decreases as the degree of dynamism increases. This is not surprising, as the instances with higher levels of dynamism are more constrained in the sense that there is less time, on average, between the disclosure times of dynamic customers and the allowable time budget.

Table 4.7: Summary comparison of the algorithms

Degree of Dynamism	#INS	# Best			% Total Profit Collected			
		MPAd	MPAc	Greedy	Offline	MPAd	MPAc	Greedy
Weakly	387	319	257	123	39.6%	34.6%	34.3%	29.3%
Moderately	387	329	301	193	29.0%	20.1%	20.0%	17.8%
Highly	387	369	367	321	11.3%	5.6%	5.6%	5.5%
All	1161	1017	925	637	26.6%	20.1%	20.0%	17.5%

Table 4.8 illustrates the average percentage deviation of collected profit associated with solutions produced by the dynamic algorithms from the reference offline solutions, across all instances for each problem set. It can be observed that the gaps associated with the average deviation from the offline solution are relatively high for all algorithms. This is due to the fact that the offline reference approach benefits from knowing in advance the disclosure time of customers (i.e. it produces a solution in hindsight) while the other algorithms produce a solution based on a myopic approach. Thus, the offline algorithm yields upper bounds on the objective value of the corresponding DTOP instances, and the average percentage deviation provides a performance comparison among the algorithms instead of an absolute performance metric. For a weakly dynamic instance, when most of the customers are known

a priori, the results produced by the dynamic algorithms more closely resemble those of the offline algorithm. For example, when the level of dynamism decreases from highly dynamic to weakly dynamic, the average deviation of MPAd from the offline algorithm decreases from 42.8% to 14.0%. Among the online algorithms, when the degree of dynamism is weak, the preference for MPAd is more pronounced. For example, the average deviation of MPAd is 14.0% for weakly dynamic instances whereas the average deviations of MPAc and Greedy are 14.7% and 23.8%, respectively. But, when the level of dynamism is moderate, the average deviations of MPAd, MPAc, and Greedy are 32.7%, 32.8%, and 37.4%, respectively. This is because the reaction time to a new customer, which is the difference between its disclosure time and the end of the planning period, is considerably longer when an instance is weakly dynamic. There is more time available, on average, to react to newly revealed dynamic customers in order to reoptimize the routing plans. On average, the algorithms perform best on problem set p6 compared with other problem sets. For example, the average deviations of MPAd from the offline algorithm are 6.7% and 17.8% for p6 and p3, respectively.

Note that the values reported in Tables 4.7 and 4.8 are average deviations are from the solutions obtained by the CP model, which are not always optimal for the offline problem. The CP model finds 925 optimal and 236 non-optimal solutions. Tables 4.19 and 4.20 in Appendix A illustrate the average percentage deviations from the optimal and the non-optimal offline solutions, respectively. Because CP finds all the optimal solutions for the problem sets of p1,p2, and p3 and for highly dynamic instances, the average percentage deviations in Table 4.19 are the same as reported in Table 4.8, while Table 4.20 does not report any values for those instances. As expected, it can be observed that the overall gaps associated with the average deviation from the optimal offline solution for weakly and moderately dynamic instances increase while the overall deviations from the non-optimal solution decrease. For example, the average deviation of MPAd for weakly dynamic instances increases from 14.0% to 17.9% when comparing the overall offline solutions to the optimal offline solutions, and it decreases from 14.0% to 8.5% when comparing the overall offline solutions to the non-optimal offline solutions.

Table 4.8: Average deviation from offline algorithm

Problem Set	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
p1	17.5%	18.0%	22.8%	40.6%	40.8%	43.5%	39.2%	39.2%	39.5%
p2	15.0%	15.8%	20.9%	45.4%	45.5%	46.1%	51.2%	51.2%	51.2%
p3	17.8%	18.0%	26.8%	39.8%	39.1%	41.7%	46.6%	46.6%	47.2%
p4	11.4%	11.8%	28.8%	23.6%	23.9%	34.6%	48.8%	48.5%	50.6%
p5	13.5%	14.4%	22.2%	31.9%	32.5%	38.7%	47.1%	46.9%	47.6%
p6	6.7%	7.3%	11.9%	16.8%	17.1%	18.9%	25.7%	25.7%	28.0%
p7	16.3%	17.6%	32.9%	30.5%	30.9%	38.4%	41.0%	41.6%	42.8%
Average	14.0%	14.7%	23.8%	32.7%	32.8%	37.4%	42.8%	42.8%	43.8%

The impact of request volume on the algorithms can also be assessed. Three levels of the total number of customers are considered, with low indicating fewer than 33 customers, medium indicating around 66 customers, and high indicating more than 99 customers. The instance classification is shown in Table 4.9, where the problem sets of p1, p2 and p3 are classified as low, the problem set of p5 and p6 are classified as medium and the problem sets of p4 and p6 are classified as high. Table 4.9 indicates that the algorithms produce solutions with lower average deviation from offline solutions when the request volume is medium. The relative performance of Greedy when compared with the others tends to worsen as the level of dynamism decreases and the request volume increases. For example, when the request volume increases from low to high and the dynamism is weak, the difference between Greedy and MPAd increases from 6.7% (23.5%-16.8%) to 17.1% (30.9%-13.8%). When the degree of dynamism increases from weak to high and the request volume is high, the difference between Greedy and MPAd decreases from 17.1% (30.9%-13.8%) to 1.8% (46.7%-44.9%). In general, the algorithms achieve a better average gap on high request volume instances than on low request volume instances. For example, when the level of dynamism is moderate, the average deviations of MPAc are 27.4% and 41.8% for high and low request volumes, respectively, because more customers become available at the beginning of the time horizon when the request volume is high than when it is low. Therefore, the algorithms produce a better initial solution with more available customers.

Tables 4.21 and 4.22 in Appendix A illustrate the impact of request volume on the algorithms when the optimal and non-optimal offline solutions are considered, respectively. When the request volume is medium and high, the average deviations increase from the optimal offline solutions while they decrease from non-optimal offline solutions for weakly and moderately dynamic instances. For example, when the level of dynamism is moderate and the request volume is high, the average deviation of MPAd increases from 13.8% to 25.6% when comparing the overall offline solutions to the optimal offline solutions and it decreases from 13.8% to 7.6% when comparing the overall offline solutions to the non-optimal offline solutions.

Table 4.9: Average deviation on request volume

Request Volume	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
Low (p1, p2 and p3)	16.8%	17.3%	23.5%	41.9%	41.8%	43.8%	45.7%	45.7%	46.0%
Medium (p5 and p6)	10.1%	10.9%	17.0%	24.4%	24.8%	28.8%	36.4%	36.3%	37.8%
High (p4 and p7)	13.8%	14.7%	30.9%	27.1%	27.4%	36.5%	44.9%	45.1%	46.7%

Table 4.10 reports the average computational time of the algorithms in CPU seconds for each set of instances. In general, as the level of dynamism increases, the average computational time greatly decreases. For example, MPAd has a high computational time of 159.2 seconds for weakly dynamic instances while its computational time is only 0.68 seconds for highly dynamic instances. This is because the runtime increases proportionally to the number of customers that have been disclosed but not yet visited at a particular simulation clock time. Because the weakly dynamic instances contain more static customers than highly dynamic instances, and also contain more customers with earlier disclosure times than moderately and highly dynamic instances, their runtimes are much longer. Because the greedy algorithm maintains a single routing plan throughout the planning horizon, the time taken to solve an instance is on average less than a second. On average, the computational

times for MPAd are slightly higher than the computational times for MPAc, with an average difference of 8 seconds or less. The higher runtimes for MPAd are likely due to the larger number of plans that must be deleted and regenerated at each planning stage. That is, with MPAc, the consensus function chooses a distinguished plan most similar to others in the pool. Therefore, many of the plans in the pool are already consistent with the distinguished plan and do not need to be modified. However, in MPAd, the distinguished plan is selected based on the collected profit and there may not be other plans in the pool similar to it. To see the impact of total request volume (number of customers) on runtimes, note that problem sets p4 and p7, the high request volume instances, have the highest computational times.

Table 4.10: Average CPU (seconds)

Problem Set	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
p1	5.05	5.19	0.06	0.66	0.67	0.01	0.19	0.19	0.01
p2	1.48	1.40	0.05	0.26	0.28	0.01	0.09	0.10	0.00
p3	7.76	7.94	0.08	1.12	1.05	0.02	0.26	0.25	0.01
p4	601.76	578.82	0.97	66.95	67.52	0.08	1.89	1.77	0.02
p5	96.55	88.98	0.27	16.75	16.04	0.05	0.57	0.56	0.01
p6	57.79	53.51	0.16	7.89	7.47	0.03	0.36	0.36	0.01
p7	344.03	322.60	0.76	46.95	45.67	0.07	1.38	1.40	0.02
Average	159.20	151.21	0.33	20.08	19.81	0.04	0.68	0.66	0.01

Finally, detailed results across five replicates for the DTOP benchmark instances are given in Tables 4.11-4.18 for each algorithm. In these tables, the first column represents the DTOP benchmark instance number. The Z_{CP} column reports the objective values of offline solutions produced by the CP for the static variant of DTOP. If CP obtained the optimal solution for an offline instance, the value for Z_{CP} is indicated in bold. For the three online algorithms, the Z column gives the best solution value (again bold indicates optimal) across five replicates for an algorithm and the CPU column reports the average computational time across five replicates in seconds. Note that the reported computational times cannot guarantee to find the best solution. To obtain the best solution, the algorithm is run five times, and the CPU column indicates the average runtime of these five replicates. Finally, %Dev presents the percentage deviation of an algorithm?s best solution across five replicates to the offline algorithm?s solution. The objective value of some instances is found to be 0 because the DTOP instances are modified from the TOP benchmark instances which originally also have the objective value of 0 for the same instances because it is not feasible to visit any of the customers due to a tight time limit. On average, the average deviation for the MPAd, MPAc, and greedy algorithms are 29.8%, 30.1%, and 35.0%, respectively. Based on these results, in general, MPAd outperforms MPAc and Greedy. However, MPAc finds 50 known optimal solutions whereas MPAd finds 48 optimal solutions when the solution values of 0 are excluded. It is interesting to note that Greedy can also find 43 optimal solutions. For example, in the instance of dw.p1.3.g, all three algorithms find the optimal solutions. For such instances, it is relatively easy to find the optimal solutions because either the time limit, T_{max}, or the number of vehicles are large enough to visit all customer locations in the offline instances, which indicates an excess of capacity compared to the demand present. In addition to the number of optimal solutions found, there is a tradeoff between the computational time and the solution quality. MPAd and MPAc are capable of producing good solutions with a low average percent deviation from the offline algorithm on the DTOP benchmark instances within a reasonable amount of computational time.

Table 4.11: Instance p1

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dw.p1.2.a	0	0	0.26	0.00	0	0.27	0.00	0	0.06	0.00
dw.p1.2.b	5	5	0.88	0.00	5	0.87	0.00	5	0.06	0.00
dw.p1.2.c	20	20	0.81	0.00	20	0.78	0.00	20	0.03	0.00
dw.p1.2.d	30	25	0.93	0.17	25	0.92	0.17	25	0.05	0.17
dw.p1.2.e	45	25	0.97	0.44	25	0.94	0.44	25	0.05	0.44
dw.p1.2.f	70	60	1.54	0.14	60	1.13	0.14	55	0.06	0.21
dw.p1.2.g	80	65	1.90	0.19	60	1.90	0.25	65	0.06	0.19
dw.p1.2.h	110	105	5.76	0.05	105	5.66	0.05	80	0.06	0.27
dw.p1.2.i	130	105	4.75	0.19	90	4.32	0.31	80	0.07	0.38
dw.p1.2.j	145	120	5.21	0.17	115	5.13	0.21	85	0.05	0.41
dw.p1.2.k	150	135	6.46	0.10	135	5.99	0.10	110	0.05	0.27
dw.p1.2.l	185	170	12.00	0.08	170	11.57	0.08	160	0.07	0.14
dw.p1.2.m	200	160	8.18	0.20	160	9.04	0.20	135	0.03	0.33
dw.p1.2.n	205	150	9.08	0.27	150	9.12	0.27	135	0.06	0.34
dw.p1.2.o	230	190	9.51	0.17	185	10.34	0.20	190	0.09	0.17
dw.p1.2.p	230	205	13.97	0.11	195	12.56	0.15	170	0.06	0.26
dw.p1.2.q	240	205	12.41	0.15	205	12.73	0.15	185	0.03	0.23
dw.p1.2.r	260	210	12.10	0.19	205	12.12	0.21	195	0.12	0.25
dw.p1.3.a	0	0	0.21	0.00	0	0.22	0.00	0	0.01	0.00
dw.p1.3.b	0	0	0.24	0.00	0	0.23	0.00	0	0.01	0.00
dw.p1.3.c	15	0	0.24	1.00	0	0.21	1.00	0	0.01	1.00
dw.p1.3.d	15	15	1.92	0.00	15	1.92	0.00	15	0.07	0.00
dw.p1.3.e	30	25	0.88	0.17	25	0.78	0.17	25	0.04	0.17
dw.p1.3.f	40	25	1.48	0.38	25	1.68	0.38	25	0.05	0.38
dw.p1.3.g	50	50	0.89	0.00	50	0.85	0.00	50	0.04	0.00
dw.p1.3.h	70	50	2.84	0.29	50	3.10	0.29	50	0.09	0.29
dw.p1.3.i	100	75	5.33	0.25	75	5.76	0.25	75	0.12	0.25
dw.p1.3.j	100	90	5.25	0.10	85	5.35	0.15	75	0.11	0.25
dw.p1.3.k	130	105	6.11	0.19	105	6.47	0.19	105	0.08	0.19
dw.p1.3.l	150	135	7.14	0.10	135	7.31	0.10	125	0.11	0.17
dw.p1.3.m	165	145	10.80	0.12	155	11.00	0.06	120	0.09	0.27
dw.p1.3.n	185	130	8.64	0.30	130	8.82	0.30	95	0.07	0.49
dw.p1.3.o	190	165	10.78	0.13	160	9.94	0.16	135	0.10	0.29
dw.p1.3.p	195	155	7.20	0.21	155	8.29	0.21	120	0.07	0.38
dw.p1.3.q	215	175	9.37	0.19	185	9.35	0.14	165	0.06	0.23
dw.p1.3.r	240	230	10.38	0.04	230	12.10	0.04	205	0.06	0.15
dw.p1.4.a	0	0	0.21	0.00	0	0.30	0.00	0	0.01	0.00
dw.p1.4.b	0	0	0.28	0.00	0	0.39	0.00	0	0.02	0.00
dw.p1.4.c	0	0	0.30	0.00	0	0.30	0.00	0	0.01	0.00
dw.p1.4.d	15	5	1.27	0.67	5	1.20	0.67	5	0.04	0.67
dw.p1.4.e	15	15	1.55	0.00	15	1.52	0.00	15	0.06	0.00
dw.p1.4.f	25	20	1.96	0.20	20	1.76	0.20	20	0.08	0.20
dw.p1.4.g	35	20	1.05	0.43	20	0.96	0.43	20	0.04	0.43
dw.p1.4.h	45	40	1.79	0.11	40	1.77	0.11	40	0.07	0.11
dw.p1.4.i	60	50	1.97	0.17	50	1.89	0.17	50	0.08	0.17
dw.p1.4.j	70	70	2.92	0.00	70	2.82	0.00	65	0.10	0.07
dw.p1.4.k	95	65	5.24	0.32	65	6.18	0.32	50	0.04	0.47
dw.p1.4.l	105	80	5.86	0.24	80	6.19	0.24	75	0.12	0.29
dw.p1.4.m	120	100	7.34	0.17	100	7.70	0.17	100	0.11	0.17
dw.p1.4.n	150	125	8.94	0.17	125	9.08	0.17	120	0.08	0.20
dw.p1.4.o	160	130	9.72	0.19	130	10.23	0.19	125	0.07	0.22
dw.p1.4.p	165	150	10.49	0.09	150	11.86	0.09	140	0.09	0.15
dw.p1.4.q	170	115	7.32	0.32	115	8.09	0.32	110	0.06	0.35
dw.p1.4.r	200	140	8.27	0.30	140	9.53	0.30	150	0.09	0.25

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dm.p1.2.a	0	0	0.10	0.00	0	0.09	0.00	0	0.01	0.00
dm.p1.2.b	15	0	0.11	1.00	0	0.10	1.00	0	0.01	1.00
dm.p1.2.c	15	0	0.11	1.00	0	0.11	1.00	0	0.01	1.00
dm.p1.2.d	10	5	0.29	0.50	5	0.22	0.50	5	0.01	0.50
dm.p1.2.e	40	15	0.29	0.63	15	0.24	0.63	15	0.01	0.63
dm.p1.2.f	45	25	0.34	0.44	25	0.25	0.44	25	0.01	0.44
dm.p1.2.g	55	30	0.30	0.45	30	0.27	0.45	25	0.01	0.55
dm.p1.2.h	85	50	0.41	0.41	50	0.35	0.41	45	0.01	0.47
dm.p1.2.i	85	45	0.48	0.47	45	0.44	0.47	45	0.01	0.47
dm.p1.2.j	105	65	1.03	0.38	60	0.82	0.43	60	0.02	0.43
dm.p1.2.k	110	70	0.62	0.36	75	0.74	0.32	70	0.01	0.36
dm.p1.2.l	130	65	0.79	0.50	75	0.91	0.42	65	0.01	0.50
dm.p1.2.m	135	115	1.11	0.15	115	1.28	0.15	85	0.01	0.37
dm.p1.2.n	140	100	1.20	0.29	100	1.23	0.29	85	0.01	0.39
dm.p1.2.o	160	130	2.34	0.19	135	2.66	0.16	85	0.01	0.47
dm.p1.2.p	160	110	1.49	0.31	120	1.67	0.25	90	0.01	0.44
dm.p1.2.q	165	110	1.04	0.33	95	1.10	0.42	80	0.01	0.52
dm.p1.2.r	180	140	1.98	0.22	140	2.80	0.22	125	0.02	0.31
dm.p1.3.a	0	0	0.11	0.00	0	0.10	0.00	0	0.00	0.00
dm.p1.3.b	0	0	0.14	0.00	0	0.16	0.00	0	0.01	0.00
dm.p1.3.c	0	0	0.18	0.00	0	0.17	0.00	0	0.01	0.00
dm.p1.3.d	15	5	0.19	0.67	5	0.13	0.67	5	0.01	0.67
dm.p1.3.e	20	5	0.14	0.75	5	0.21	0.75	5	0.00	0.75
dm.p1.3.f	25	15	0.33	0.40	15	0.41	0.40	15	0.01	0.40
dm.p1.3.g	40	20	0.68	0.50	20	0.70	0.50	20	0.02	0.50
dm.p1.3.h	60	25	0.56	0.58	25	0.45	0.58	25	0.02	0.58
dm.p1.3.i	60	30	0.66	0.50	30	0.85	0.50	30	0.02	0.50
dm.p1.3.j	95	60	0.64	0.37	60	0.57	0.37	60	0.01	0.37
dm.p1.3.k	95	50	0.56	0.47	50	0.60	0.47	50	0.02	0.47
dm.p1.3.l	95	60	0.71	0.37	60	0.82	0.37	55	0.02	0.42
dm.p1.3.m	105	70	0.66	0.33	70	0.77	0.33	70	0.01	0.33
dm.p1.3.n	130	60	0.86	0.54	60	0.74	0.54	60	0.02	0.54
dm.p1.3.o	145	100	1.18	0.31	105	0.75	0.28	105	0.02	0.28
dm.p1.3.p	140	70	0.78	0.50	65	0.68	0.54	65	0.02	0.54
dm.p1.3.q	150	115	0.96	0.23	115	1.11	0.23	85	0.02	0.43
dm.p1.3.r	170	105	1.07	0.38	85	0.66	0.50	95	0.01	0.44
dm.p1.4.a	0	0	0.13	0.00	0	0.16	0.00	0	0.01	0.00
dm.p1.4.b	0	0	0.17	0.00	0	0.17	0.00	0	0.01	0.00
dm.p1.4.c	0	0	0.19	0.00	0	0.14	0.00	0	0.01	0.00
dm.p1.4.d	0	0	0.19	0.00	0	0.16	0.00	0	0.00	0.00
dm.p1.4.e	10	0	0.15	1.00	0	0.15	1.00	0	0.01	1.00
dm.p1.4.f	0	0	0.17	0.00	0	0.23	0.00	0	0.01	0.00
dm.p1.4.g	30	20	0.61	0.33	20	0.49	0.33	20	0.02	0.33
dm.p1.4.h	30	5	0.46	0.83	5	0.49	0.83	5	0.02	0.83
dm.p1.4.i	40	5	0.44	0.88	5	0.47	0.88	5	0.01	0.88
dm.p1.4.j	70	35	0.77	0.50	35	0.69	0.50	35	0.02	0.50
dm.p1.4.k	85	35	1.02	0.59	35	0.83	0.59	35	0.02	0.59
dm.p1.4.l	70	45	0.80	0.36	45	0.73	0.36	45	0.02	0.36
dm.p1.4.m	95	40	0.78	0.58	40	0.71	0.58	40	0.02	0.58
dm.p1.4.n	90	40	1.06	0.56	40	1.32	0.56	40	0.02	0.56
dm.p1.4.o	105	65	1.12	0.38	65	0.88	0.38	65	0.02	0.38
dm.p1.4.p	120	65	1.02	0.46	65	1.00	0.46	65	0.01	0.46
dm.p1.4.q	130	80	0.95	0.38	75	1.26	0.42	75	0.02	0.42
dm.p1.4.r	135	60	0.99	0.56	60	0.89	0.56	60	0.02	0.56

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dh.p1.2.a	0	0	0.09	0.00	0	0.08	0.00	0	0.00	0.00
dh.p1.2.b	0	0	0.08	0.00	0	0.08	0.00	0	0.01	0.00
dh.p1.2.c	0	0	0.10	0.00	0	0.09	0.00	0	0.00	0.00
dh.p1.2.d	5	0	0.13	1.00	0	0.12	1.00	0	0.00	1.00
dh.p1.2.e	5	5	0.11	0.00	5	0.12	0.00	5	0.00	0.00
dh.p1.2.f	25	15	0.17	0.40	15	0.20	0.40	15	0.01	0.40
dh.p1.2.g	35	15	0.18	0.57	15	0.15	0.57	15	0.01	0.57
dh.p1.2.h	20	5	0.10	0.75	5	0.08	0.75	5	0.00	0.75
dh.p1.2.i	30	20	0.18	0.33	20	0.16	0.33	20	0.00	0.33
dh.p1.2.j	50	35	0.25	0.30	35	0.20	0.30	35	0.01	0.30
dh.p1.2.k	45	15	0.21	0.67	15	0.18	0.67	15	0.01	0.67
dh.p1.2.l	45	15	0.18	0.67	15	0.13	0.67	15	0.00	0.67
dh.p1.2.m	50	35	0.29	0.30	35	0.35	0.30	35	0.00	0.30
dh.p1.2.n	50	15	0.22	0.70	15	0.29	0.70	15	0.01	0.70
dh.p1.2.o	75	30	0.13	0.60	30	0.15	0.60	30	0.01	0.60
dh.p1.2.p	80	50	0.25	0.38	50	0.35	0.38	50	0.01	0.38
dh.p1.2.q	40	20	0.17	0.50	20	0.23	0.50	15	0.01	0.63
dh.p1.2.r	60	40	0.18	0.33	40	0.27	0.33	40	0.01	0.33
dh.p1.3.a	0	0	0.08	0.00	0	0.08	0.00	0	0.01	0.00
dh.p1.3.b	0	0	0.11	0.00	0	0.09	0.00	0	0.01	0.00
dh.p1.3.c	0	0	0.10	0.00	0	0.15	0.00	0	0.01	0.00
dh.p1.3.d	5	0	0.10	1.00	0	0.11	1.00	0	0.01	1.00
dh.p1.3.e	0	0	0.11	0.00	0	0.11	0.00	0	0.01	0.00
dh.p1.3.f	0	0	0.12	0.00	0	0.11	0.00	0	0.01	0.00
dh.p1.3.g	15	5	0.22	0.67	5	0.19	0.67	5	0.01	0.67
dh.p1.3.h	20	5	0.18	0.75	5	0.21	0.75	5	0.01	0.75
dh.p1.3.i	25	15	0.18	0.40	15	0.24	0.40	15	0.01	0.40
dh.p1.3.j	15	10	0.17	0.33	10	0.18	0.33	10	0.00	0.33
dh.p1.3.k	15	5	0.18	0.67	5	0.20	0.67	5	0.01	0.67
dh.p1.3.l	45	10	0.19	0.78	10	0.21	0.78	10	0.01	0.78
dh.p1.3.m	20	15	0.19	0.25	15	0.16	0.25	15	0.01	0.25
dh.p1.3.n	45	15	0.16	0.67	15	0.15	0.67	15	0.01	0.67
dh.p1.3.o	60	20	0.34	0.67	20	0.35	0.67	20	0.01	0.67
dh.p1.3.p	50	15	0.22	0.70	15	0.16	0.70	15	0.01	0.70
dh.p1.3.q	65	20	0.20	0.69	20	0.19	0.69	20	0.01	0.69
dh.p1.3.r	75	40	0.57	0.47	40	0.54	0.47	40	0.01	0.47
dh.p1.4.a	0	0	0.10	0.00	0	0.16	0.00	0	0.01	0.00
dh.p1.4.b	0	0	0.11	0.00	0	0.11	0.00	0	0.01	0.00
dh.p1.4.c	0	0	0.11	0.00	0	0.17	0.00	0	0.01	0.00
dh.p1.4.d	0	0	0.11	0.00	0	0.18	0.00	0	0.01	0.00
dh.p1.4.e	0	0	0.18	0.00	0	0.16	0.00	0	0.00	0.00
dh.p1.4.f	0	0	0.11	0.00	0	0.11	0.00	0	0.01	0.00
dh.p1.4.g	0	0	0.12	0.00	0	0.12	0.00	0	0.01	0.00
dh.p1.4.h	5	5	0.31	0.00	5	0.38	0.00	5	0.02	0.00
dh.p1.4.i	10	10	0.30	0.00	10	0.21	0.00	10	0.02	0.00
dh.p1.4.j	15	10	0.13	0.33	10	0.12	0.33	10	0.00	0.33
dh.p1.4.k	30	5	0.25	0.83	5	0.22	0.83	5	0.02	0.83
dh.p1.4.l	20	5	0.19	0.75	5	0.18	0.75	5	0.01	0.75
dh.p1.4.m	20	5	0.24	0.75	5	0.23	0.75	5	0.01	0.75
dh.p1.4.n	30	10	0.23	0.67	10	0.22	0.67	10	0.02	0.67
dh.p1.4.o	25	15	0.31	0.40	15	0.32	0.40	15	0.01	0.40
dh.p1.4.p	40	10	0.35	0.75	10	0.26	0.75	10	0.02	0.75
dh.p1.4.q	10	5	0.22	0.50	5	0.23	0.50	5	0.02	0.50
dh.p1.4.r	30	10	0.25	0.67	10	0.32	0.67	10	0.02	0.67

Table 4.12: Instance p2

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dw.p2.2.a	**90**	70	1.09	0.22	70	1.07	0.22	70	0.24	0.22
dw.p2.2.b	**120**	95	1.20	0.21	95	1.23	0.21	85	0.06	0.29
dw.p2.2.c	**130**	110	1.10	0.15	110	1.05	0.15	110	0.05	0.15
dw.p2.2.d	**160**	135	0.94	0.16	130	0.89	0.19	120	0.02	0.25
dw.p2.2.e	**180**	160	2.51	0.11	155	2.35	0.14	155	0.06	0.14
dw.p2.2.f	**200**	160	1.54	0.20	160	1.46	0.20	145	0.03	0.28
dw.p2.2.g	**200**	180	1.48	0.10	180	1.36	0.10	150	0.02	0.25
dw.p2.2.h	**230**	155	1.32	0.33	155	1.17	0.33	130	0.02	0.43
dw.p2.2.i	**200**	**200**	2.44	0.00	**200**	2.36	0.00	145	0.06	0.28
dw.p2.2.j	**235**	200	1.80	0.15	200	2.41	0.15	180	0.03	0.23
dw.p2.2.k	**240**	210	1.99	0.13	210	1.91	0.13	185	0.05	0.23
dw.p2.3.a	**70**	**70**	1.56	0.00	**70**	1.48	0.00	**70**	0.06	0.00
dw.p2.3.b	**70**	**70**	1.13	0.00	**70**	1.06	0.00	**70**	0.04	0.00
dw.p2.3.c	**90**	70	0.77	0.22	70	0.75	0.22	70	0.02	0.22
dw.p2.3.d	**105**	85	1.12	0.19	85	1.01	0.19	75	0.04	0.29
dw.p2.3.e	**120**	90	1.18	0.25	90	1.13	0.25	90	0.04	0.25
dw.p2.3.f	**105**	80	0.93	0.24	60	0.88	0.43	60	0.04	0.43
dw.p2.3.g	**145**	105	1.36	0.28	105	1.25	0.28	80	0.04	0.45
dw.p2.3.h	**160**	120	1.52	0.25	120	1.39	0.25	120	0.02	0.25
dw.p2.3.i	**200**	165	2.74	0.18	165	2.68	0.18	165	0.07	0.18
dw.p2.3.j	**200**	140	1.47	0.30	140	1.05	0.30	120	0.02	0.40
dw.p2.3.k	**200**	180	2.62	0.10	180	1.99	0.10	170	0.04	0.15
dw.p2.4.a	**10**	**10**	0.77	0.00	**10**	0.72	0.00	**10**	0.03	0.00
dw.p2.4.b	**70**	**70**	1.33	0.00	**70**	1.26	0.00	**70**	0.07	0.00
dw.p2.4.c	**70**	60	0.50	0.14	60	0.48	0.14	60	0.02	0.14
dw.p2.4.d	**70**	**70**	1.13	0.00	**70**	1.07	0.00	**70**	0.04	0.00
dw.p2.4.e	**70**	**70**	0.99	0.00	**70**	0.92	0.00	**70**	0.03	0.00
dw.p2.4.f	**105**	70	1.39	0.33	70	1.31	0.33	70	0.06	0.33
dw.p2.4.g	**90**	70	1.26	0.22	70	1.16	0.22	70	0.04	0.22
dw.p2.4.h	**105**	90	1.42	0.14	90	1.53	0.14	90	0.06	0.14
dw.p2.4.i	**105**	**105**	1.56	0.00	**105**	1.55	0.00	85	0.03	0.19
dw.p2.4.j	**105**	**105**	2.68	0.00	**105**	2.35	0.00	**105**	0.04	0.00
dw.p2.4.k	**165**	105	1.93	0.36	105	1.83	0.36	80	0.03	0.52

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dm.p2.2.a	**55**	35	0.23	0.36	35	0.22	36.4%	35	0.02	0.36
dm.p2.2.b	**90**	35	0.18	0.61	35	0.15	61.1%	35	0.01	0.61
dm.p2.2.c	**110**	55	0.14	0.50	55	0.16	50.0%	55	0.00	0.50
dm.p2.2.d	**115**	65	0.18	0.43	65	0.18	43.5%	65	0.01	0.43
dm.p2.2.e	**110**	40	0.35	0.64	40	0.35	63.6%	40	0.02	0.64
dm.p2.2.f	**95**	70	0.23	0.26	70	0.23	26.3%	70	0.00	0.26
dm.p2.2.g	**160**	100	0.19	0.38	100	0.19	37.5%	75	0.01	0.53
dm.p2.2.h	**185**	85	0.32	0.54	85	0.35	54.1%	75	0.01	0.59
dm.p2.2.i	**190**	145	0.64	0.24	145	0.79	23.7%	145	0.02	0.24
dm.p2.2.j	**180**	170	0.74	0.06	170	0.88	5.6%	170	0.01	0.06
dm.p2.2.k	**240**	145	0.60	0.40	135	0.43	43.8%	135	0.01	0.44
dm.p2.3.a	**45**	10	0.13	0.78	10	0.15	77.8%	10	0.01	0.78
dm.p2.3.b	**60**	10	0.41	0.83	10	0.49	83.3%	10	0.02	0.83
dm.p2.3.c	**80**	25	0.14	0.69	25	0.16	68.8%	25	0.01	0.69
dm.p2.3.d	**20**	10	0.13	0.50	10	0.16	50.0%	10	0.01	0.50
dm.p2.3.e	**50**	**50**	0.17	0.00	**50**	0.21	0.0%	**50**	0.01	0.00
dm.p2.3.f	**80**	60	0.19	0.25	60	0.20	25.0%	60	0.01	0.25
dm.p2.3.g	**110**	50	0.23	0.55	50	0.25	54.5%	50	0.01	0.55
dm.p2.3.h	**110**	60	0.20	0.45	60	0.24	45.5%	60	0.01	0.45
dm.p2.3.i	**160**	80	0.34	0.50	80	0.29	50.0%	80	0.01	0.50
dm.p2.3.j	**120**	80	0.24	0.33	80	0.24	33.3%	80	0.01	0.33
dm.p2.3.k	**125**	70	0.25	0.44	70	0.26	44.0%	70	0.01	0.44
dm.p2.4.a	**10**	0	0.17	1.00	0	0.19	100.0%	0	0.01	1.00
dm.p2.4.b	**35**	10	0.15	0.71	10	0.18	71.4%	10	0.01	0.71
dm.p2.4.c	**55**	35	0.16	0.36	35	0.19	36.4%	35	0.01	0.36
dm.p2.4.d	**50**	10	0.15	0.80	10	0.18	80.0%	10	0.01	0.80
dm.p2.4.e	**35**	25	0.16	0.29	25	0.18	28.6%	25	0.01	0.29
dm.p2.4.f	**50**	25	0.16	0.50	25	0.19	50.0%	25	0.01	0.50
dm.p2.4.g	**80**	55	0.60	0.31	55	0.72	31.3%	55	0.02	0.31
dm.p2.4.h	**60**	**60**	0.24	0.00	**60**	0.28	0.0%	**60**	0.01	0.00
dm.p2.4.i	**25**	10	0.16	0.60	10	0.17	60.0%	10	0.01	0.60
dm.p2.4.j	**95**	75	0.33	0.21	75	0.36	21.1%	75	0.01	0.21
dm.p2.4.k	**45**	25	0.15	0.44	25	0.14	44.4%	25	0.01	0.44

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dh.p2.2.a	**10**	0	0.08	1.00	0	0.08	1.00	0	0.00	1.00
dh.p2.2.b	**25**	0	0.09	1.00	0	0.09	1.00	0	0.01	1.00
dh.p2.2.c	**25**	0	0.08	1.00	0	0.09	1.00	0	0.00	1.00
dh.p2.2.d	**45**	20	0.08	0.56	20	0.09	0.56	20	0.00	0.56
dh.p2.2.e	**10**	**10**	0.08	0.00	**10**	0.11	0.00	**10**	0.00	0.00
dh.p2.2.f	**65**	10	0.08	0.85	10	0.09	0.85	10	0.00	0.85
dh.p2.2.g	**20**	0	0.08	1.00	0	0.08	1.00	0	0.00	1.00
dh.p2.2.h	**35**	25	0.08	0.29	25	0.07	0.29	25	0.00	0.29
dh.p2.2.i	**80**	35	0.10	0.56	35	0.12	0.56	35	0.00	0.56
dh.p2.2.j	**65**	45	0.11	0.31	45	0.11	0.31	45	0.00	0.31
dh.p2.2.k	**80**	10	0.10	0.88	10	0.12	0.88	10	0.00	0.88
dh.p2.3.a	**15**	0	0.08	1.00	0	0.10	1.00	0	0.01	1.00
dh.p2.3.b	**10**	0	0.09	1.00	0	0.10	1.00	0	0.01	1.00
dh.p2.3.c	**0**	**0**	0.08	0.00	**0**	0.08	0.00	**0**	0.00	0.00
dh.p2.3.d	**0**	**0**	0.09	0.00	**0**	0.08	0.00	**0**	0.00	0.00
dh.p2.3.e	**10**	**10**	0.14	0.00	**10**	0.14	0.00	**10**	0.01	0.00
dh.p2.3.f	**10**	0	0.09	1.00	0	0.11	1.00	0	0.01	1.00
dh.p2.3.g	**20**	0	0.09	1.00	0	0.11	1.00	0	0.01	1.00
dh.p2.3.h	**65**	20	0.09	0.69	20	0.11	0.69	20	0.00	0.69
dh.p2.3.i	**0**	**0**	0.09	0.00	**0**	0.11	0.00	**0**	0.01	0.00
dh.p2.3.j	**20**	**20**	0.12	0.00	**20**	0.11	0.00	**20**	0.01	0.00
dh.p2.3.k	**10**	0	0.09	1.00	0	0.09	1.00	0	0.01	1.00
dh.p2.4.a	**0**	**0**	0.09	0.00	**0**	0.09	0.00	**0**	0.00	0.00
dh.p2.4.b	**0**	**0**	0.10	0.00	**0**	0.09	0.00	**0**	0.00	0.00
dh.p2.4.c	**10**	**10**	0.09	0.00	**10**	0.09	0.00	**10**	0.01	0.00
dh.p2.4.d	**0**	**0**	0.11	0.00	**0**	0.11	0.00	**0**	0.00	0.00
dh.p2.4.e	**0**	**0**	0.09	0.00	**0**	0.09	0.00	**0**	0.01	0.00
dh.p2.4.f	**25**	0	0.09	1.00	0	0.10	1.00	0	0.01	1.00
dh.p2.4.g	**25**	0	0.14	1.00	0	0.10	1.00	0	0.01	1.00
dh.p2.4.h	**0**	**0**	0.09	0.00	**0**	0.10	0.00	**0**	0.00	0.00
dh.p2.4.i	**0**	**0**	0.10	0.00	**0**	0.12	0.00	**0**	0.00	0.00
dh.p2.4.j	**45**	10	0.09	0.78	10	0.12	0.78	10	0.01	0.78
dh.p2.4.k	**25**	0	0.09	1.00	0	0.12	1.00	0	0.00	1.00

Table 4.13: Instance p3

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dw.p3.2.a	**40**	**40**	0.90	0.00	**40**	0.90	0.00	**40**	0.10	0.00
dw.p3.2.b	**130**	80	1.95	0.38	80	1.95	0.38	80	0.11	0.38
dw.p3.2.c	**160**	130	1.41	0.19	110	1.39	0.31	110	0.03	0.31
dw.p3.2.d	**200**	160	1.86	0.20	160	1.74	0.20	120	0.02	0.40
dw.p3.2.e	**250**	220	5.45	0.12	220	5.16	0.12	140	0.12	0.44
dw.p3.2.f	**290**	230	4.26	0.21	230	4.09	0.21	160	0.05	0.45
dw.p3.2.g	**340**	310	9.27	0.09	290	9.49	0.15	290	0.16	0.15
dw.p3.2.h	**390**	360	8.60	0.08	340	8.69	0.13	290	0.12	0.26
dw.p3.2.i	**430**	360	6.08	0.16	360	6.41	0.16	280	0.07	0.35
dw.p3.2.j	**480**	340	7.29	0.29	340	7.31	0.29	310	0.06	0.35
dw.p3.2.k	**530**	430	11.79	0.19	410	9.70	0.23	290	0.09	0.45
dw.p3.2.l	**500**	440	9.77	0.12	440	10.88	0.12	350	0.10	0.30
dw.p3.2.m	**610**	550	10.74	0.10	540	10.74	0.11	420	0.06	0.31
dw.p3.2.n	**620**	510	14.74	0.18	510	15.12	0.18	430	0.07	0.31
dw.p3.2.o	**660**	550	15.34	0.17	560	12.80	0.15	430	0.06	0.35
dw.p3.2.p	**700**	590	14.95	0.16	590	18.67	0.16	570	0.13	0.19
dw.p3.2.q	**710**	620	11.05	0.13	630	12.27	0.11	530	0.12	0.25
dw.p3.2.r	**710**	610	12.17	0.14	620	12.98	0.13	480	0.07	0.32
dw.p3.2.s	**700**	650	10.64	0.07	650	10.62	0.07	620	0.08	0.11
dw.p3.2.t	**790**	740	13.52	0.06	730	13.18	0.08	540	0.03	0.32
dw.p3.3.a	**30**	**30**	1.36	0.00	**30**	1.68	0.00	**30**	0.05	0.00
dw.p3.3.b	**40**	30	1.46	0.25	30	1.79	0.25	30	0.05	0.25
dw.p3.3.c	**120**	60	1.08	0.50	60	1.23	0.50	60	0.05	0.50
dw.p3.3.d	**170**	110	2.42	0.35	110	2.43	0.35	100	0.05	0.41
dw.p3.3.e	**210**	140	1.90	0.33	140	2.01	0.33	120	0.06	0.43
dw.p3.3.f	**230**	180	6.18	0.22	180	6.80	0.22	160	0.15	0.30
dw.p3.3.g	**270**	230	6.60	0.15	230	6.87	0.15	230	0.12	0.15
dw.p3.3.h	**300**	290	6.32	0.03	270	6.14	0.10	280	0.11	0.07
dw.p3.3.i	**330**	280	10.30	0.15	300	11.35	0.09	240	0.12	0.27
dw.p3.3.j	**380**	360	8.97	0.05	350	9.93	0.08	330	0.13	0.13
dw.p3.3.k	**420**	360	8.76	0.14	360	9.48	0.14	270	0.06	0.36
dw.p3.3.l	**450**	370	7.33	0.18	370	7.78	0.18	300	0.08	0.33
dw.p3.3.m	**490**	370	11.38	0.24	370	12.12	0.24	340	0.13	0.31
dw.p3.3.n	**560**	430	12.28	0.23	430	13.64	0.23	450	0.09	0.20
dw.p3.3.o	**560**	380	8.53	0.32	390	8.91	0.30	300	0.07	0.46
dw.p3.3.p	**630**	550	18.21	0.13	540	19.01	0.14	540	0.12	0.14
dw.p3.3.q	**630**	480	10.80	0.24	490	8.89	0.22	490	0.05	0.22
dw.p3.3.r	**680**	580	12.36	0.15	580	13.90	0.15	440	0.06	0.35
dw.p3.3.s	**680**	480	12.35	0.29	470	13.28	0.31	470	0.02	0.31
dw.p3.3.t	**710**	560	10.80	0.21	580	10.76	0.18	580	0.11	0.18
dw.p3.4.a	**20**	**20**	1.34	0.00	**20**	1.26	0.00	**20**	0.05	0.00
dw.p3.4.b	**30**	**30**	1.47	0.00	**30**	1.42	0.00	**30**	0.03	0.00
dw.p3.4.c	**90**	30	1.12	0.67	30	1.30	0.67	30	0.04	0.67
dw.p3.4.d	**90**	80	1.67	0.11	80	2.00	0.11	80	0.04	0.11
dw.p3.4.e	**130**	120	2.97	0.08	120	3.27	0.08	120	0.08	0.08
dw.p3.4.f	**190**	160	3.98	0.16	160	4.62	0.16	160	0.10	0.16
dw.p3.4.g	**190**	160	6.10	0.16	160	6.67	0.16	160	0.16	0.16
dw.p3.4.h	**240**	210	5.83	0.13	210	6.30	0.13	160	0.10	0.33
dw.p3.4.i	**250**	220	6.12	0.12	230	6.68	0.08	210	0.13	0.16
dw.p3.4.j	**300**	230	5.30	0.23	230	4.93	0.23	230	0.05	0.23
dw.p3.4.k	**350**	280	8.44	0.20	290	8.50	0.17	260	0.11	0.26
dw.p3.4.l	**370**	290	5.98	0.22	290	5.64	0.22	280	0.07	0.24
dw.p3.4.m	**370**	270	4.65	0.27	270	4.77	0.27	220	0.08	0.41
dw.p3.4.n	**430**	340	8.88	0.21	310	9.99	0.28	290	0.09	0.33
dw.p3.4.o	**480**	370	5.48	0.23	370	5.64	0.23	320	0.02	0.33
dw.p3.4.p	**540**	450	12.87	0.17	450	12.58	0.17	390	0.07	0.28
dw.p3.4.q	**540**	450	12.82	0.17	460	11.25	0.15	410	0.11	0.24
dw.p3.4.r	**520**	460	9.27	0.12	460	9.19	0.12	340	0.06	0.35
dw.p3.4.s	**620**	540	16.06	0.13	560	16.05	0.10	530	0.14	0.15
dw.p3.4.t	**660**	580	13.98	0.12	620	12.32	0.06	530	0.10	0.20

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dm.p3.2.a	**70**	10	0.29	0.86	10	0.29	0.86	10	0.03	0.86
dm.p3.2.b	**120**	60	0.37	0.50	60	0.40	0.50	60	0.01	0.50
dm.p3.2.c	**160**	70	0.52	0.56	70	0.40	0.56	70	0.01	0.56
dm.p3.2.d	**150**	80	0.35	0.47	80	0.30	0.47	80	0.01	0.47
dm.p3.2.e	**190**	120	0.55	0.37	120	0.45	0.37	120	0.01	0.37
dm.p3.2.f	**230**	80	0.40	0.65	120	0.40	0.48	80	0.01	0.65
dm.p3.2.g	**290**	170	0.46	0.41	180	0.52	0.38	130	0.01	0.55
dm.p3.2.h	**280**	180	0.68	0.36	180	0.60	0.36	180	0.01	0.36
dm.p3.2.i	**370**	210	0.83	0.43	210	0.79	0.43	240	0.02	0.35
dm.p3.2.j	**420**	230	0.68	0.45	230	0.56	0.45	230	0.02	0.45
dm.p3.2.k	**330**	190	1.27	0.42	190	0.95	0.42	190	0.02	0.42
dm.p3.2.l	**370**	250	1.47	0.32	250	0.99	0.32	240	0.01	0.35
dm.p3.2.m	**410**	280	1.66	0.32	280	1.15	0.32	250	0.01	0.39
dm.p3.2.n	**380**	230	0.88	0.39	270	1.52	0.29	210	0.02	0.45
dm.p3.2.o	**430**	230	0.36	0.47	280	0.86	0.35	230	0.02	0.47
dm.p3.2.p	**500**	310	1.11	0.38	370	1.77	0.26	310	0.01	0.38
dm.p3.2.q	**480**	380	1.54	0.21	380	1.85	0.21	280	0.01	0.42
dm.p3.2.r	**500**	360	1.36	0.28	360	1.02	0.28	350	0.02	0.30
dm.p3.2.s	**590**	490	3.54	0.17	490	3.51	0.17	470	0.01	0.20
dm.p3.2.t	**620**	530	5.12	0.15	530	4.96	0.15	490	0.02	0.21
dm.p3.3.a	**10**	**10**	0.45	0.00	**10**	0.44	0.00	**10**	0.02	0.00
dm.p3.3.b	**70**	10	0.16	0.86	10	0.14	0.86	10	0.01	0.86
dm.p3.3.c	**80**	30	0.36	0.63	30	0.22	0.63	30	0.01	0.63
dm.p3.3.d	**110**	60	0.51	0.45	60	0.52	0.45	60	0.02	0.45
dm.p3.3.e	**90**	70	0.54	0.22	70	0.50	0.22	70	0.02	0.22
dm.p3.3.f	**190**	140	0.71	0.26	140	0.54	0.26	140	0.01	0.26
dm.p3.3.g	**200**	140	0.73	0.30	140	0.72	0.30	140	0.02	0.30
dm.p3.3.h	**230**	130	0.84	0.43	130	0.63	0.43	130	0.02	0.43
dm.p3.3.i	**250**	150	0.70	0.40	150	0.74	0.40	150	0.02	0.40
dm.p3.3.j	**270**	210	0.92	0.22	210	0.83	0.22	210	0.02	0.22
dm.p3.3.k	**310**	220	1.02	0.29	220	1.27	0.29	210	0.01	0.32
dm.p3.3.l	**380**	240	1.34	0.37	200	0.88	0.47	200	0.02	0.47
dm.p3.3.m	**390**	180	0.52	0.54	180	0.45	0.54	180	0.02	0.54
dm.p3.3.n	**390**	160	0.61	0.59	160	0.48	0.59	150	0.02	0.62
dm.p3.3.o	**360**	210	1.68	0.42	210	1.62	0.42	190	0.01	0.47
dm.p3.3.p	**480**	280	3.66	0.42	280	2.83	0.42	260	0.02	0.46
dm.p3.3.q	**500**	260	1.23	0.48	260	1.08	0.48	260	0.03	0.48
dm.p3.3.r	**510**	290	2.04	0.43	290	1.85	0.43	250	0.01	0.51
dm.p3.3.s	**510**	360	1.89	0.29	360	1.85	0.29	360	0.02	0.29
dm.p3.3.t	**570**	430	6.27	0.25	430	5.75	0.25	370	0.01	0.35
dm.p3.4.a	**20**	**20**	0.16	0.00	**20**	0.16	0.00	**20**	0.01	0.00
dm.p3.4.b	**20**	10	0.24	0.50	10	0.17	0.50	10	0.01	0.50
dm.p3.4.c	**30**	10	0.19	0.67	10	0.18	0.67	10	0.01	0.67
dm.p3.4.d	**70**	**70**	0.23	0.00	70	0.23	0.00	**70**	0.01	0.00
dm.p3.4.e	**90**	70	0.62	0.22	70	0.63	0.22	70	0.02	0.22
dm.p3.4.f	**140**	100	0.55	0.29	100	0.52	0.29	100	0.02	0.29
dm.p3.4.g	**100**	60	0.58	0.40	60	0.58	0.40	60	0.02	0.40
dm.p3.4.h	**130**	70	0.82	0.46	70	0.83	0.46	70	0.02	0.46
dm.p3.4.i	**220**	120	0.63	0.45	120	0.73	0.45	120	0.02	0.45
dm.p3.4.j	**200**	160	0.89	0.20	160	0.83	0.20	160	0.02	0.20
dm.p3.4.k	**260**	90	0.76	0.65	90	0.60	0.65	90	0.02	0.65
dm.p3.4.l	**260**	160	1.03	0.38	160	0.82	0.38	150	0.02	0.42
dm.p3.4.m	**240**	130	1.90	0.46	130	1.49	0.46	130	0.02	0.46
dm.p3.4.n	**270**	130	1.07	0.52	130	1.05	0.52	120	0.02	0.56
dm.p3.4.o	**360**	200	1.04	0.44	200	0.91	0.44	200	0.02	0.44
dm.p3.4.p	**340**	200	1.88	0.41	200	1.85	0.41	200	0.02	0.41
dm.p3.4.q	**430**	200	1.08	0.53	190	0.86	0.56	190	0.02	0.56
dm.p3.4.r	**460**	300	1.39	0.35	300	1.09	0.35	300	0.02	0.35
dm.p3.4.s	**420**	250	1.71	0.40	240	1.53	0.43	240	0.02	0.43
dm.p3.4.t	**460**	230	1.09	0.50	240	1.12	0.48	220	0.03	0.52

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dh.p3.2.a	**10**	**10**	0.10	0.00	**10**	0.10	0.00	**10**	0.01	0.00
dh.p3.2.b	**80**	30	0.14	0.63	30	0.14	0.63	30	0.01	0.63
dh.p3.2.c	**50**	40	0.22	0.20	40	0.20	0.20	40	0.01	0.20
dh.p3.2.d	**60**	0	0.10	1.00	0	0.10	1.00	0	0.00	1.00
dh.p3.2.e	**70**	20	0.14	0.71	20	0.15	0.71	20	0.01	0.71
dh.p3.2.f	**100**	30	0.15	0.70	30	0.16	0.70	30	0.01	0.70
dh.p3.2.g	**90**	40	0.17	0.56	40	0.18	0.56	40	0.01	0.56
dh.p3.2.h	**140**	90	0.25	0.36	90	0.26	0.36	90	0.01	0.36
dh.p3.2.i	**150**	60	0.18	0.60	60	0.25	0.60	60	0.01	0.60
dh.p3.2.j	**140**	70	0.27	0.50	70	0.28	0.50	70	0.01	0.50
dh.p3.2.k	**220**	110	0.25	0.50	110	0.27	0.50	110	0.01	0.50
dh.p3.2.l	**160**	70	0.32	0.56	70	0.32	0.56	70	0.01	0.56
dh.p3.2.m	**200**	150	0.41	0.25	150	0.29	0.25	150	0.01	0.25
dh.p3.2.n	**180**	150	0.38	0.17	150	0.33	0.17	150	0.01	0.17
dh.p3.2.o	**200**	90	0.24	0.55	90	0.19	0.55	90	0.01	0.55
dh.p3.2.p	**330**	190	0.21	0.42	190	0.20	0.42	170	0.01	0.48
dh.p3.2.q	**260**	180	0.29	0.31	180	0.33	0.31	180	0.01	0.31
dh.p3.2.r	**270**	120	0.32	0.56	120	0.30	0.56	120	0.00	0.56
dh.p3.2.s	**250**	150	0.34	0.40	150	0.32	0.40	150	0.01	0.40
dh.p3.2.t	**240**	110	0.30	0.54	110	0.27	0.54	110	0.00	0.54
dh.p3.3.a	**0**	**0**	0.11	0.00	**0**	0.11	0.00	**0**	0.00	0.00
dh.p3.3.b	**0**	**0**	0.11	0.00	**0**	0.10	0.00	**0**	0.01	0.00
dh.p3.3.c	**10**	**10**	0.11	0.00	**10**	0.11	0.00	**10**	0.01	0.00
dh.p3.3.d	**20**	**20**	0.11	0.00	**20**	0.12	0.00	**20**	0.01	0.00
dh.p3.3.e	**10**	**10**	0.27	0.00	**10**	0.28	0.00	**10**	0.02	0.00
dh.p3.3.f	**70**	30	0.18	0.57	30	0.18	0.57	20	0.01	0.71
dh.p3.3.g	**100**	20	0.18	0.80	20	0.17	0.80	20	0.01	0.80
dh.p3.3.h	**40**	20	0.21	0.50	20	0.22	0.50	20	0.01	0.50
dh.p3.3.i	**90**	30	0.18	0.67	30	0.19	0.67	30	0.01	0.67
dh.p3.3.j	**130**	30	0.18	0.77	30	0.21	0.77	30	0.01	0.77
dh.p3.3.k	**110**	30	0.33	0.73	30	0.20	0.73	30	0.01	0.73
dh.p3.3.l	**100**	30	0.21	0.70	30	0.20	0.70	30	0.01	0.70
dh.p3.3.m	**150**	90	0.24	0.40	90	0.22	0.40	80	0.01	0.47
dh.p3.3.n	**150**	50	0.38	0.67	50	0.31	0.67	50	0.01	0.67
dh.p3.3.o	**150**	90	0.27	0.40	90	0.24	0.40	90	0.01	0.40
dh.p3.3.p	**160**	50	0.33	0.69	50	0.32	0.69	50	0.01	0.69
dh.p3.3.q	**200**	100	0.38	0.50	100	0.28	0.50	100	0.01	0.50
dh.p3.3.r	**270**	110	0.38	0.59	110	0.33	0.59	110	0.01	0.59
dh.p3.3.s	**210**	90	0.55	0.57	90	0.43	0.57	90	0.01	0.57
dh.p3.3.t	**210**	70	0.46	0.67	70	0.36	0.67	70	0.01	0.67
dh.p3.4.a	**0**	**0**	0.11	0.00	**0**	0.10	0.00	**0**	0.01	0.00
dh.p3.4.b	**10**	**10**	0.19	0.00	**10**	0.18	0.00	**10**	0.01	0.00
dh.p3.4.c	**20**	0	0.13	1.00	0	0.12	1.00	0	0.01	1.00
dh.p3.4.d	**0**	**0**	0.13	0.00	**0**	0.12	0.00	**0**	0.01	0.00
dh.p3.4.e	**0**	**0**	0.13	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p3.4.f	**30**	20	0.26	0.33	20	0.24	0.33	20	0.01	0.33
dh.p3.4.g	**10**	0	0.14	1.00	0	0.13	1.00	0	0.01	1.00
dh.p3.4.h	**100**	30	0.25	0.70	30	0.27	0.70	30	0.02	0.70
dh.p3.4.i	**50**	**50**	0.39	0.00	**50**	0.30	0.00	**50**	0.01	0.00
dh.p3.4.j	**70**	30	0.25	0.57	30	0.27	0.57	30	0.01	0.57
dh.p3.4.k	**80**	30	0.26	0.63	30	0.25	0.63	30	0.02	0.63
dh.p3.4.l	**90**	40	0.41	0.56	40	0.41	0.56	40	0.02	0.56
dh.p3.4.m	**80**	30	0.24	0.63	30	0.23	0.63	30	0.01	0.63
dh.p3.4.n	**110**	50	0.42	0.55	50	0.42	0.55	50	0.01	0.55
dh.p3.4.o	**110**	50	0.34	0.55	50	0.37	0.55	50	0.01	0.55
dh.p3.4.p	**190**	100	0.52	0.47	100	0.50	0.47	90	0.02	0.53
dh.p3.4.q	**160**	90	0.35	0.44	90	0.34	0.44	80	0.01	0.50
dh.p3.4.r	**160**	80	0.32	0.50	80	0.35	0.50	80	0.02	0.50
dh.p3.4.s	**190**	90	0.49	0.53	90	0.51	0.53	90	0.02	0.53
dh.p3.4.t	**170**	40	0.27	0.76	40	0.28	0.76	40	0.02	0.76

Table 4.14: Instance p4

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dw.p4.2.a	**143**	125	20.01	12.6%	125	20.21	12.6%	111	0.37	22.4%
dw.p4.2.b	**281**	234	81.50	16.7%	230	83.67	18.1%	202	0.46	28.1%
dw.p4.2.c	**380**	351	117.36	7.6%	351	124.64	7.6%	250	0.39	34.2%
dw.p4.2.d	471	433	225.10	8.1%	405	187.43	14.0%	265	0.64	43.7%
dw.p4.2.e	521	455	211.94	12.7%	468	255.50	10.2%	323	0.60	38.0%
dw.p4.2.f	613	521	313.82	15.0%	542	323.24	11.6%	358	0.69	41.6%
dw.p4.2.g	642	630	406.31	1.9%	622	389.21	3.1%	495	0.91	22.9%
dw.p4.2.h	685	672	595.60	1.9%	637	477.49	7.0%	447	1.65	34.7%
dw.p4.2.i	726	725	660.75	0.1%	707	632.49	2.6%	525	1.65	27.7%
dw.p4.2.j	781	804	707.34	-2.9%	804	666.07	-2.9%	593	1.94	24.1%
dw.p4.2.k	830	815	851.20	1.8%	778	682.66	6.3%	557	1.95	32.9%
dw.p4.2.l	936	908	722.95	3.0%	861	689.40	8.0%	686	1.04	26.7%
dw.p4.2.m	927	892	1138.28	3.8%	900	1057.45	2.9%	601	1.28	35.2%
dw.p4.2.n	992	971	960.20	2.1%	934	944.64	5.8%	726	1.45	26.8%
dw.p4.2.o	1063	1024	1722.06	3.7%	1054	1599.59	0.8%	837	5.94	21.3%
dw.p4.2.p	1018	1053	1544.56	-3.4%	996	1282.74	2.2%	784	2.67	23.0%
dw.p4.2.q	1008	1026	1216.95	-1.8%	1004	1182.69	0.4%	831	1.67	17.6%
dw.p4.2.r	1077	1013	1450.38	5.9%	979	1433.91	9.1%	707	0.74	34.4%
dw.p4.2.s	1220	1089	1928.87	10.7%	1081	1822.33	11.4%	825	2.66	32.4%
dw.p4.2.t	1192	1133	1583.53	4.9%	1118	1453.72	6.2%	863	1.19	27.6%
dw.p4.3.a	**0**	**0**	0.98	0.0%	**0**	0.82	0.0%	**0**	0.04	0.0%
dw.p4.3.b	**1**	0	0.64	100.0%	0	0.48	100.0%	0	0.02	100.0%
dw.p4.3.c	**135**	81	13.34	40.0%	81	14.34	40.0%	58	0.26	57.0%
dw.p4.3.d	**278**	241	57.26	13.3%	241	56.20	13.3%	206	0.46	25.9%
dw.p4.3.e	**408**	373	105.24	8.6%	362	113.66	11.3%	291	0.76	28.7%
dw.p4.3.f	**485**	400	178.49	17.5%	379	202.69	21.9%	345	0.32	28.9%
dw.p4.3.g	549	389	138.28	29.1%	470	305.64	14.4%	314	0.42	42.8%
dw.p4.3.h	672	592	388.63	11.9%	645	409.02	4.0%	424	0.93	36.9%
dw.p4.3.i	628	587	337.52	6.5%	594	379.21	5.4%	517	0.88	17.7%
dw.p4.3.j	731	714	480.54	2.3%	746	537.09	-2.1%	508	0.71	30.5%
dw.p4.3.k	767	731	671.41	4.7%	698	680.04	9.0%	523	0.71	31.8%
dw.p4.3.l	799	740	517.33	7.4%	738	538.47	7.6%	513	0.65	35.8%
dw.p4.3.m	888	898	852.23	-1.1%	888	859.93	0.0%	746	1.51	16.0%
dw.p4.3.n	885	855	746.07	3.4%	842	716.63	4.9%	610	0.70	31.1%
dw.p4.3.o	939	898	965.51	4.4%	909	961.71	3.2%	718	1.16	23.5%
dw.p4.3.p	972	992	1126.70	-2.1%	1029	1051.63	-5.9%	785	0.99	19.2%
dw.p4.3.q	1076	1016	954.47	5.6%	998	910.99	7.2%	789	0.80	26.7%
dw.p4.3.r	1075	1054	939.69	2.0%	1036	821.97	3.6%	863	0.61	19.7%
dw.p4.3.s	1215	1152	1265.74	5.2%	1105	1174.63	9.1%	914	1.15	24.8%
dw.p4.3.t	1116	1038	1204.47	7.0%	1008	1173.49	9.7%	933	0.69	16.4%
dw.p4.4.a	**0**	**0**	0.86	0.0%	**0**	0.75	0.0%	**0**	0.03	0.0%
dw.p4.4.b	**0**	**0**	0.61	0.0%	**0**	0.64	0.0%	**0**	0.02	0.0%
dw.p4.4.c	**0**	**0**	0.69	0.0%	**0**	0.70	0.0%	**0**	0.03	0.0%
dw.p4.4.d	**12**	0	0.67	100.0%	0	0.70	100.0%	0	0.03	100.0%
dw.p4.4.e	**183**	145	26.70	20.8%	145	27.49	20.8%	161	0.34	12.0%
dw.p4.4.f	**262**	194	46.59	26.0%	197	49.26	24.8%	157	0.30	40.1%
dw.p4.4.g	**409**	328	203.15	19.8%	328	209.94	19.8%	298	0.67	27.1%
dw.p4.4.h	**526**	447	197.40	15.0%	447	208.38	15.0%	412	0.60	21.7%
dw.p4.4.i	595	413	224.77	30.6%	422	264.32	29.1%	334	0.68	43.9%
dw.p4.4.j	623	584	392.37	6.3%	573	384.99	8.0%	437	0.65	29.9%
dw.p4.4.k	702	605	545.01	13.8%	630	486.60	10.3%	480	1.56	31.6%
dw.p4.4.l	681	657	535.16	3.5%	673	487.47	1.2%	514	0.64	24.5%
dw.p4.4.m	780	713	574.14	8.6%	715	577.73	8.3%	584	0.88	25.1%
dw.p4.4.n	856	704	642.39	17.8%	691	611.89	19.3%	598	1.35	30.1%
dw.p4.4.o	977	867	835.19	11.3%	818	908.81	16.3%	675	1.74	30.9%
dw.p4.4.p	944	832	851.85	11.9%	876	806.23	7.2%	706	0.69	25.2%
dw.p4.4.q	995	907	865.89	8.8%	908	846.34	8.7%	729	0.69	26.7%
dw.p4.4.r	969	936	949.73	3.4%	909	939.95	6.2%	760	1.47	21.6%
dw.p4.4.s	1064	1002	993.31	5.8%	967	965.29	9.1%	933	1.23	12.3%
dw.p4.4.t	1084	995	815.76	8.2%	987	733.96	8.9%	876	0.77	19.2%

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dm.p4.2.a	**146**	70	1.73	0.52	70	1.71	0.52	70	0.03	0.52
dm.p4.2.b	**215**	111	1.61	0.48	111	1.58	0.48	111	0.05	0.48
dm.p4.2.c	**273**	171	4.09	0.37	169	4.00	0.38	199	0.06	0.27
dm.p4.2.d	**335**	272	15.47	0.19	272	15.66	0.19	241	0.06	0.28
dm.p4.2.e	**399**	304	27.84	0.24	304	26.92	0.24	185	0.11	0.54
dm.p4.2.f	**480**	340	37.39	0.29	340	41.35	0.29	268	0.07	0.44
dm.p4.2.g	502	398	58.84	0.21	398	58.65	0.21	338	0.07	0.33
dm.p4.2.h	545	431	74.36	0.21	431	70.29	0.21	272	0.07	0.50
dm.p4.2.i	604	429	63.81	0.29	443	82.91	0.27	294	0.06	0.51
dm.p4.2.j	600	527	80.65	0.12	527	76.53	0.12	395	0.08	0.34
dm.p4.2.k	645	545	131.76	0.16	553	128.73	0.14	386	0.07	0.40
dm.p4.2.l	589	490	84.89	0.17	491	85.47	0.17	343	0.07	0.42
dm.p4.2.m	680	558	81.50	0.18	554	81.42	0.19	412	0.08	0.39
dm.p4.2.n	808	673	121.04	0.17	673	113.10	0.17	517	0.08	0.36
dm.p4.2.o	801	728	181.29	0.09	701	192.26	0.12	573	0.17	0.28
dm.p4.2.p	825	771	164.91	0.07	749	147.13	0.09	519	0.16	0.37
dm.p4.2.q	826	731	182.71	0.12	742	177.36	0.10	544	0.09	0.34
dm.p4.2.r	807	683	109.55	0.15	626	105.16	0.22	494	0.10	0.39
dm.p4.2.s	860	756	145.71	0.12	752	143.83	0.13	632	0.11	0.27
dm.p4.2.t	869	788	197.54	0.09	758	181.16	0.13	539	0.10	0.38
dm.p4.3.a	**0**	**0**	0.38	0.00	**0**	0.43	0.00	**0**	0.01	0.00
dm.p4.3.b	**12**	11	1.57	0.08	11	1.75	0.08	11	0.06	0.08
dm.p4.3.c	**158**	106	3.30	0.33	106	3.65	0.33	106	0.07	0.33
dm.p4.3.d	**236**	143	2.07	0.39	143	2.26	0.39	113	0.06	0.52
dm.p4.3.e	**285**	218	5.12	0.24	218	5.49	0.24	174	0.07	0.39
dm.p4.3.f	**330**	214	12.13	0.35	214	12.57	0.35	202	0.08	0.39
dm.p4.3.g	**475**	343	17.08	0.28	343	19.20	0.28	296	0.06	0.38
dm.p4.3.h	**441**	326	34.09	0.26	326	32.51	0.26	269	0.07	0.39
dm.p4.3.i	**488**	358	45.76	0.27	340	45.56	0.30	290	0.07	0.41
dm.p4.3.j	568	436	59.47	0.23	436	51.12	0.23	375	0.08	0.34
dm.p4.3.k	573	439	83.92	0.23	438	83.09	0.24	403	0.10	0.30
dm.p4.3.l	638	440	80.95	0.31	461	89.03	0.28	327	0.08	0.49
dm.p4.3.m	671	488	65.77	0.27	443	37.48	0.34	480	0.07	0.28
dm.p4.3.n	702	508	97.09	0.28	476	107.26	0.32	494	0.08	0.30
dm.p4.3.o	747	526	110.27	0.30	585	139.77	0.22	443	0.09	0.41
dm.p4.3.p	745	716	146.24	0.04	717	149.88	0.04	522	0.07	0.30
dm.p4.3.q	807	681	148.42	0.16	691	176.52	0.14	543	0.11	0.33
dm.p4.3.r	749	637	116.56	0.15	637	119.13	0.15	519	0.09	0.31
dm.p4.3.s	796	601	137.72	0.24	648	127.96	0.19	572	0.12	0.28
dm.p4.3.t	854	642	126.72	0.25	639	128.46	0.25	480	0.09	0.44
dm.p4.4.a	**0**	**0**	0.45	0.00	**0**	0.44	0.00	**0**	0.02	0.00
dm.p4.4.b	**0**	**0**	0.31	0.00	**0**	0.32	0.00	**0**	0.02	0.00
dm.p4.4.c	**0**	**0**	0.30	0.00	**0**	0.31	0.00	**0**	0.01	0.00
dm.p4.4.d	**12**	11	1.94	0.08	11	2.02	0.08	11	0.03	0.08
dm.p4.4.e	**106**	68	2.70	0.36	68	2.84	0.36	68	0.05	0.36
dm.p4.4.f	**179**	57	3.03	0.68	57	3.22	0.68	56	0.06	0.69
dm.p4.4.g	**232**	140	2.80	0.40	140	3.00	0.40	140	0.08	0.40
dm.p4.4.h	**385**	248	6.39	0.36	248	6.50	0.36	248	0.08	0.36
dm.p4.4.i	**429**	263	14.49	0.39	261	13.04	0.39	263	0.07	0.39
dm.p4.4.j	**382**	295	32.76	0.23	292	41.76	0.24	283	0.09	0.26
dm.p4.4.k	**548**	381	50.36	0.30	413	53.95	0.25	380	0.12	0.31
dm.p4.4.l	**558**	387	36.63	0.31	387	41.16	0.31	288	0.09	0.48
dm.p4.4.m	**561**	359	34.07	0.36	343	33.68	0.39	310	0.07	0.45
dm.p4.4.n	636	462	94.82	0.27	420	85.57	0.34	317	0.09	0.50
dm.p4.4.o	671	411	45.95	0.39	441	69.19	0.34	378	0.09	0.44
dm.p4.4.p	714	533	84.87	0.25	526	82.16	0.26	508	0.08	0.29
dm.p4.4.q	709	523	116.56	0.26	523	124.33	0.26	423	0.10	0.40
dm.p4.4.r	723	555	150.67	0.23	543	152.71	0.25	507	0.09	0.30
dm.p4.4.s	723	543	80.96	0.25	559	89.18	0.23	464	0.09	0.36
dm.p4.4.t	820	675	195.53	0.18	655	177.45	0.20	602	0.14	0.27

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dh.p4.2.a	**84**	28	0.64	0.67	28	0.69	0.67	28	0.02	0.67
dh.p4.2.b	**127**	86	0.91	0.32	86	0.92	0.32	86	0.03	0.32
dh.p4.2.c	**156**	101	0.57	0.35	101	0.57	0.35	101	0.02	0.35
dh.p4.2.d	**180**	120	0.68	0.33	120	0.69	0.33	120	0.01	0.33
dh.p4.2.e	**206**	87	0.77	0.58	87	0.75	0.58	86	0.02	0.58
dh.p4.2.f	**208**	140	1.77	0.33	140	1.63	0.33	140	0.02	0.33
dh.p4.2.g	**233**	123	1.31	0.47	123	1.27	0.47	123	0.02	0.47
dh.p4.2.h	**258**	146	1.55	0.43	170	1.18	0.34	170	0.02	0.34
dh.p4.2.i	**273**	121	0.87	0.56	121	0.85	0.56	137	0.02	0.50
dh.p4.2.j	**311**	190	1.71	0.39	190	1.69	0.39	190	0.02	0.39
dh.p4.2.k	**293**	208	4.03	0.29	208	3.96	0.29	168	0.02	0.43
dh.p4.2.l	**347**	296	5.81	0.15	296	5.73	0.15	296	0.02	0.15
dh.p4.2.m	**275**	182	2.25	0.34	182	2.23	0.34	182	0.02	0.34
dh.p4.2.n	**353**	219	2.38	0.38	219	2.36	0.38	206	0.02	0.42
dh.p4.2.o	**368**	241	1.86	0.35	241	1.85	0.35	216	0.02	0.41
dh.p4.2.p	**349**	211	6.01	0.40	211	5.97	0.40	211	0.01	0.40
dh.p4.2.q	**381**	186	1.44	0.51	186	1.40	0.51	183	0.02	0.52
dh.p4.2.r	**437**	321	18.73	0.27	321	12.37	0.27	226	0.02	0.48
dh.p4.2.s	**389**	255	6.40	0.34	255	6.23	0.34	254	0.02	0.35
dh.p4.2.t	**418**	254	2.47	0.39	254	2.42	0.39	196	0.02	0.53
dh.p4.3.a	**0**	**0**	0.30	0.00	**0**	0.29	0.00	**0**	0.01	0.00
dh.p4.3.b	**1**	0	0.21	1.00	0	0.18	1.00	0	0.01	1.00
dh.p4.3.c	**45**	0	0.20	1.00	0	0.18	1.00	0	0.01	1.00
dh.p4.3.d	**119**	28	0.65	0.76	28	0.63	0.76	28	0.03	0.76
dh.p4.3.e	**140**	67	1.16	0.52	67	1.12	0.52	66	0.02	0.53
dh.p4.3.f	**123**	89	1.16	0.28	89	1.10	0.28	89	0.02	0.28
dh.p4.3.g	**175**	83	1.91	0.53	83	1.77	0.53	82	0.03	0.53
dh.p4.3.h	**209**	77	1.04	0.63	80	0.93	0.62	80	0.03	0.62
dh.p4.3.i	**198**	90	0.93	0.55	92	1.15	0.54	78	0.04	0.61
dh.p4.3.j	**213**	78	0.90	0.63	61	1.19	0.71	49	0.03	0.77
dh.p4.3.k	**221**	81	1.38	0.63	81	1.34	0.63	52	0.03	0.76
dh.p4.3.l	**195**	84	0.93	0.57	84	0.91	0.57	83	0.03	0.57
dh.p4.3.m	**285**	160	1.83	0.44	181	2.33	0.36	201	0.03	0.29
dh.p4.3.n	**300**	153	2.41	0.49	153	2.63	0.49	153	0.03	0.49
dh.p4.3.o	**342**	225	2.21	0.34	225	2.26	0.34	217	0.02	0.37
dh.p4.3.p	**360**	183	1.57	0.49	213	2.47	0.41	191	0.03	0.47
dh.p4.3.q	**300**	171	2.21	0.43	171	2.29	0.43	173	0.02	0.42
dh.p4.3.r	**341**	163	1.66	0.52	163	1.62	0.52	163	0.02	0.52
dh.p4.3.s	**405**	200	4.15	0.51	200	2.83	0.51	166	0.02	0.59
dh.p4.3.t	**404**	273	5.12	0.32	273	5.16	0.32	273	0.02	0.32
dh.p4.4.a	**0**	**0**	0.24	0.00	**0**	0.23	0.00	**0**	0.01	0.00
dh.p4.4.b	**0**	**0**	0.24	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dh.p4.4.c	**0**	**0**	0.23	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dh.p4.4.d	**1**	0	0.22	1.00	0	0.23	1.00	0	0.01	1.00
dh.p4.4.e	**51**	18	0.62	0.65	18	0.60	0.65	17	0.02	0.67
dh.p4.4.f	**94**	45	0.84	0.52	45	0.87	0.52	45	0.02	0.52
dh.p4.4.g	**127**	46	1.06	0.64	46	1.09	0.64	45	0.02	0.65
dh.p4.4.h	**172**	82	0.57	0.52	82	0.58	0.52	82	0.02	0.52
dh.p4.4.i	**124**	48	0.91	0.61	48	0.92	0.61	47	0.04	0.62
dh.p4.4.j	**192**	92	1.37	0.52	92	1.31	0.52	92	0.03	0.52
dh.p4.4.k	**161**	41	1.49	0.75	41	1.50	0.75	41	0.03	0.75
dh.p4.4.l	**159**	35	0.74	0.78	35	0.75	0.78	35	0.04	0.78
dh.p4.4.m	**234**	96	1.49	0.59	96	1.56	0.59	51	0.04	0.78
dh.p4.4.n	**257**	119	0.84	0.54	119	0.84	0.54	107	0.02	0.58
dh.p4.4.o	**303**	104	1.02	0.66	104	0.99	0.66	104	0.03	0.66
dh.p4.4.p	**342**	147	1.27	0.57	164	1.36	0.52	146	0.03	0.57
dh.p4.4.q	**311**	129	1.23	0.59	129	1.25	0.59	129	0.03	0.59
dh.p4.4.r	**345**	95	1.25	0.72	95	1.25	0.72	95	0.02	0.72
dh.p4.4.s	**348**	158	1.93	0.55	158	1.96	0.55	158	0.03	0.55
dh.p4.4.t	**284**	115	1.82	0.60	99	1.52	0.65	99	0.04	0.65

Table 4.15: Instance p5

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dw.p5.2.a	**0**	**0**	0.50	0.00	**0**	0.52	0.00	**0**	0.07	0.00
dw.p5.2.b	**20**	15	2.04	0.25	15	2.08	0.25	15	0.12	0.25
dw.p5.2.c	**45**	40	2.66	0.11	40	2.60	0.11	40	0.13	0.11
dw.p5.2.d	**80**	70	2.48	0.13	70	2.35	0.13	70	0.11	0.13
dw.p5.2.e	**170**	140	9.12	0.18	135	9.22	0.21	135	0.17	0.21
dw.p5.2.f	**230**	210	19.16	0.09	220	19.26	0.04	205	0.28	0.11
dw.p5.2.g	**310**	235	16.23	0.24	260	15.71	0.16	225	0.19	0.27
dw.p5.2.h	**395**	315	38.56	0.20	320	36.43	0.19	285	0.30	0.28
dw.p5.2.i	**480**	410	48.85	0.15	410	49.20	0.15	350	0.25	0.27
dw.p5.2.j	550	550	50.29	0.00	550	49.90	0.00	455	0.25	0.17
dw.p5.2.k	645	585	67.03	0.09	575	63.72	0.11	505	0.28	0.22
dw.p5.2.l	760	695	112.98	0.09	600	121.26	0.21	600	0.75	0.21
dw.p5.2.m	845	765	113.41	0.09	685	109.16	0.19	635	0.50	0.25
dw.p5.2.n	840	815	120.89	0.03	720	103.36	0.14	685	0.38	0.18
dw.p5.2.o	1005	935	176.26	0.07	905	146.03	0.10	760	0.37	0.24
dw.p5.2.p	1130	875	125.70	0.23	880	107.82	0.22	770	0.27	0.32
dw.p5.2.q	1065	1055	141.36	0.01	1030	116.11	0.03	870	0.29	0.18
dw.p5.2.r	1195	1120	179.64	0.06	1060	189.44	0.11	1015	0.30	0.15
dw.p5.2.s	1285	1130	179.07	0.12	1110	148.51	0.14	990	0.23	0.23
dw.p5.2.t	1290	1145	204.44	0.11	1120	174.87	0.13	1005	0.36	0.22
dw.p5.2.u	1395	1245	244.63	0.11	1210	176.28	0.13	1015	0.38	0.27
dw.p5.2.v	1395	1285	270.52	0.08	1305	235.44	0.06	1175	0.55	0.16
dw.p5.2.w	1485	1370	251.62	0.08	1345	248.64	0.09	1165	0.41	0.22
dw.p5.2.x	1400	1245	218.01	0.11	1185	188.36	0.15	945	0.19	0.33
dw.p5.2.y	1550	1330	202.77	0.14	1395	185.80	0.10	1170	0.29	0.25
dw.p5.2.z	1570	1495	294.64	0.05	1495	282.19	0.05	1350	1.00	0.14
dw.p5.3.a	**0**	**0**	0.34	0.00	**0**	0.42	0.00	**0**	0.02	0.00
dw.p5.3.b	**15**	5	1.98	0.67	5	1.64	0.67	5	0.07	0.67
dw.p5.3.c	**20**	10	2.09	0.50	10	1.95	0.50	10	0.08	0.50
dw.p5.3.d	**60**	45	2.09	0.25	45	2.13	0.25	45	0.12	0.25
dw.p5.3.e	**80**	60	3.29	0.25	60	3.17	0.25	60	0.12	0.25
dw.p5.3.f	**110**	105	9.16	0.05	105	8.54	0.05	95	0.20	0.14
dw.p5.3.g	**185**	140	11.69	0.24	140	11.54	0.24	140	0.14	0.24
dw.p5.3.h	**255**	220	17.75	0.14	210	14.73	0.18	215	0.23	0.16
dw.p5.3.i	**320**	230	20.29	0.28	225	18.67	0.30	225	0.23	0.30
dw.p5.3.j	**460**	390	52.36	0.15	350	44.23	0.24	365	0.38	0.21
dw.p5.3.k	490	430	65.25	0.12	410	65.05	0.16	395	0.33	0.19
dw.p5.3.l	585	490	55.22	0.16	495	64.42	0.15	395	0.18	0.32
dw.p5.3.m	650	545	84.98	0.16	545	81.87	0.16	525	0.43	0.19
dw.p5.3.n	720	655	101.37	0.09	640	90.88	0.11	600	0.25	0.17
dw.p5.3.o	815	780	106.74	0.04	780	94.96	0.04	645	0.18	0.21
dw.p5.3.p	885	795	102.33	0.10	790	95.15	0.11	730	0.22	0.18
dw.p5.3.q	1000	925	135.92	0.08	885	114.12	0.12	760	0.19	0.24
dw.p5.3.r	1010	805	111.43	0.20	805	116.15	0.20	725	0.17	0.28
dw.p5.3.s	1085	1035	162.34	0.05	990	163.05	0.09	840	0.31	0.23
dw.p5.3.t	1200	1030	160.72	0.14	1040	137.57	0.13	880	0.22	0.27
dw.p5.3.u	1305	1165	223.11	0.11	1120	198.27	0.14	1090	0.30	0.16
dw.p5.3.v	1270	1230	215.15	0.03	1215	202.77	0.04	980	0.37	0.23
dw.p5.3.w	1320	1260	199.10	0.05	1240	199.52	0.06	1145	0.36	0.13
dw.p5.3.x	1425	1310	206.67	0.08	1380	205.04	0.03	1085	0.37	0.24
dw.p5.3.y	1460	1330	213.35	0.09	1295	217.43	0.11	1050	0.41	0.28
dw.p5.3.z	1505	1285	220.46	0.15	1335	208.12	0.11	1060	0.26	0.30

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dm.p5.2.a	**0**	**0**	0.28	0.00	**0**	0.28	0.00	**0**	0.01	0.00
dm.p5.2.b	**10**	0	0.19	1.00	0	0.21	1.00	0	0.01	1.00
dm.p5.2.c	**40**	30	0.86	0.25	30	0.89	0.25	30	0.04	0.25
dm.p5.2.d	**70**	45	0.85	0.36	45	0.93	0.36	45	0.02	0.36
dm.p5.2.e	**130**	80	0.87	0.38	80	0.86	0.38	80	0.03	0.38
dm.p5.2.f	**180**	120	1.51	0.33	155	1.93	0.14	155	0.04	0.14
dm.p5.2.g	**255**	135	1.30	0.47	135	1.36	0.47	130	0.04	0.49
dm.p5.2.h	**295**	195	5.79	0.34	245	5.22	0.17	195	0.07	0.34
dm.p5.2.i	**435**	250	4.62	0.43	235	4.43	0.46	235	0.05	0.46
dm.p5.2.j	**455**	365	7.78	0.20	365	8.67	0.20	240	0.07	0.47
dm.p5.2.k	**505**	410	11.41	0.19	410	11.67	0.19	380	0.05	0.25
dm.p5.2.l	**575**	415	8.02	0.28	370	8.10	0.36	385	0.05	0.33
dm.p5.2.m	680	430	13.37	0.37	430	13.50	0.37	455	0.04	0.33
dm.p5.2.n	675	600	20.06	0.11	535	22.04	0.21	435	0.05	0.36
dm.p5.2.o	735	570	23.28	0.22	610	25.00	0.17	495	0.06	0.33
dm.p5.2.p	830	675	25.35	0.19	645	26.60	0.22	660	0.05	0.20
dm.p5.2.q	745	630	26.24	0.15	605	23.87	0.19	430	0.05	0.42
dm.p5.2.r	870	675	23.55	0.22	665	24.06	0.24	590	0.05	0.32
dm.p5.2.s	**955**	735	34.80	0.23	740	34.65	0.23	600	0.05	0.37
dm.p5.2.t	940	795	51.59	0.15	810	43.12	0.14	630	0.06	0.33
dm.p5.2.u	1045	835	50.57	0.20	865	42.90	0.17	610	0.06	0.42
dm.p5.2.v	1060	795	51.22	0.25	780	49.85	0.26	630	0.08	0.41
dm.p5.2.w	1125	935	44.22	0.17	890	42.80	0.21	770	0.07	0.32
dm.p5.2.x	**1060**	845	46.18	0.20	820	46.78	0.23	720	0.08	0.32
dm.p5.2.y	**1210**	970	40.51	0.20	970	41.01	0.20	725	0.08	0.40
dm.p5.2.z	**1130**	880	34.20	0.22	930	28.24	0.18	745	0.06	0.34
dm.p5.3.a	**0**	**0**	0.19	0.00	**0**	0.18	0.00	**0**	0.01	0.00
dm.p5.3.b	**10**	0	0.22	1.00	0	0.23	1.00	0	0.01	1.00
dm.p5.3.c	**10**	0	0.37	1.00	0	0.36	1.00	0	0.01	1.00
dm.p5.3.d	**55**	35	1.20	0.36	35	1.04	0.36	35	0.04	0.36
dm.p5.3.e	**75**	45	2.04	0.40	45	1.78	0.40	45	0.06	0.40
dm.p5.3.f	**95**	65	1.77	0.32	65	1.65	0.32	65	0.06	0.32
dm.p5.3.g	**165**	95	1.44	0.42	100	1.36	0.39	95	0.04	0.42
dm.p5.3.h	**205**	100	1.48	0.51	100	1.44	0.51	100	0.03	0.51
dm.p5.3.i	**220**	170	1.67	0.23	170	1.63	0.23	160	0.05	0.27
dm.p5.3.j	**335**	245	2.48	0.27	200	1.99	0.40	220	0.05	0.34
dm.p5.3.k	**350**	300	8.52	0.14	240	4.11	0.31	205	0.05	0.41
dm.p5.3.l	**405**	335	4.81	0.17	330	5.20	0.19	265	0.06	0.35
dm.p5.3.m	**500**	390	9.26	0.22	390	7.65	0.22	290	0.05	0.42
dm.p5.3.n	560	480	15.43	0.14	460	12.15	0.18	400	0.05	0.29
dm.p5.3.o	705	495	14.17	0.30	490	16.71	0.30	395	0.05	0.44
dm.p5.3.p	675	500	16.45	0.26	445	12.75	0.34	480	0.05	0.29
dm.p5.3.q	825	525	13.20	0.36	525	12.74	0.36	480	0.05	0.42
dm.p5.3.r	765	575	20.26	0.25	615	22.41	0.20	530	0.05	0.31
dm.p5.3.s	860	625	37.85	0.27	625	36.21	0.27	580	0.05	0.33
dm.p5.3.t	980	795	38.23	0.19	795	37.88	0.19	620	0.07	0.37
dm.p5.3.u	955	660	31.56	0.31	645	29.05	0.32	645	0.06	0.32
dm.p5.3.v	1010	725	47.87	0.28	710	42.71	0.30	625	0.08	0.38
dm.p5.3.w	1015	765	60.94	0.25	760	57.56	0.25	620	0.07	0.39
dm.p5.3.x	1040	855	48.08	0.18	850	43.04	0.18	750	0.08	0.28
dm.p5.3.y	1170	890	49.86	0.24	890	48.98	0.24	760	0.10	0.35
dm.p5.3.z	1060	745	39.16	0.30	800	37.41	0.25	605	0.07	0.43

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dh.p5.2.a	**0**	**0**	0.19	0.00	**0**	0.19	0.00	**0**	0.00	0.00
dh.p5.2.b	**10**	5	0.42	0.50	5	0.47	0.50	5	0.01	0.50
dh.p5.2.c	**10**	0	0.09	1.00	0	0.09	1.00	0	0.01	1.00
dh.p5.2.d	**40**	5	0.14	0.88	5	0.12	0.88	5	0.01	0.88
dh.p5.2.e	**45**	10	0.12	0.78	10	0.12	0.78	10	0.00	0.78
dh.p5.2.f	**85**	10	0.10	0.88	10	0.10	0.88	10	0.01	0.88
dh.p5.2.g	**115**	60	0.62	0.48	60	0.59	0.48	60	0.02	0.48
dh.p5.2.h	**170**	90	0.15	0.47	90	0.14	0.47	90	0.01	0.47
dh.p5.2.i	**215**	120	0.25	0.44	120	0.25	0.44	120	0.02	0.44
dh.p5.2.j	**155**	90	0.21	0.42	90	0.21	0.42	90	0.01	0.42
dh.p5.2.k	**255**	135	0.38	0.47	135	0.38	0.47	135	0.02	0.47
dh.p5.2.l	**245**	130	0.35	0.47	130	0.35	0.47	130	0.02	0.47
dh.p5.2.m	**205**	115	0.22	0.44	115	0.22	0.44	115	0.01	0.44
dh.p5.2.n	**290**	195	0.39	0.33	195	0.39	0.33	195	0.01	0.33
dh.p5.2.o	**320**	210	0.76	0.34	210	0.75	0.34	210	0.01	0.34
dh.p5.2.p	**370**	200	0.76	0.46	200	0.36	0.46	200	0.00	0.46
dh.p5.2.q	**345**	185	0.58	0.46	185	0.57	0.46	180	0.02	0.48
dh.p5.2.r	**410**	260	0.87	0.37	250	0.82	0.39	265	0.01	0.35
dh.p5.2.s	**365**	230	0.80	0.37	230	0.80	0.37	230	0.02	0.37
dh.p5.2.t	**395**	260	0.80	0.34	260	0.94	0.34	260	0.02	0.34
dh.p5.2.u	**435**	260	1.78	0.40	255	1.39	0.41	245	0.01	0.44
dh.p5.2.v	**475**	270	1.42	0.43	285	1.33	0.40	255	0.01	0.46
dh.p5.2.w	**415**	275	0.67	0.34	265	0.65	0.36	270	0.00	0.35
dh.p5.2.x	**520**	310	0.81	0.40	295	0.69	0.43	295	0.02	0.43
dh.p5.2.y	**500**	370	0.75	0.26	370	0.82	0.26	370	0.02	0.26
dh.p5.2.z	**455**	295	0.89	0.35	270	0.84	0.41	295	0.00	0.35
dh.p5.3.a	**0**	**0**	0.12	0.00	**0**	0.12	0.00	**0**	0.01	0.00
dh.p5.3.b	**0**	**0**	0.13	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p5.3.c	**5**	0	0.11	1.00	0	0.11	1.00	0	0.01	1.00
dh.p5.3.d	**25**	20	0.37	0.20	20	0.38	0.20	20	0.02	0.20
dh.p5.3.e	**25**	**25**	0.46	0.00	**25**	0.46	0.00	**25**	0.02	0.00
dh.p5.3.f	**55**	35	0.48	0.36	35	0.49	0.36	35	0.02	0.36
dh.p5.3.g	**80**	35	0.32	0.56	35	0.33	0.56	35	0.01	0.56
dh.p5.3.h	**50**	0	0.16	1.00	0	0.16	1.00	0	0.01	1.00
dh.p5.3.i	**95**	20	0.16	0.79	20	0.16	0.79	20	0.01	0.79
dh.p5.3.j	**115**	60	0.68	0.48	60	0.68	0.48	60	0.02	0.48
dh.p5.3.k	**85**	80	0.73	0.06	80	0.72	0.06	80	0.03	0.06
dh.p5.3.l	**140**	45	0.35	0.68	45	0.35	0.68	45	0.01	0.68
dh.p5.3.m	**210**	105	0.71	0.50	105	0.70	0.50	105	0.02	0.50
dh.p5.3.n	**225**	85	0.77	0.62	85	0.72	0.62	85	0.02	0.62
dh.p5.3.o	**295**	90	0.88	0.69	125	0.96	0.58	90	0.02	0.69
dh.p5.3.p	**310**	190	0.57	0.39	190	0.56	0.39	190	0.02	0.39
dh.p5.3.q	**315**	155	0.73	0.51	155	0.72	0.51	155	0.01	0.51
dh.p5.3.r	**280**	190	0.62	0.32	190	0.62	0.32	190	0.02	0.32
dh.p5.3.s	**330**	210	0.89	0.36	210	0.82	0.36	210	0.02	0.36
dh.p5.3.t	**335**	140	0.59	0.58	140	0.57	0.58	135	0.01	0.60
dh.p5.3.u	**330**	215	0.93	0.35	215	0.83	0.35	215	0.01	0.35
dh.p5.3.v	**380**	215	0.52	0.43	215	0.52	0.43	215	0.02	0.43
dh.p5.3.w	**385**	205	0.92	0.47	205	0.84	0.47	205	0.02	0.47
dh.p5.3.x	**395**	230	0.75	0.42	205	0.82	0.48	205	0.03	0.48
dh.p5.3.y	**495**	260	0.61	0.47	260	0.62	0.47	260	0.02	0.47
dh.p5.3.z	**465**	190	1.04	0.59	215	1.17	0.54	190	0.02	0.59

Table 4.16: Instance p5 cont.

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dw.p5.4.a	**0**	**0**	0.51	0.00	**0**	0.70	0.00	**0**	0.02	0.00
dw.p5.4.b	**0**	**0**	0.45	0.00	**0**	0.60	0.00	**0**	0.02	0.00
dw.p5.4.c	**20**	15	3.02	0.25	15	3.06	0.25	15	0.12	0.25
dw.p5.4.d	**20**	**20**	2.60	0.00	**20**	2.33	0.00	**20**	0.09	0.00
dw.p5.4.e	**20**	15	2.56	0.25	15	2.49	0.25	15	0.07	0.25
dw.p5.4.f	**75**	60	3.70	0.20	60	3.47	0.20	60	0.12	0.20
dw.p5.4.g	**120**	80	4.69	0.33	80	4.52	0.33	80	0.15	0.33
dw.p5.4.h	**140**	135	16.05	0.04	135	18.12	0.04	105	0.19	0.25
dw.p5.4.i	**220**	155	6.52	0.30	155	7.15	0.30	155	0.17	0.30
dw.p5.4.j	**320**	245	30.88	0.23	245	29.48	0.23	245	0.22	0.23
dw.p5.4.k	**340**	270	35.05	0.21	260	36.35	0.24	260	0.22	0.24
dw.p5.4.l	**400**	350	43.08	0.13	350	41.30	0.13	315	0.22	0.21
dw.p5.4.m	495	420	39.96	0.15	435	40.10	0.12	380	0.17	0.23
dw.p5.4.n	600	450	63.16	0.25	450	60.24	0.25	450	0.26	0.25
dw.p5.4.o	665	540	79.42	0.19	570	83.97	0.14	470	0.19	0.29
dw.p5.4.p	755	655	109.15	0.13	635	99.14	0.16	570	0.45	0.25
dw.p5.4.q	800	675	85.64	0.16	665	81.89	0.17	600	0.29	0.25
dw.p5.4.r	920	765	120.94	0.17	765	111.84	0.17	620	0.21	0.33
dw.p5.4.s	1005	895	151.12	0.11	900	127.23	0.10	695	0.17	0.31
dw.p5.4.t	1030	1010	153.05	0.02	975	128.84	0.05	825	0.22	0.20
dw.p5.4.u	1150	1115	208.53	0.03	1095	210.90	0.05	1110	1.17	0.03
dw.p5.4.v	1200	995	140.65	0.17	990	130.38	0.18	810	0.23	0.33
dw.p5.4.w	1290	1195	156.66	0.07	1180	133.56	0.09	945	0.34	0.27
dw.p5.4.x	1325	1330	154.64	0.00	1325	139.20	0.00	1205	0.31	0.09
dw.p5.4.y	1365	1210	164.64	0.11	1225	143.83	0.10	885	0.31	0.35
dw.p5.4.z	1390	1350	176.22	0.03	1310	153.96	0.06	1125	0.51	0.19

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dm.p5.4.a	**0**	**0**	0.24	0.00	**0**	0.21	0.00	**0**	0.01	0.00
dm.p5.4.b	**0**	**0**	0.23	0.00	**0**	0.24	0.00	**0**	0.01	0.00
dm.p5.4.c	**20**	0	0.45	1.00	0	0.29	1.00	0	0.01	1.00
dm.p5.4.d	**10**	5	0.35	0.50	5	0.37	0.50	5	0.01	0.50
dm.p5.4.e	**20**	10	0.36	0.50	10	0.33	0.50	10	0.02	0.50
dm.p5.4.f	**75**	40	1.75	0.47	40	1.59	0.47	40	0.06	0.47
dm.p5.4.g	**105**	50	2.54	0.52	50	2.33	0.52	50	0.06	0.52
dm.p5.4.h	**120**	55	2.16	0.54	55	2.26	0.54	55	0.04	0.54
dm.p5.4.i	**185**	80	2.26	0.57	80	2.45	0.57	80	0.04	0.57
dm.p5.4.j	**230**	145	4.44	0.37	120	3.19	0.48	120	0.03	0.48
dm.p5.4.k	**215**	120	2.72	0.44	120	2.20	0.44	120	0.04	0.44
dm.p5.4.l	**355**	170	4.34	0.52	170	3.88	0.52	170	0.05	0.52
dm.p5.4.m	**410**	255	4.06	0.38	235	3.57	0.43	235	0.07	0.43
dm.p5.4.n	**495**	335	10.02	0.32	330	10.28	0.33	325	0.07	0.34
dm.p5.4.o	**495**	400	10.95	0.19	400	10.62	0.19	335	0.07	0.32
dm.p5.4.p	650	440	17.12	0.32	440	16.67	0.32	330	0.06	0.49
dm.p5.4.q	650	425	15.27	0.35	425	28.37	0.35	385	0.05	0.41
dm.p5.4.r	700	590	21.78	0.16	590	20.07	0.16	505	0.06	0.28
dm.p5.4.s	745	565	13.17	0.24	555	12.24	0.26	535	0.06	0.28
dm.p5.4.t	810	545	22.68	0.33	545	23.43	0.33	570	0.06	0.30
dm.p5.4.u	825	610	19.48	0.26	610	19.66	0.26	575	0.08	0.30
dm.p5.4.v	860	580	12.97	0.33	570	12.73	0.34	575	0.06	0.33
dm.p5.4.w	960	665	25.20	0.31	665	24.13	0.31	545	0.07	0.43
dm.p5.4.x	**955**	720	21.08	0.25	680	21.19	0.29	610	0.07	0.36
dm.p5.4.y	1020	745	42.02	0.27	755	33.95	0.26	720	0.07	0.29
dm.p5.4.z	1155	820	51.32	0.29	820	47.47	0.29	770	0.07	0.33

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dh.p5.4.a	**0**	**0**	0.13	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p5.4.b	**0**	**0**	0.15	0.00	**0**	0.15	0.00	**0**	0.01	0.00
dh.p5.4.c	**0**	**0**	0.13	0.00	**0**	0.12	0.00	0	0.01	0.00
dh.p5.4.d	**15**	5	0.37	0.67	5	0.38	0.67	5	0.02	0.67
dh.p5.4.e	**0**	**0**	0.13	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p5.4.f	**20**	**20**	0.46	0.00	**20**	0.47	0.00	**20**	0.02	0.00
dh.p5.4.g	**30**	15	0.37	0.50	15	0.38	0.50	15	0.02	0.50
dh.p5.4.h	**15**	0	0.17	1.00	0	0.17	1.00	0	0.01	1.00
dh.p5.4.i	**45**	30	0.74	0.33	30	0.73	0.33	30	0.02	0.33
dh.p5.4.j	**95**	20	0.42	0.79	20	0.43	0.79	20	0.02	0.79
dh.p5.4.k	**105**	10	0.19	0.90	10	0.19	0.90	10	0.00	0.90
dh.p5.4.l	**140**	60	0.75	0.57	60	0.74	0.57	60	0.03	0.57
dh.p5.4.m	**175**	115	0.44	0.34	115	0.44	0.34	105	0.02	0.40
dh.p5.4.n	**185**	85	0.84	0.54	85	0.85	0.54	85	0.03	0.54
dh.p5.4.o	**175**	40	0.24	0.77	40	0.24	0.77	35	0.01	0.80
dh.p5.4.p	**245**	125	0.52	0.49	125	0.51	0.49	125	0.01	0.49
dh.p5.4.q	**265**	115	0.91	0.57	115	0.93	0.57	115	0.02	0.57
dh.p5.4.r	**300**	100	0.89	0.67	100	0.92	0.67	100	0.03	0.67
dh.p5.4.s	**255**	115	0.91	0.55	115	0.91	0.55	115	0.03	0.55
dh.p5.4.t	**245**	80	1.03	0.67	80	1.09	0.67	80	0.03	0.67
dh.p5.4.u	**250**	75	0.69	0.70	75	0.65	0.70	55	0.01	0.78
dh.p5.4.v	**390**	130	0.84	0.67	130	0.88	0.67	130	0.02	0.67
dh.p5.4.w	**355**	120	1.02	0.66	120	1.07	0.66	120	0.03	0.66
dh.p5.4.x	**420**	195	0.94	0.54	250	0.98	0.40	195	0.02	0.54
dh.p5.4.y	**495**	340	1.02	0.31	340	0.99	0.31	340	0.00	0.31
dh.p5.4.z	**420**	180	1.31	0.57	180	1.35	0.57	180	0.02	0.57

Table 4.17: Instance p6

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dw.p6.2.a	**0**	**0**	0.32	0.00	**0**	0.31	0.00	**0**	0.08	0.00
dw.p6.2.b	**0**	**0**	0.25	0.00	**0**	0.24	0.00	**0**	0.02	0.00
dw.p6.2.c	**0**	**0**	0.26	0.00	**0**	0.27	0.00	**0**	0.02	0.00
dw.p6.2.d	**186**	156	14.07	0.16	156	11.10	0.16	156	0.17	0.16
dw.p6.2.e	**348**	318	39.05	0.09	306	34.08	0.12	306	0.28	0.12
dw.p6.2.f	**540**	474	63.03	0.12	456	64.51	0.16	384	0.29	0.29
dw.p6.2.g	606	570	71.60	0.06	552	77.67	0.09	480	0.35	0.21
dw.p6.2.h	744	666	77.68	0.10	582	73.15	0.22	516	0.16	0.31
dw.p6.2.i	852	780	111.99	0.08	780	111.31	0.08	648	0.18	0.24
dw.p6.2.j	906	828	107.64	0.09	774	106.95	0.15	684	0.19	0.25
dw.p6.2.k	996	954	151.67	0.04	918	148.55	0.08	852	0.31	0.14
dw.p6.2.l	1056	972	144.12	0.08	996	126.57	0.06	870	0.24	0.18
dw.p6.2.m	1140	1044	149.72	0.08	1020	132.32	0.11	906	0.25	0.21
dw.p6.2.n	1134	1056	163.97	0.07	1086	165.71	0.04	942	0.40	0.17
dw.p6.3.a	**0**	**0**	0.46	0.00	**0**	0.39	0.00	**0**	0.02	0.00
dw.p6.3.b	**0**	**0**	0.36	0.00	**0**	0.35	0.00	**0**	0.01	0.00
dw.p6.3.c	**0**	**0**	0.41	0.00	**0**	0.34	0.00	**0**	0.02	0.00
dw.p6.3.d	**0**	**0**	0.44	0.00	**0**	0.47	0.00	**0**	0.02	0.00
dw.p6.3.e	**0**	**0**	0.39	0.00	**0**	0.38	0.00	**0**	0.02	0.00
dw.p6.3.f	**0**	**0**	0.34	0.00	**0**	0.32	0.00	**0**	0.02	0.00
dw.p6.3.g	**258**	150	10.25	0.42	150	9.09	0.42	150	0.13	0.42
dw.p6.3.h	**444**	408	44.42	0.08	402	43.72	0.09	366	0.20	0.18
dw.p6.3.i	564	504	86.93	0.11	522	82.33	0.07	474	0.27	0.16
dw.p6.3.j	822	684	109.81	0.17	690	98.02	0.16	666	0.28	0.19
dw.p6.3.k	852	822	150.02	0.04	810	126.66	0.05	714	0.59	0.16
dw.p6.3.l	978	876	163.51	0.10	870	135.11	0.11	726	0.19	0.26
dw.p6.3.m	1050	984	160.60	0.06	972	138.13	0.07	906	0.24	0.14
dw.p6.3.n	1092	966	146.15	0.12	1008	123.31	0.08	996	0.32	0.09
dw.p6.4.a	**0**	**0**	0.40	0.00	**0**	0.49	0.00	**0**	0.02	0.00
dw.p6.4.b	**0**	**0**	0.44	0.00	**0**	0.43	0.00	**0**	0.02	0.00
dw.p6.4.c	**0**	**0**	0.64	0.00	**0**	0.55	0.00	**0**	0.03	0.00
dw.p6.4.d	**0**	**0**	0.60	0.00	**0**	0.79	0.00	**0**	0.03	0.00
dw.p6.4.e	**0**	**0**	0.37	0.00	**0**	0.35	0.00	**0**	0.02	0.00
dw.p6.4.f	**0**	**0**	0.51	0.00	**0**	0.40	0.00	**0**	0.02	0.00
dw.p6.4.g	**0**	**0**	0.38	0.00	**0**	0.37	0.00	**0**	0.02	0.00
dw.p6.4.h	**0**	**0**	0.49	0.00	**0**	0.45	0.00	**0**	0.02	0.00
dw.p6.4.i	**0**	**0**	0.44	0.00	**0**	0.50	0.00	**0**	0.02	0.00
dw.p6.4.j	**294**	246	27.32	0.16	264	25.97	0.10	264	0.19	0.10
dw.p6.4.k	**498**	348	41.56	0.30	348	40.44	0.30	276	0.15	0.45
dw.p6.4.l	672	582	113.45	0.13	588	108.91	0.13	522	0.25	0.22
dw.p6.4.m	810	756	111.56	0.07	696	110.64	0.14	702	0.16	0.13
dw.p6.4.n	1014	936	159.55	0.08	924	145.82	0.09	786	0.39	0.22

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dm.p6.2.a	**0**	**0**	0.22	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dm.p6.2.b	**0**	**0**	0.21	0.00	**0**	0.23	0.00	**0**	0.01	0.00
dm.p6.2.c	**0**	**0**	0.27	0.00	**0**	0.16	0.00	**0**	0.00	0.00
dm.p6.2.d	**174**	156	3.19	0.10	156	2.77	0.10	144	0.05	0.17
dm.p6.2.e	**288**	216	8.92	0.25	216	7.27	0.25	210	0.04	0.27
dm.p6.2.f	**378**	216	4.18	0.43	216	4.35	0.43	204	0.04	0.46
dm.p6.2.g	**498**	384	10.45	0.23	384	10.65	0.23	378	0.04	0.24
dm.p6.2.h	**492**	390	13.24	0.21	378	13.31	0.23	384	0.05	0.22
dm.p6.2.i	**654**	636	21.12	0.03	630	20.86	0.04	480	0.06	0.27
dm.p6.2.j	**672**	480	15.03	0.29	432	13.14	0.36	408	0.06	0.39
dm.p6.2.k	**684**	432	9.14	0.37	474	7.87	0.31	504	0.07	0.26
dm.p6.2.l	**720**	552	27.88	0.23	552	24.99	0.23	492	0.05	0.32
dm.p6.2.m	**840**	618	26.28	0.26	612	29.12	0.27	564	0.07	0.33
dm.p6.2.n	**840**	600	31.99	0.29	600	31.09	0.29	678	0.05	0.19
dm.p6.3.a	**0**	**0**	0.20	0.00	**0**	0.26	0.00	**0**	0.01	0.00
dm.p6.3.b	**0**	**0**	0.19	0.00	**0**	0.18	0.00	**0**	0.01	0.00
dm.p6.3.c	**0**	**0**	0.23	0.00	**0**	0.25	0.00	**0**	0.01	0.00
dm.p6.3.d	**0**	**0**	0.22	0.00	**0**	0.19	0.00	**0**	0.01	0.00
dm.p6.3.e	**0**	**0**	0.20	0.00	**0**	0.20	0.00	**0**	0.01	0.00
dm.p6.3.f	**0**	**0**	0.19	0.00	**0**	0.18	0.00	**0**	0.01	0.00
dm.p6.3.g	**210**	138	1.60	0.34	132	1.65	0.37	132	0.04	0.37
dm.p6.3.h	**288**	222	3.43	0.23	222	3.11	0.23	222	0.05	0.23
dm.p6.3.i	**450**	294	6.94	0.35	270	6.38	0.40	252	0.04	0.44
dm.p6.3.j	**540**	432	12.92	0.20	432	14.13	0.20	396	0.06	0.27
dm.p6.3.k	**672**	462	21.78	0.31	462	20.44	0.31	402	0.05	0.40
dm.p6.3.l	**726**	546	22.58	0.25	546	21.03	0.25	438	0.05	0.40
dm.p6.3.m	**690**	486	18.31	0.30	486	18.49	0.30	420	0.05	0.39
dm.p6.3.n	**744**	480	26.53	0.35	480	24.78	0.35	480	0.07	0.35
dm.p6.4.a	**0**	**0**	0.23	0.00	**0**	0.31	0.00	**0**	0.01	0.00
dm.p6.4.b	**0**	**0**	0.23	0.00	**0**	0.28	0.00	**0**	0.01	0.00
dm.p6.4.c	**0**	**0**	0.29	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dm.p6.4.d	**0**	**0**	0.23	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dm.p6.4.e	**0**	**0**	0.23	0.00	**0**	0.22	0.00	**0**	0.01	0.00
dm.p6.4.f	**0**	**0**	0.26	0.00	**0**	0.25	0.00	**0**	0.02	0.00
dm.p6.4.g	**0**	**0**	0.40	0.00	**0**	0.25	0.00	**0**	0.01	0.00
dm.p6.4.h	**0**	**0**	0.23	0.00	**0**	0.28	0.00	**0**	0.01	0.00
dm.p6.4.i	**0**	**0**	0.23	0.00	**0**	0.23	0.00	**0**	0.01	0.00
dm.p6.4.j	**234**	138	2.72	0.41	144	2.79	0.38	144	0.08	0.38
dm.p6.4.k	**318**	162	4.21	0.49	162	3.63	0.49	174	0.06	0.45
dm.p6.4.l	**438**	264	8.37	0.40	264	8.38	0.40	264	0.05	0.40
dm.p6.4.m	**552**	312	7.94	0.43	312	7.99	0.43	300	0.05	0.46
dm.p6.4.n	**558**	384	18.25	0.31	384	11.44	0.31	402	0.06	0.28

Problem	Z_{CP}	MPAd			MPAc			Greedy		
		Z	CPU	%Dev	Z	CPU	%Dev	Z	CPU	%Dev
dh.p6.2.a	**0**	**0**	0.11	0.00	**0**	0.10	0.00	**0**	0.01	0.00
dh.p6.2.b	**0**	**0**	0.23	0.00	**0**	0.10	0.00	**0**	0.00	0.00
dh.p6.2.c	**0**	**0**	0.15	0.00	**0**	0.10	0.00	**0**	0.00	0.00
dh.p6.2.d	**108**	54	0.40	0.50	54	0.26	0.50	54	0.01	0.50
dh.p6.2.e	**108**	72	0.44	0.33	72	0.34	0.33	72	0.02	0.33
dh.p6.2.f	**210**	108	0.42	0.49	108	0.40	0.49	108	0.01	0.49
dh.p6.2.g	**114**	30	0.22	0.74	30	0.22	0.74	30	0.01	0.74
dh.p6.2.h	**174**	102	0.38	0.41	102	0.44	0.41	102	0.02	0.41
dh.p6.2.i	**216**	114	0.52	0.47	114	0.51	0.47	114	0.01	0.47
dh.p6.2.j	**210**	120	0.66	0.43	120	0.96	0.43	114	0.01	0.46
dh.p6.2.k	**324**	186	0.61	0.43	186	0.68	0.43	186	0.02	0.43
dh.p6.2.l	**342**	186	0.71	0.46	186	0.59	0.46	186	0.01	0.46
dh.p6.2.m	**246**	180	0.66	0.27	180	0.80	0.27	180	0.01	0.27
dh.p6.2.n	**312**	228	0.61	0.27	228	0.63	0.27	228	0.01	0.27
dh.p6.3.a	**0**	**0**	0.14	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p6.3.b	**0**	**0**	0.13	0.00	**0**	0.14	0.00	**0**	0.01	0.00
dh.p6.3.c	**0**	**0**	0.20	0.00	**0**	0.13	0.00	**0**	0.01	0.00
dh.p6.3.d	**0**	**0**	0.14	0.00	**0**	0.24	0.00	**0**	0.01	0.00
dh.p6.3.e	**0**	**0**	0.13	0.00	**0**	0.12	0.00	**0**	0.01	0.00
dh.p6.3.f	**0**	**0**	0.13	0.00	**0**	0.12	0.00	**0**	0.01	0.00
dh.p6.3.g	**102**	72	0.43	0.29	72	0.34	0.29	54	0.02	0.47
dh.p6.3.h	**120**	84	0.44	0.30	84	0.45	0.30	72	0.02	0.40
dh.p6.3.i	**168**	102	0.56	0.39	102	0.63	0.39	90	0.01	0.46
dh.p6.3.j	**198**	72	0.79	0.64	72	0.57	0.64	60	0.02	0.70
dh.p6.3.k	**120**	72	0.42	0.40	72	0.42	0.40	48	0.01	0.60
dh.p6.3.l	**168**	36	0.36	0.79	36	0.41	0.79	30	0.02	0.82
dh.p6.3.m	**210**	120	0.73	0.43	120	0.80	0.43	120	0.02	0.43
dh.p6.3.n	**252**	108	0.88	0.57	108	1.04	0.57	108	0.02	0.57
dh.p6.4.a	**0**	**0**	0.16	0.00	**0**	0.21	0.00	**0**	0.01	0.00
dh.p6.4.b	**0**	**0**	0.16	0.00	**0**	0.15	0.00	**0**	0.01	0.00
dh.p6.4.c	**0**	**0**	0.16	0.00	**0**	0.15	0.00	**0**	0.01	0.00
dh.p6.4.d	**0**	**0**	0.14	0.00	**0**	0.14	0.00	**0**	0.01	0.00
dh.p6.4.e	**0**	**0**	0.14	0.00	**0**	0.14	0.00	**0**	0.01	0.00
dh.p6.4.f	**0**	**0**	0.20	0.00	**0**	0.20	0.00	**0**	0.01	0.00
dh.p6.4.g	**0**	**0**	0.28	0.00	**0**	0.20	0.00	**0**	0.01	0.00
dh.p6.4.h	**0**	**0**	0.23	0.00	**0**	0.21	0.00	**0**	0.01	0.00
dh.p6.4.i	**0**	**0**	0.15	0.00	**0**	0.14	0.00	**0**	0.01	0.00
dh.p6.4.j	**102**	48	0.30	0.53	48	0.24	0.53	30	0.02	0.71
dh.p6.4.k	**84**	54	0.48	0.36	54	0.44	0.36	54	0.02	0.36
dh.p6.4.l	**96**	66	0.27	0.31	66	0.27	0.31	54	0.02	0.44
dh.p6.4.m	**186**	54	0.33	0.71	54	0.31	0.71	54	0.02	0.71
dh.p6.4.n	**126**	90	0.69	0.29	90	0.58	0.29	90	0.02	0.29

Table 4.18: Instance p7

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dw.p7.2.a	**30**	14	0.87	0.53	14	0.85	0.53	14	0.09	0.53
dw.p7.2.b	**64**	29	2.31	0.55	29	2.28	0.55	29	0.14	0.55
dw.p7.2.c	**101**	48	2.24	0.52	48	2.21	0.52	48	0.07	0.52
dw.p7.2.d	**190**	179	23.26	0.06	152	17.04	0.20	148	0.33	0.22
dw.p7.2.e	**267**	233	50.20	0.13	206	33.86	0.23	205	0.37	0.23
dw.p7.2.f	**365**	299	76.18	0.18	276	67.26	0.24	220	0.51	0.40
dw.p7.2.g	438	416	174.78	0.05	400	166.07	0.09	267	0.67	0.39
dw.p7.2.h	505	360	124.78	0.29	451	160.73	0.11	301	0.65	0.40
dw.p7.2.i	561	509	288.87	0.09	505	268.14	0.10	410	0.84	0.27
dw.p7.2.j	614	531	308.79	0.14	514	285.59	0.16	391	1.01	0.36
dw.p7.2.k	665	614	428.32	0.08	605	366.84	0.09	474	1.39	0.29
dw.p7.2.l	657	670	432.98	-0.02	655	397.00	0.00	428	1.00	0.35
dw.p7.2.m	758	719	463.42	0.05	642	438.52	0.15	538	0.74	0.29
dw.p7.2.n	782	793	837.31	-0.01	773	722.51	0.01	593	2.60	0.24
dw.p7.2.o	831	807	752.90	0.03	807	699.67	0.03	575	1.63	0.31
dw.p7.2.p	926	805	755.51	0.13	845	756.02	0.09	630	1.83	0.32
dw.p7.2.q	867	832	704.67	0.04	832	680.28	0.04	622	0.28	0.28
dw.p7.2.r	1038	939	969.79	0.10	911	802.09	0.12	645	2.50	0.38
dw.p7.2.s	995	967	1125.58	0.03	976	1015.04	0.02	750	0.85	0.25
dw.p7.2.t	1065	940	888.16	0.12	984	766.78	0.08	753	0.97	0.29
dw.p7.3.a	**0**	**0**	0.89	0.00	**0**	0.91	0.00	**0**	0.02	0.00
dw.p7.3.b	**30**	0	0.92	1.00	0	1.05	1.00	0	0.03	1.00
dw.p7.3.c	**79**	46	4.10	0.42	46	4.34	0.42	46	0.12	0.42
dw.p7.3.d	**114**	64	3.51	0.44	64	3.76	0.44	64	0.07	0.44
dw.p7.3.e	**175**	124	8.27	0.29	124	8.55	0.29	112	0.22	0.36
dw.p7.3.f	**237**	145	18.81	0.39	135	15.35	0.43	136	0.29	0.43
dw.p7.3.g	**335**	276	73.33	0.18	256	68.80	0.24	224	0.29	0.33
dw.p7.3.h	**415**	331	107.64	0.20	341	104.80	0.18	255	0.45	0.39
dw.p7.3.i	468	377	133.73	0.19	382	124.58	0.18	233	0.38	0.50
dw.p7.3.j	504	443	191.75	0.12	439	186.85	0.13	372	0.46	0.26
dw.p7.3.k	582	523	326.56	0.10	495	329.57	0.15	442	1.01	0.24
dw.p7.3.l	645	587	398.66	0.09	588	347.17	0.09	466	1.14	0.28
dw.p7.3.m	694	657	436.90	0.05	637	383.28	0.08	508	0.81	0.27
dw.p7.3.n	747	696	555.04	0.07	689	538.16	0.08	475	1.44	0.36
dw.p7.3.o	791	715	556.85	0.10	709	504.35	0.10	523	1.39	0.34
dw.p7.3.p	808	748	503.24	0.07	758	470.58	0.06	490	0.86	0.39
dw.p7.3.q	884	840	750.74	0.05	816	705.08	0.08	601	1.11	0.32
dw.p7.3.r	964	899	822.12	0.07	891	775.41	0.08	667	0.81	0.31
dw.p7.3.s	899	899	809.24	0.00	899	737.81	0.00	636	1.13	0.29
dw.p7.3.t	1032	960	995.95	0.07	987	942.81	0.04	633	1.17	0.39
dw.p7.4.a	**0**	**0**	1.08	0.00	**0**	1.13	0.00	**0**	0.03	0.00
dw.p7.4.b	**14**	**14**	0.97	0.00	**14**	1.02	0.00	**14**	0.02	0.00
dw.p7.4.c	**46**	30	1.04	0.35	30	1.11	0.35	30	0.03	0.35
dw.p7.4.d	**79**	64	6.31	0.19	64	7.00	0.19	46	0.12	0.42
dw.p7.4.e	**123**	86	5.98	0.30	86	6.58	0.30	86	0.16	0.30
dw.p7.4.f	**139**	102	6.17	0.27	102	6.75	0.27	102	0.18	0.27
dw.p7.4.g	**217**	168	22.47	0.23	168	23.84	0.23	189	0.29	0.13
dw.p7.4.h	**285**	236	39.63	0.17	198	42.53	0.31	198	0.41	0.31
dw.p7.4.i	**353**	294	76.71	0.17	278	78.85	0.21	251	0.32	0.29
dw.p7.4.j	**438**	385	194.02	0.12	371	211.40	0.15	332	0.61	0.24
dw.p7.4.k	510	431	144.37	0.15	383	132.43	0.25	268	0.49	0.47
dw.p7.4.l	558	500	286.29	0.10	500	307.61	0.10	411	0.74	0.26
dw.p7.4.m	627	531	401.14	0.15	527	410.50	0.16	398	1.44	0.37
dw.p7.4.n	677	640	494.80	0.05	640	463.26	0.05	524	1.82	0.23
dw.p7.4.o	722	629	437.39	0.13	649	484.78	0.10	456	0.48	0.37
dw.p7.4.p	814	753	542.08	0.07	744	520.20	0.09	541	1.51	0.34
dw.p7.4.q	829	793	666.47	0.04	743	697.32	0.10	647	1.96	0.22
dw.p7.4.r	861	736	629.84	0.15	762	615.92	0.11	567	0.53	0.34
dw.p7.4.s	951	873	793.85	0.08	871	786.19	0.08	633	1.65	0.33
dw.p7.4.t	960	867	782.03	0.10	899	655.93	0.06	639	1.06	0.33

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dm.p7.2.a	**0**	**0**	0.63	0.00	**0**	0.61	0.0%	**0**	0.00	0.00
dm.p7.2.b	**30**	14	0.56	0.53	14	0.54	53.3%	14	0.01	0.53
dm.p7.2.c	**86**	58	1.20	0.33	58	1.18	32.6%	58	0.02	0.33
dm.p7.2.d	**147**	104	1.95	0.29	104	1.83	29.3%	104	0.05	0.29
dm.p7.2.e	**213**	163	3.98	0.23	167	4.14	21.6%	197	0.06	0.08
dm.p7.2.f	**254**	202	8.05	0.20	202	8.04	20.5%	197	0.09	0.22
dm.p7.2.g	**305**	261	7.29	0.14	230	10.98	24.6%	215	0.09	0.30
dm.p7.2.h	**410**	321	25.41	0.22	321	16.67	21.7%	249	0.07	0.39
dm.p7.2.i	**392**	282	18.19	0.28	320	18.79	18.4%	242	0.14	0.38
dm.p7.2.j	471	330	27.89	0.30	304	43.70	35.5%	322	0.10	0.32
dm.p7.2.k	496	406	64.63	0.18	406	63.96	18.1%	266	0.07	0.46
dm.p7.2.l	528	423	49.62	0.20	413	57.68	21.8%	298	0.07	0.44
dm.p7.2.m	570	449	73.80	0.21	459	54.37	19.5%	424	0.11	0.26
dm.p7.2.n	600	545	147.13	0.09	538	160.06	10.3%	385	0.13	0.36
dm.p7.2.o	699	597	122.74	0.15	590	126.69	15.6%	414	0.13	0.41
dm.p7.2.p	665	566	128.70	0.15	556	136.18	16.4%	451	0.09	0.32
dm.p7.2.q	752	608	129.56	0.19	606	137.35	19.4%	503	0.09	0.33
dm.p7.2.r	798	641	129.74	0.20	635	129.79	20.4%	493	0.09	0.38
dm.p7.2.s	723	614	174.97	0.15	629	133.21	13.0%	552	0.11	0.24
dm.p7.2.t	778	720	175.32	0.07	711	145.92	8.6%	603	0.12	0.22
dm.p7.3.a	**0**	**0**	0.67	0.00	**0**	0.70	0.0%	**0**	0.02	0.00
dm.p7.3.b	**30**	0	0.60	1.00	0	0.67	100.0%	0	0.01	1.00
dm.p7.3.c	**32**	0	0.59	1.00	0	0.65	100.0%	0	0.01	1.00
dm.p7.3.d	**75**	27	1.37	0.64	27	1.54	64.0%	27	0.03	0.64
dm.p7.3.e	**111**	48	1.83	0.57	48	2.19	56.8%	48	0.02	0.57
dm.p7.3.f	**209**	93	3.35	0.56	105	3.71	49.8%	93	0.07	0.56
dm.p7.3.g	**183**	136	4.83	0.26	136	4.93	25.7%	131	0.08	0.28
dm.p7.3.h	**291**	208	14.10	0.29	218	19.48	25.1%	197	0.07	0.32
dm.p7.3.i	**362**	295	12.07	0.19	295	12.63	18.5%	243	0.12	0.33
dm.p7.3.j	**438**	309	39.50	0.29	283	44.32	35.4%	285	0.10	0.35
dm.p7.3.k	**425**	320	25.21	0.25	320	22.59	24.7%	265	0.07	0.38
dm.p7.3.l	465	379	57.49	0.18	381	43.40	18.1%	352	0.08	0.24
dm.p7.3.m	559	481	74.06	0.14	496	82.47	11.3%	412	0.08	0.26
dm.p7.3.n	529	354	19.55	0.33	358	25.46	32.3%	344	0.08	0.35
dm.p7.3.o	608	495	84.37	0.19	495	82.74	18.6%	363	0.07	0.40
dm.p7.3.p	686	509	79.61	0.26	529	65.84	22.9%	414	0.08	0.40
dm.p7.3.q	718	618	124.08	0.14	630	115.96	12.3%	506	0.11	0.30
dm.p7.3.r	635	522	79.42	0.18	517	71.93	18.6%	426	0.09	0.33
dm.p7.3.s	764	677	107.92	0.11	687	106.29	10.1%	632	0.12	0.17
dm.p7.3.t	781	654	150.89	0.16	603	143.82	22.8%	522	0.10	0.33
dm.p7.4.a	**0**	**0**	0.72	0.00	**0**	0.77	0.0%	**0**	0.02	0.00
dm.p7.4.b	**14**	0	0.72	1.00	0	0.74	100.0%	0	0.02	1.00
dm.p7.4.c	**30**	0	0.79	1.00	0	0.81	100.0%	0	0.02	1.00
dm.p7.4.d	**47**	15	0.80	0.68	15	0.83	68.1%	15	0.01	0.68
dm.p7.4.e	**114**	73	2.15	0.36	73	2.38	36.0%	73	0.03	0.36
dm.p7.4.f	**131**	60	2.73	0.54	60	3.09	54.2%	60	0.06	0.54
dm.p7.4.g	**157**	73	3.75	0.54	73	4.21	53.5%	73	0.07	0.54
dm.p7.4.h	**188**	100	4.08	0.47	100	4.62	46.8%	100	0.07	0.47
dm.p7.4.i	**221**	180	9.47	0.19	180	9.62	18.6%	180	0.07	0.19
dm.p7.4.j	**347**	284	20.84	0.18	265	16.32	23.6%	284	0.09	0.18
dm.p7.4.k	**365**	257	19.19	0.30	244	14.23	33.2%	236	0.10	0.35
dm.p7.4.l	**454**	353	42.41	0.22	353	44.09	22.2%	306	0.11	0.33
dm.p7.4.m	440	272	21.54	0.38	272	18.98	38.2%	238	0.11	0.46
dm.p7.4.n	489	332	25.91	0.32	314	20.82	35.8%	281	0.11	0.43
dm.p7.4.o	611	478	64.68	0.22	476	60.86	22.1%	408	0.07	0.33
dm.p7.4.p	573	493	52.28	0.14	495	46.65	13.6%	383	0.10	0.33
dm.p7.4.q	642	479	89.57	0.25	469	94.97	26.9%	490	0.08	0.24
dm.p7.4.r	605	410	60.00	0.32	404	62.71	33.2%	420	0.08	0.31
dm.p7.4.s	747	633	101.69	0.15	596	96.69	20.2%	376	0.07	0.50
dm.p7.4.t	711	572	120.78	0.20	572	133.98	19.5%	468	0.08	0.34

Problem	Z_{CP}	MPAd Z	MPAd CPU	MPAd %Dev	MPAc Z	MPAc CPU	MPAc %Dev	Greedy Z	Greedy CPU	Greedy %Dev
dh.p7.2.a	**0**	**0**	0.43	0.00	**0**	0.41	0.00	**0**	0.01	0.00
dh.p7.2.b	**16**	**16**	0.31	0.00	**16**	0.29	0.00	**16**	0.01	0.00
dh.p7.2.c	**23**	0	0.31	1.00	0	0.30	1.00	0	0.01	1.00
dh.p7.2.d	**49**	0	0.30	1.00	0	0.29	1.00	0	0.00	1.00
dh.p7.2.e	**90**	60	0.66	0.33	60	0.63	0.33	60	0.02	0.33
dh.p7.2.f	**145**	67	0.52	0.54	62	0.55	0.57	62	0.03	0.57
dh.p7.2.g	**134**	96	0.63	0.28	96	0.66	0.28	96	0.02	0.28
dh.p7.2.h	**142**	89	0.90	0.37	89	1.00	0.37	89	0.02	0.37
dh.p7.2.i	**180**	61	0.49	0.66	61	0.50	0.66	61	0.02	0.66
dh.p7.2.j	**255**	199	1.01	0.22	148	1.01	0.42	199	0.02	0.22
dh.p7.2.k	**233**	108	1.22	0.54	108	1.20	0.54	108	0.03	0.54
dh.p7.2.l	**260**	153	1.43	0.41	153	1.37	0.41	144	0.02	0.45
dh.p7.2.m	**297**	137	0.89	0.54	115	0.84	0.61	121	0.01	0.59
dh.p7.2.n	**349**	227	1.21	0.35	227	1.51	0.35	243	0.02	0.30
dh.p7.2.o	**382**	204	4.22	0.47	204	4.63	0.47	204	0.02	0.47
dh.p7.2.p	**401**	187	1.23	0.53	176	0.78	0.56	152	0.02	0.62
dh.p7.2.q	**390**	307	6.03	0.21	307	4.61	0.21	234	0.02	0.40
dh.p7.2.r	**413**	278	1.35	0.33	278	2.26	0.33	283	0.02	0.31
dh.p7.2.s	**372**	206	2.96	0.45	206	4.96	0.45	158	0.03	0.58
dh.p7.2.t	**349**	215	1.39	0.38	215	1.47	0.38	239	0.02	0.32
dh.p7.3.a	**0**	**0**	0.30	0.00	**0**	0.29	0.00	**0**	0.01	0.00
dh.p7.3.b	**16**	**16**	0.61	0.00	**16**	0.60	0.00	**16**	0.02	0.00
dh.p7.3.c	**16**	**16**	0.62	0.00	**16**	0.62	0.00	**16**	0.02	0.00
dh.p7.3.d	**34**	0	0.40	1.00	0	0.40	1.00	0	0.01	1.00
dh.p7.3.e	**15**	**15**	0.32	0.00	**15**	0.32	0.00	**15**	0.01	0.00
dh.p7.3.f	**47**	44	1.77	0.06	44	1.43	0.06	44	0.03	0.06
dh.p7.3.g	**96**	55	0.57	0.43	55	0.59	0.43	55	0.02	0.43
dh.p7.3.h	**143**	87	0.66	0.39	87	0.66	0.39	87	0.03	0.39
dh.p7.3.i	**237**	156	1.51	0.34	156	1.48	0.34	156	0.03	0.34
dh.p7.3.j	**208**	132	1.20	0.37	132	1.21	0.37	132	0.03	0.37
dh.p7.3.k	**182**	122	1.56	0.33	122	1.78	0.33	103	0.02	0.43
dh.p7.3.l	**241**	130	1.61	0.46	148	1.37	0.39	117	0.02	0.51
dh.p7.3.m	**353**	223	1.78	0.37	197	1.98	0.44	195	0.02	0.45
dh.p7.3.n	**271**	131	2.48	0.52	131	2.44	0.52	131	0.04	0.52
dh.p7.3.o	**252**	136	1.07	0.46	136	1.42	0.46	136	0.04	0.46
dh.p7.3.p	**312**	149	1.13	0.52	149	1.18	0.52	149	0.02	0.52
dh.p7.3.q	**399**	205	1.54	0.49	205	1.68	0.49	145	0.04	0.64
dh.p7.3.r	**276**	138	2.29	0.50	138	2.51	0.50	138	0.02	0.50
dh.p7.3.s	**423**	295	1.33	0.30	295	1.42	0.30	278	0.03	0.34
dh.p7.3.t	**465**	230	2.63	0.51	240	2.13	0.48	230	0.02	0.51
dh.p7.4.a	**0**	**0**	0.36	0.00	**0**	0.34	0.00	**0**	0.01	0.00
dh.p7.4.b	**0**	**0**	0.39	0.00	**0**	0.41	0.00	**0**	0.01	0.00
dh.p7.4.c	**0**	**0**	0.46	0.00	**0**	0.38	0.00	**0**	0.01	0.00
dh.p7.4.d	**14**	0	0.44	1.00	0	0.45	1.00	0	0.01	1.00
dh.p7.4.e	**16**	**16**	0.41	0.00	**16**	0.43	0.00	**16**	0.01	0.00
dh.p7.4.f	**16**	0	0.45	1.00	0	0.47	1.00	0	0.01	1.00
dh.p7.4.g	**70**	0	0.43	1.00	0	0.44	1.00	0	0.00	1.00
dh.p7.4.h	**93**	28	0.75	0.70	28	0.78	0.70	28	0.02	0.70
dh.p7.4.i	**87**	46	2.18	0.47	46	2.07	0.47	30	0.02	0.66
dh.p7.4.j	**176**	114	2.30	0.35	114	2.19	0.35	114	0.02	0.35
dh.p7.4.k	**116**	75	0.81	0.35	75	0.79	0.35	75	0.02	0.35
dh.p7.4.l	**127**	57	2.69	0.55	57	2.53	0.55	57	0.05	0.55
dh.p7.4.m	**196**	126	1.20	0.36	126	1.21	0.36	126	0.02	0.36
dh.p7.4.n	**261**	163	2.68	0.38	163	2.67	0.38	163	0.02	0.38
dh.p7.4.o	**200**	117	1.30	0.42	117	1.27	0.42	117	0.03	0.42
dh.p7.4.p	**306**	119	3.09	0.61	119	3.10	0.61	119	0.02	0.61
dh.p7.4.q	**247**	102	2.75	0.59	102	2.71	0.59	102	0.03	0.59
dh.p7.4.r	**296**	201	1.55	0.32	201	1.39	0.32	174	0.03	0.41
dh.p7.4.s	**248**	167	3.11	0.33	144	3.16	0.42	167	0.03	0.33
dh.p7.4.t	**328**	162	2.59	0.51	162	2.29	0.51	162	0.02	0.51

4.6 Conclusions

In this paper, we introduce a new dynamic routing problem, namely the Dynamic Team Orienteering Problem (DTOP) in which some customers are known *a priori* while others are dynamic, each associated with a profit. The goal is to maximize the sum of collected profits by visiting a set of customer locations within a time limit. This problem arises in several practical applications such as disaster relief, technician, tourist and school bus routing problems. We adapt a Multiple Plan Approach (MPA) to solve the proposed problem and consider both a consensus function method (MPAc) and demand served method (MPAd) for selecting a distinguished plan from the pool of alternative solutions. To evaluate the quality of MPAd and MPAc, a sophisticated greedy algorithm is employed for solving the DTOP and a reference offline algorithm is employed for solving the static variant. A total of 1161 new DTOP benchmark instances are introduced, adapted from TOP benchmark instances by generating customer disclosure times and varying the effective degree of dynamism. Algorithms are compared on the basis of average percent deviation from the offline solutions. The average deviation for MPAd, MPAc, and Greedy are 29.8%, 30.1%, and 35.0%, respectively, while Greedy has a lower runtime. A lower percent of deviation indicates a better performance, and thus, MPAd outperforms MPAc and Greedy. These results indicate that it is beneficial to use maximizing profit for choosing the distinguished plan instead of consensus function. In addition, the greedy algorithm yields high-quality results in less computational time.

The results presented in this paper rely on several limitations. First, we assume that a vehicle can only wait at its current location until a new customer becomes available to visit. However, alternative waiting strategies may produce high-quality routing plans such as delaying departure time as much as possible from the vehicle's current location, or intermediate waiting strategies instead of moving as soon as possible to its next location. Second, in this study, MPA is implemented sequentially such that one plan is created at a time. However, the parallel implementation of MPA, where several plans are considered simultaneously, can be developed for time-consuming tasks (e.g. creating multiple initial solutions) to reduce the overall computational time of the algorithm. This implementation would differ from parallel ALNS (pALNS) proposed by Pillac et al. **??** because MPA continuously optimizes multiple solutions whereas pALNS produces a new single solution each time a new customer appears. Third, ALNS is used as a local search in MPA. However, alternative heuristic algorithms (e.g. tabu search and simulated annealing), or exact solution algorithms (e.g. constraint programming) can also be employed as a local search algorithm in order to improve solution quality. Finally, in this study, we only consider two ranking function methods (consensus function and demand served) to choose a distinguished plan. However, alternative methods (e.g. distance traveled method) in MPA can be investigated. Additional areas for future study are to consider a single vehicle variant of DTOP, called Dynamic Orienteering Problem (DOP), and to consider a time window variant of DTOP, called Dynamic Team Orienteering Problem with Time Windows (DTOPTW). A set of new benchmark instances and alternative solution methods may be introduced for the computational study of those problems.

4.7 Appendix. Average percentage deviations

Table 4.19: Average deviation from optimal offline solutions

Problem Set	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
p1	17.5%	18.0%	22.8%	40.6%	40.8%	43.5%	39.2%	39.2%	39.5%
p2	15.0%	15.8%	20.9%	45.4%	45.5%	46.1%	51.2%	51.2%	51.2%
p3	17.8%	18.0%	26.8%	39.8%	39.1%	41.7%	46.6%	46.6%	47.2%
p4	23.4%	23.8%	30.9%	27.7%	27.8%	32.8%	48.8%	48.5%	50.6%
p5	18.7%	18.9%	21.3%	37.1%	37.6%	41.6%	47.1%	46.9%	47.6%
p6	5.3%	5.4%	6.9%	16.8%	17.1%	18.9%	25.7%	25.7%	28.0%
p7	27.8%	30.3%	33.8%	39.8%	39.9%	42.4%	41.0%	41.6%	42.8%
Average	17.9%	18.6%	23.3%	35.3%	35.4%	38.1%	42.8%	42.8%	43.8%

Table 4.20: Average deviation from non-optimal offline solutions

Problem Set	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
p1	-	-	-	-	-	-	-	-	-
p2	-	-	-	-	-	-	-	-	-
p3	-	-	-	-	-	-	-	-	-
p4	6.6%	7.0%	28.0%	17.9%	18.2%	32.2%	-	-	-
p5	10.0%	11.5%	22.8%	24.9%	25.5%	34.7%	-	-	-
p6	8.7%	10.1%	19.2%	-	-	-	-	-	-
p7	8.7%	9.1%	32.3%	19.9%	20.5%	33.7%	-	-	-
Average	8.5%	9.4%	25.6%	20.9%	21.4%	33.6%	-	-	-

Table 4.21: Average deviation of request volume from optimal offline solutions

Request Volume	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
Low (p1, p2 and p3)	16.8%	17.3%	23.5%	41.9%	41.8%	43.8%	45.7%	45.7%	46.0%
Medium (p5 and p6)	12.0%	12.2%	14.1%	27.0%	27.3%	30.3%	36.4%	36.3%	37.8%
High (p4 and p7)	25.6%	27.1%	32.4%	33.7%	33.9%	37.6%	44.9%	45.1%	46.7%

Table 4.22: Average deviation of request volume from non-optimal offline solutions

Request Volume	Weakly Dynamic			Moderately Dynamic			Highly Dynamic		
	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy	MPAd	MPAc	Greedy
Low (p1, p2 and p3)	-	-	-	-	-	-	-	-	-
Medium (p5 and p6)	9.4%	10.8%	21.0%	24.9%	25.5%	34.7%	-	-	-
High (p4 and p7)	7.6%	8.1%	30.2%	18.9%	19.4%	33.0%	-	-	-

Bibliography

[1] W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, and D. V. Oudheusden, "A personalized tourist trip design algorithm for mobile tourist guides," Applied Artificial Intelligence, vol. 22, no. 10, pp. 964–985, 2008.

[2] S. E. Butt and T. M. Cavalier, "A heuristic for the multiple tour maximum collection problem," Computers & Operations Research, vol. 21, no. 1, pp. 101–111, 1994.

[3] H. Tang and E. Miller-Hooks, "A tabu search heuristic for the team orienteering problem," Computers & Operations Research, vol. 32, no. 6, pp. 1379–1407, 2005.

[4] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," European journal of operational research, vol. 88, no. 3, pp. 464–474, 1996.

[5] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," European Journal of Operational Research, vol. 209, no. 1, pp. 1–10, 2011.

[6] H. C. Lau, W. Yeoh, P. Varakantham, D. T. Nguyen, and H. Chen, "Dynamic stochastic orienteering problems for risk-aware applications," arXiv preprint arXiv:1210.4874, 2012.

[7] S. Zhang, J. W. Ohlmann, and B. W. Thomas, "Dynamic orienteering on a network of queues," working paper, 2015.

[8] J. Li, "Research on team orienteering problem with dynamic travel times," Journal of Software, vol. 7, no. 2, pp. 249–255, 2012.

[9] C. G. Victor Pillac, Michel Gendreau and A. L. Medaglia, "A review of dynamic vehicle routing problems," European Journal of Operational Research, vol. 225, no. 1, pp. 1–11, 2013.

[10] M. Gendreau, F. Guertin, J.-Y. Potvin, and E. Taillard, "Parallel tabu search for real-time vehicle routing and dispatching," Transportation science, vol. 33, no. 4, pp. 381–390, 1999.

[11] A. Larsen, O. Madsen, and M. Solomon, "Partially dynamic vehicle routing-models and algorithms," Journal of the operational research society, pp. 637–646, 2002.

[12] L. M. Hvattum, A. Løkketangen, and G. Laporte, "Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic," Transportation Science, vol. 40, no. 4, pp. 421–438, 2006.

[13] H.-K. C. Che-Fu Hsueh and H.-W. Chou, Dynamic Vehicle Routing for Relief Logistics in Natural Disasters, Vehicle Routing Problem, Tonci Caric and Hrvoje Gold (Ed.). ISBN: 978-953-7619-09-1: InTech, 2008.

[14] V. Pillac, C. Guéret, and A. Medaglia, "On the dynamic technician routing and scheduling problem," tech. rep., Écoledes Minesde Nantes, Nantes, France, 2012.

[15] R. W. Bent and P. V. Hentenryck, "Scenario-based planning for partially dynamic vehicle routing with stochastic customers," Operations Research, vol. 6, pp. 977–987, 2004.

[16] R. Gedik, E. Kirac, A. Milburn, and C. Rainwater, "A constraint programming approach for the team orienteering problem with time windows," Manuscript submitted for publication.

[17] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," European journal of operational research, vol. 88, no. 3, pp. 464–474, 1996.

[18] A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte, "Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem," Parallel Computing, vol. 30, no. 3, pp. 377–387, 2004.

[19] A. Beaudry, G. Laporte, T. Melo, and S. Nickel, "Dynamic transportation of patients in hospitals," OR spectrum, vol. 32, no. 1, pp. 77–107, 2010.

[20] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," Journal of Combinatorial Optimization, vol. 10, no. 4, pp. 327–343, 2005.

[21] A. E. Rizzoli, R. Montemanni, E. Lucibello, and L. M. Gambardella, "Ant colony optimization for real-world vehicle routing problems," Swarm Intelligence, vol. 1, no. 2, pp. 135–151, 2007.

[22] A. Goel and V. Gruhn, "A general vehicle routing problem," European Journal of Operational Research, vol. 191, no. 3, pp. 650–660, 2008.

[23] I. Benyahia and J.-Y. Potvin, "Decision support for vehicle dispatching using genetic programming," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 28, no. 3, pp. 306–314, 1998.

[24] J. Yang, P. Jaillet, and H. Mahmassani, "Real-time multivehicle truckload pickup and delivery problems," Transportation Science, vol. 38, no. 2, pp. 135–148, 2004.

[25] Z.-L. Chen and H. Xu, "Dynamic column generation for dynamic vehicle routing with time windows," Transportation Science, vol. 40, no. 1, pp. 74–88, 2006.

[26] S. Mitrović-Minić, R. Krishnamurti, and G. Laporte, "Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows," Transportation Research Part B: Methodological, vol. 38, no. 8, pp. 669–685, 2004.

[27] R. Montemanni and L. M. Gambardella, "An ant colony system for team orienteering problems with time windows," Foundations of Computing and Decision Sciences, vol. 34, pp. 287–306, 2009.

[28] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "Iterated local search for the team orienteering problem with time windows," Computers & Operations Research, vol. 36, no. 12, pp. 3281 – 3290, 2009.

[29] F. Tricoire, M. Romauch, K. Doerner, and R. Hartl, "Heuristics for the multi-period orienteering problem with multiple time windows," Computers & Operations Research, vol. 37 (2), pp. 351–367, 2010.

[30] S.-W. Lin and V. F. Yu, "A simulated annealing heuristic for the team orienteering problem with time windows," European Journal of Operational Research, vol. 217(1), pp. 94–107, 2012.

[31] N. Labadie, R. Mansini, and C. R. Melechovsk, J., "The team orienteering problem with time windows: An lp-based granular variable neighborhood search," European Journal of Operational Research, vol. 220(1), pp. 15–27, 2012.

[32] H. Qian and L. Andrew, "An iterative three-component heuristic for the team orienteering problem with time windows," European Journal of Operational Research, vol. 232, pp. 276–286, 2014.

[33] T. Cura, "An artificial bee colony algorithm approach for the team orienteering problem with time windows," Computers & Industrial Engineering, vol. 74, pp. 270–290, 2014.

[34] S. Wohlgemuth, R. Oloruntoba, and U. Clausen, "Dynamic vehicle routing with anticipation in disaster relief," Socio-Economic Planning Sciences, vol. 46, no. 4, pp. 261–271, 2012.

[35] A. Larsen, The dynamic vehicle routing problem. PhD thesis, Technical University of Denmark, 2000.

[36] S. Hozo, B. Djulbegovic, and I. Hozo, "Estimating the mean and variance from the median, range, and the size of a sample," BMC medical research methodology, vol. 5, no. 1, p. 1, 2005.

5. CONCLUSIONS

Natural disasters are unpredictable incidents that create emergency situations and cause economic, social, and environmental damages. These extreme events leave people in need of emergency supplies like water, food, and medical help. The rapid delivery of these basic supplies is critical in minimizing the suffering of impacted populations. Humanitarian relief agencies and nongovernment organizations play a key role in disaster response processes using their logistics activities. Those processes often involve complex logistics problems with a high degree of uncertainty, and the use of scarce resources needs to be optimized to save lives after a disaster. **The focus of this dissertation is to introduce a new class of decision support models for disaster response logistics planning.**

Possessing an understanding of the disaster situation at hand in order to inform better decisions is critical in planning disaster logistics activities. Traditionally, assessment teams collect information regarding the impacted population by visiting the disaster area, but some in the disaster response community are exploring ways to utilize social media to collect more situational awareness information in a shorter amount of time. However, the trustworthiness and reliability of social data are significant concerns for the emergency response community because a large portion of this information is initially not verified, and some of it may be inaccurate. Therefore, this dissertation investigates whether considering and acting on social data prior to its absolute verification improves the efficacy of response plans.

The models and case studies presented in this dissertation are intended to provide insight into the impact of considering social data in tour planning. In Chapter 2, a new problem framework that describes a formal method for assessing the impact of incorporating unverified social data in disaster relief planning is introduced. This framework is demonstrated in the context of the traveling salesman problem. Two alternative decision strategies (one that considers social data and one that does not) are compared across a wide variety of demand scenarios in order to explore the benefits associated with incorporating social data in disaster relief tour planning. The computational results demonstrate that in general, the decision strategy that includes social data outperforms one that does not. Chapter 3 extends the framework introduced in the second chapter by considering practical model elements such as multiple vehicles, location specific demand magnitude, and service time. Additionally, a new objective, maximizing amount of demand served, is considered. The problem under consideration is a Team Orienteering Problem variant with a single depot. Similar to Chapter 2, two social data logistics strategies are presented: one strategy acts on social data while the other strategy does not until it is verified. However, only extreme "all or nothing" types of decision strategies are considered in the second chapter. In addition to these two approaches, three new strategies that an emergency manager could adapt in practice, are constructed. These strategies consider and act on social data in some capacity: one that specifies acting on social data if all verified request have been visited, one that specifies acting social data but higher priority is given to verified requests, and one that specifies acting on a social data location if it is located close to a verified location. A case study motivated by the 2010 Haiti earthquake is used to test the developed models. The results show that blending social data with traditional data in disaster response is promising. In Chapter 4, a new orienteering problem variant, called Dynamic Team Orienteering Problem (DTOP) is introduced in order to extend the static decision framework studied in Chapters 2 and 3 to a dynamic decision framework in the future. The dynamic TOP differs from the static TOP in that demand locations are revealed over time instead of being known a priori. The goal is to maximize the sum of collected demand by visiting a set of request locations within a time limit. We employ a Multiple Plan Approach (MPA) to solve DTOP. A total of 1161 new DTOP benchmark instances are introduced, adapted from TOP benchmark instances by generating customer disclosure times and varying the effective degree of dynamism. The average percentage deviation for the developed algorithms is computed using an optimal offline algorithm.

As future work, the modeling and framework introduced in Chapters 2 and 3 can be extended by considering the dynamic nature of social and traditional data available to emergency response planners. The decision process can be performed considering two input streams: (i) unverified social data describing demand that is not known with certainty, obtained by social media platforms and (ii) verified data describing demand known with certainty, obtained by trusted traditional sources. Additionally, information describing the degree of belief in the accuracy of social data elements, which will be available from the verification process, can be also considered. These streams (traditional data, social data, and information from the verification process) will be continuously updated and changed throughout the planning period as needs are received and discovered. The focus of this future study will be to develop decision support models considering these input streams when developing real-time disaster response planning. This complex planning problem is different from the classical dynamic and stochastic VRP models in the literature because, unlike traditional assumptions of having a homeostatic probability distribution to model random variables, a heterostatic probability distribution from verification efforts will be used to describe uncertain demands. Results of this research will be addressing concerns over the usefulness of social data in emergency response decision-making by quantifying the value of considering the information in real time.