


8-2012

Optimization Models and Algorithms for Truckload Relay Network Design

Hector Andres Vergara Arteaga
University of Arkansas, Fayetteville

Follow this and additional works at: <https://scholarworks.uark.edu/etd>

 Part of the [Industrial Technology Commons](#), [Operational Research Commons](#), and the [Transportation Engineering Commons](#)

Citation

Vergara Arteaga, H. A. (2012). Optimization Models and Algorithms for Truckload Relay Network Design. *Graduate Theses and Dissertations* Retrieved from <https://scholarworks.uark.edu/etd/494>

This Dissertation is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu.

OPTIMIZATION MODELS AND ALGORITHMS FOR TRUCKLOAD RELAY NETWORK
DESIGN

OPTIMIZATION MODELS AND ALGORITHMS FOR TRUCKLOAD RELAY NETWORK
DESIGN

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

By

Hector Andres Vergara Arteaga
Universidad San Francisco de Quito
Bachelor of Science in Industrial Engineering, 2002
Oregon State University
Master of Science in Industrial Engineering, 2005

August 2012
University of Arkansas

ABSTRACT

Driver turnover is a significant problem for full truckload (TL) carriers that operate using point-to-point (PtP) dispatching. The low quality of life of drivers due to the long periods of time they spend away from home is usually identified as one of the main reasons for the high turnover. In contrast, driver turnover is not as significant for less-than-truckload (LTL) carriers that use hub-and-spoke transportation networks which allow drivers to return home more frequently. Based on the differences between TL and LTL, the use of a *relay network* (RN) has been proposed as an alternative dispatching method for TL transportation in order to improve driver retention. In a RN, a truckload visits one or more *relay points* (RPs) where drivers and trailers are exchanged while the truckload continues its movement to the final destination.

In this research, we propose a new *composite variable model* (CVM) to address the strategic TL relay network design (TLRND) problem. With this approach, we capture operational considerations implicitly within the variable definition instead of adding them as constraints in our model. Our composites represent feasible routes for the truckloads through the RN that satisfy limitations on circuitry, number of RPs visited, and distances between RPs and between a RP and origin-destination nodes. Given a strict limitation on the number of RPs allowed to be visited, we developed a methodology to generate feasible routes using predefined templates. This methodology was preferred over an exact feasible path enumeration algorithm that was also developed to generate valid routes. The proposed approach was successfully used to obtain high quality solutions to largely-sized problem instances of TLRND.

Furthermore extending the original CVM formulation, we incorporate mixed fleet dispatching decisions into the design of the RN. This alternative system allows routing some truckloads through the RN while the remaining truckloads are dispatched PtP.

We analyze the performance of our models and the solutions obtained for TLRND problems through extensive computational testing. Finally, we conclude with a description of directions for future research.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

Dr. Sarah E. Root

Dissertation Committee:

Dr. Chase Rainwater

Dr. Ed Pohl

Dr. Scott J. Mason

© 2012 by Hector Andres Vergara Arteaga
All Rights Reserved

DISSERTATION DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed _____
Hector Andres Vergara Arteaga

Refused _____
Hector Andres Vergara Arteaga

ACKNOWLEDGMENTS

I would like to sincerely thank the people who have contributed to the completion of this dissertation and others who have made my experience in Arkansas an unforgettable one. First, I want to thank my advisor Dr. Sarah Root for providing guidance and support from the very first day. I greatly appreciate her involvement with my research and the long hours devoted to discussing the many ideas that appear in this dissertation. I am also extremely grateful for the coaching I received from her during my job search. Thank you very much for helping me to achieve even greater goals than those that I had set for myself before this journey started.

I also want to thank my doctoral committee for showing genuine interest about the progress of my research, reading this document and sharing valuable comments about the completed work. In particular, I would like to acknowledge the help I received from Dr. Chase Rainwater whose modeling suggestions proved to be essential to improve the performance of my proposed approach for the research problem studied in this dissertation. Drs. Ed Pohl and Scott Mason are great role models of leadership and professionalism whom I hope to emulate in my future academic career.

My profound gratitude to Gary Whicker and Douglas Mettenburg at J.B. Hunt Transportation Services for their willingness to discuss different aspects of my research and provide real world data for the test cases presented in this dissertation. In the same way, I would like to thank Sergio Maldonado for his hard work completing an empirical analysis of the factors that affect the problem studied in this research for his undergraduate honors thesis and for helping me to process the raw test case data. The insights discovered in his work and his assistance with data processing were really helpful in the process of completing this research work.

In addition, I would like to express my sincere gratitude to all the professors that shared their knowledge and experiences with me during my time at Arkansas. The lessons learned and the continuous encouragement I received from each one of them will always be part of my treasured memories of the time spent here.

I am also grateful for the support provided by the staff in the main office. Karen, Carrie, Sandy and Tamara were always willing to lend a helping hand and make my life easier in so many opportunities. I really appreciate all your hard work.

Finally, all of this would not have been possible without the love, patience and support from Barbara. Thank you very much for your sacrifice and for always being there for me. You really complete me and I am looking forward to a life full of new adventures and challenges next to you.

DEDICATION

This dissertation is dedicated to my parents, sisters, niece and nephew who even from far away share with me the good and the difficult times. I love you all very much.

TABLE OF CONTENTS

| | |
|--|----|
| 1. Introduction | 1 |
| 1.1. Research Motivation | 1 |
| 1.2. Research Contributions | 4 |
| 1.3. Dissertation Overview | 6 |
| References | 9 |
| 2. Feasible Path Enumeration for Directed Acyclic Graphs Limiting Path Length and Nodes Visited | 10 |
| 2.1. Introduction and Problem Description | 10 |
| 2.2. Related Work..... | 13 |
| 2.3. Algorithms for Path Enumeration in Directed Acyclic Graphs | 18 |
| 2.3.1. <i>Notation</i> | 18 |
| 2.3.2. <i>Overview of the Procedure for Feasible Path Enumeration</i> | 19 |
| 2.3.3. <i>Algorithm 1: Network Reduction Algorithm</i> | 20 |
| 2.3.3.1. <i>Network Reduction to a Directed Acyclic Graph with a Unique Sink</i> | 21 |
| 2.3.3.2. <i>Proof of Correctness of Algorithm 1</i> | 22 |
| 2.3.4. <i>Algorithm 2: Counting Paths</i> | 23 |
| 2.3.4.1. <i>Determining the Number of Paths in the Network</i> | 23 |
| 2.3.4.2. <i>Proof of Correctness of Path Counting Algorithm</i> | 25 |
| 2.3.5. <i>Algorithm 3: Enumerating Paths</i> | 27 |
| 2.3.5.1. <i>Path Enumeration with Side Constraints</i> | 27 |
| 2.3.5.2. <i>Complexity of Path Enumeration Algorithm</i> | 29 |
| 2.3.5.3. <i>Proof of Correctness of Path Enumeration Algorithm</i> | 29 |
| 2.4. Computational Results | 33 |
| 2.4.1. <i>Experimental Design</i> | 33 |
| 2.4.2. <i>Random Network Generation</i> | 35 |
| 2.4.3. <i>Results</i> | 36 |
| 2.4.3.1. <i>Effect of the Network Reduction Algorithm on Test Problems</i> | 36 |
| 2.4.3.2. <i>Results for Small-Region Sparse Networks</i> | 36 |

| | | |
|------------|---|-----------|
| 2.4.3.3. | <i>Effect of Increasing the Size of the Service Region</i> | 40 |
| 2.4.3.4. | <i>Effect of Increasing Arc Density</i> | 42 |
| 2.4.3.5. | <i>Relevance of Computational Results</i> | 45 |
| 2.5. | Conclusions and Future Research | 46 |
| References | | 48 |
| 3. | Composite Variable Model for Truckload Relay Network Design | 51 |
| 3.1. | Introduction | 51 |
| 3.2. | Truckload Relay Network Design | 54 |
| 3.2.1. | <i>Problem Definition</i> | 57 |
| 3.3. | Mathematical Model for TLRND | 59 |
| 3.3.1. | <i>Composite Variable Model</i> | 60 |
| 3.3.2. | <i>Generating Composite Variables</i> | 62 |
| 3.4. | Solution Methodology and Computational Experiments | 64 |
| 3.4.1. | <i>Generation of Random Instances and Selection of Parameter Values</i> | 64 |
| 3.4.2. | <i>Exact Solution Method</i> | 66 |
| 3.4.2.1. | <i>Strength of Formulation</i> | 68 |
| 3.4.2.2. | <i>Pricing Method for Solving TLRND</i> | 69 |
| 3.4.3. | <i>Heuristic Method</i> | 72 |
| 3.4.3.1. | <i>Effect of Instance Size</i> | 75 |
| 3.4.3.2. | <i>Effect of Equipment Balance</i> | 77 |
| 3.4.4. | <i>Test Case from a Major TL Carrier</i> | 79 |
| 3.5. | Conclusions and Future Research Directions | 83 |
| References | | 87 |
| 4. | Mixed Fleet Dispatching in Truckload Relay Network Design Optimization | 90 |
| 4.1. | Introduction | 90 |
| 4.2. | Alternative Dispatching Methods in Freight Transportation | 94 |
| 4.2.1. | <i>Relay Networks for TL Transportation</i> | 97 |
| 4.3. | Problem Statement and Mathematical Formulation | 101 |
| 4.3.1. | <i>Truckload Relay Network Design with Mixed Fleet Dispatching (TLRND-MD)</i> | 101 |
| 4.3.2. | <i>Composite Variable Formulation for TLRND-MD</i> | 102 |

| | | |
|------------|--|-----|
| 4.3.2.1. | <i>Notation</i> | 103 |
| 4.3.2.2. | <i>Model Formulation</i> | 104 |
| 4.4. | <i>Solution Methodology</i> | 106 |
| 4.4.1. | <i>Generation of Composite Variables</i> | 106 |
| 4.4.2. | <i>Heuristic Solution Method</i> | 107 |
| 4.5. | <i>Computational Results</i> | 109 |
| 4.5.1. | <i>Experimental Design</i> | 110 |
| 4.5.2. | <i>Random Network Problem Generation</i> | 112 |
| 4.5.3. | <i>Results</i> | 113 |
| 4.5.3.1. | <i>Exact Solution Results</i> | 113 |
| 4.5.3.2. | <i>Heuristic Solution Results</i> | 118 |
| 4.5.3.3. | <i>Sensitivity Analysis for Different Carrier Parameters</i> | 120 |
| 4.5.3.4. | <i>Test Case Results</i> | 131 |
| 4.6. | <i>Conclusions</i> | 133 |
| Appendix A | | 136 |
| References | | 139 |
| 5. | Conclusion | 142 |
| 5.1. | <i>Research Conclusions</i> | 142 |
| 5.2. | <i>Future Research Directions</i> | 148 |

LIST OF FIGURES

| | |
|--|-----|
| Figure 1. Classification of Path Enumeration Problems..... | 14 |
| Figure 2. Illustration of Feasible and Unfeasible Paths in a Network. | 20 |
| Figure 3. Pseudocode for Algorithm 1: Network Reduction Algorithm. | 21 |
| Figure 4. Pseudocode for Algorithm 2: Path Counting Algorithm..... | 24 |
| Figure 5. Pseudocode for Algorithm 3: Path Enumeration with Side Constraints. | 31 |
| Figure 6. Effect of Circuity on Small-Region Sparse Networks (50 Nodes)..... | 38 |
| Figure 7. Effect of Nodes Allowed in a Path on Small-Region Sparse Networks (50 Nodes).... | 39 |
| Figure 8. Effects of Circuity and Nodes Allowed in a Path on Small-Region Sparse Networks. | 39 |
| Figure 9. Relay Network for Truckload Transportation. | 58 |
| Figure 10. Example Template for Composite Generation. | 62 |
| Figure 11. Set of Templates Used for Composite Generation. | 63 |
| Figure 12. Effect of Instance Size on Solution Quality and Model Performance..... | 76 |
| Figure 13. East U.S. Network for a Major TL Carrier..... | 80 |
| Figure 14. Test Case Solutions for Different Fixed Costs of Installation..... | 83 |
| Figure 15. Relay Network for Transportation of Truckloads. | 92 |
| Figure 16. Highly Circuitous Route in a Relay Network for Truckload Transportation..... | 100 |
| Figure 17. Template Types Used for Composite Variable Generation. | 107 |
| Figure 18. Dispatching Method Usage with Respect to Truckload O-D Pair Distance. | 108 |
| Figure 19. Length of Haul Results for Alternative Dispatching Systems (50 Nodes)..... | 117 |
| Figure 20. Service Time Results for Alternative Dispatching Systems (50 Nodes)..... | 117 |
| Figure 21. Differences Between RN-only and Mixed Fleet as PtP Repositioning Cost Varies. | 122 |

| | |
|---|-----|
| Figure 22. Differences Between RN-only and Mixed Fleet as Minimum Volume Required to Open RPs Varies. | 125 |
| Figure 23. Differences Between RN-only and Mixed Fleet as Maximum Imbalance Allowed Varies. | 127 |
| Figure 24. Differences Between RN-only and Mixed Fleet as Fixed Cost of RPs Varies. | 129 |
| Figure 25. Summary of Effects of Design Parameters. | 130 |

LIST OF TABLES

| | |
|--|-----|
| Table 1. Experimental Design for PENVLC. | 33 |
| Table 2. Nodes and Arcs Eliminated by Network Reduction Algorithm (Algorithm 1)..... | 36 |
| Table 3. Computational Results for Small-Region Sparse Networks..... | 37 |
| Table 4. Computational Results for Large-Region Sparse Networks..... | 41 |
| Table 5. States in Solved and Unsolved Instances for Large-Region Dense Networks. | 43 |
| Table 6. Computational Results for Small-Region Dense Networks. | 44 |
| Table 7. Computational Results for Large-Region Dense Networks. | 44 |
| Table 8. Exact Solution Results..... | 67 |
| Table 9. Strength of Formulation of Composite Variable Model for TLRND (50 Nodes)..... | 69 |
| Table 10. Pricing Method Results (50 Node Networks)..... | 72 |
| Table 11. Usage of Composites in Optimal Solutions by Template Type..... | 73 |
| Table 12. Heuristic Method Results (50 Node Networks)..... | 73 |
| Table 13. Results with Equipment Balance Enforced. | 79 |
| Table 14. Heuristic Method Results for East U.S. Network (623 Nodes and 1,386 Truckloads)..... | 82 |
| Table 15. Usage of Composite Variables in Optimal Solutions by Template Type..... | 108 |
| Table 16. Exact Results for 50 Node Networks (Baseline Scenario). | 114 |
| Table 17. Comparison RN-only and PtP-only Systems with Mixed Fleet Dispatching (50 Nodes). | 114 |
| Table 18. Heuristic Results for 50 Node Networks (Baseline Scenario)..... | 118 |
| Table 19. Results for 100 and 150 Node Networks (Baseline Scenario)..... | 119 |
| Table 20. Results for Test Case (623 Nodes and 1,386 O-D Pairs with Truckload Demand). .. | 132 |

LIST OF APPENDIX TABLES

| | |
|---|-----|
| Table A.1. Results for Low PtP Repositioning Cost. | 136 |
| Table A.2. Results for High PtP Repositioning Cost..... | 136 |
| Table A.3. Results for Limited Proportion of PtP Loads..... | 137 |
| Table A.4. Results for Moderate Volume Required to Open a RP..... | 137 |
| Table A.5. Results for High Volume Required to Open a RP. | 138 |
| Table A.6. Results for Some Imbalance Allowed at Nodes for RN Loads. | 138 |
| Table A.7. Results for High Fixed Cost of Installation of RPs..... | 139 |

1. Introduction

In this dissertation, we study a strategic planning problem in truckload transportation – the design of relay networks (RN) – and develop mathematical models and algorithms for its solution. In Section 1.1, we describe the motivation behind our research work. A discussion of the main contributions of this dissertation is presented in Section 1.2. Finally, Section 1.3 presents an overview of the research work included in this document.

1.1. Research Motivation

Trucking is the predominant mode of land transportation of commercial freight in the United States. According to the 2007 Commodity Flow Survey conducted by the U.S. Bureau of Transportation Statistics (2010), trucking accounts for 71.3% of total freight expenditures and 70% of total weight transported annually. The trucking industry is comprised of small and large carriers that provide service using one of two different types of operation: full truckload (TL) and less-than-truckload (LTL) (Campbell, 2005).

TL carriers commonly ship goods using a dispatching method called over-the-road (OTR) or point-to-point (PtP) dispatching in which a single driver is responsible for transporting a load directly between origin and destination. Under this method of operation, drivers who have just delivered a load must subsequently move a different load that originates near the drop off location for the first one in order to minimize empty miles. For example, if a driver delivers a load to Fayetteville, AR, the carrier must either find a load leaving Fayetteville or have the driver travel to a nearby city to pick up another load. Carriers must piece together a series of these trips to return drivers to their home domiciles. Often, finding such backhaul trips in close proximity to a drop off location is a very challenging task due to imbalanced freight networks and the fact that demand arrives predominantly in a random fashion (Campbell, 2005). This type of

operation often leaves drivers on the road for two or three weeks at a time. The time that drivers spend away from their homes is one of the main causes of driver turnover for TL carriers given that most drivers perceive a reduction in their quality of life and quit (Taylor et al., 1999).

The American Transportation Research Institute (2011) reports that driver retention is one of the most significant concerns for the TL industry having ranked in the top three concerns for carriers in five of the last seven years. TL carriers commonly experience an annual turnover of more than 100% (Campbell, 2005). In fact, the shortage of truck drivers during 2011 was estimated at 125,000, exceeding industry forecasts (Morris, 2011). This is very costly since carriers must invest significant resources in recruiting, training, and retaining drivers. Rodriguez et al. (2000) determined that average driver turnover costs are around \$8,200 per driver. Considering the high turnover, the expense for TL carriers is very significant and it has been estimated at around \$2.8 billion annually. Additionally, the shortage of drivers negatively affects level of service to shippers and results in lost income from demand that cannot be satisfied (Keller and Ozment, 1999). As reported by the American Trucking Associations (2007), this problem is actually able to produce negative economic impacts that go far beyond the boundaries of the TL industry given that trucking is not a self-contained industry.

In contrast, LTL carriers move shipments that do not fill up a trailer. These shipments must travel via a series of hubs where freight is sorted and consolidated to better utilize truck capacity. Since freight is moved primarily between hubs, tours for the drivers are more regular which allows them to return home much more frequently. In fact, the turnover rate for LTL carriers is often less than 10% (Campbell, 2005).

Motivated by the differences in driver retention between LTL and TL carriers, an alternative dispatching method using a network of *relay points* (RPs) has been considered for TL

transportation. This is referred to as a *relay network* (RN). In this alternative system, freight is transported from its origin through a series of relay points to its destination. This would essentially emulate LTL operations, except that no freight would be handled at the relay points; these points would only serve as locations in the network for drivers and loads to be exchanged (Campbell, 2005). Given that the transportation of a truckload from origin to destination is divided into several shorter segments between nodes in the network, tour regularity for the drivers is increased given that they now have to visit fixed RPs (Üster and Maheshwari, 2007; Üster and Kewcharoenwong, 2011). In a RN, *local drivers* are in charge of the movements between RPs and origin-destination nodes, while *lane drivers* transport loads between two RPs.

The strategic design of relay networks for TL transportation has to consider a fundamental tradeoff between the costs associated with the relay network and the reduction in driver turnover. A relay network configuration allows drivers to return to their home domiciles more frequently, but TL carriers need to consider the fixed cost of installation of RPs and the incremental cost associated with added miles for the truckloads due to circuitry (Üster and Maheshwari, 2007).

An existing prescriptive approach in the literature for the strategic design of TL relay networks considers an arc-based formulation that is a combination of the multicommodity network flow and hub location problems. This original formulation presented by Üster and Maheshwari (2007) suffers from tractability issues associated with difficult operational constraints. These constraints are eventually relaxed to develop a heuristic approach using a construction heuristic and tabu search. Üster and Kewcharoenwong (2011) overcome some of the challenges and provide an exact algorithm based on an enhanced implementation of Benders' decomposition to solve larger instances of this problem within a predefined optimality tolerance.

However, the authors report in their work that their methodology is only able to obtain solutions for up to 80 node network problems. As observed, there is a need for new prescriptive models for this problem that would allow TL carriers to solve larger problem instances that are representative of practical cases and incorporate more challenging operational constraints such as limitations on circuitry and number of RPs allowed to be visited. In our dissertation research we focus on this specific need.

1.2. Research Contributions

There are several contributions related to the work completed in this dissertation. Most of the previous research on the design of relay networks for TL transportation considers simulation analyses and the development of heuristic and algorithmic procedures. However, the literature is still sparse on prescriptive modeling approaches for this problem.

In this dissertation, we propose a new mathematical formulation for the strategic truckload relay network design (TLRND) problem that minimizes total transportation and installation costs for the RPs while explicitly considering operational constraints on load circuitry, equipment balance, distance limits for local and lane drivers, and the number of RPs that a load visits. Modeling these operational constraints represents a significant challenge. We develop a prescriptive *composite variable model* (CVM) formulation to solve this problem. Depending upon the way in which the composite variables are generated, this model can be used to obtain both exact and heuristic solutions for TLRND.

Although previous research has considered some of these operational constraints in the formulation of the problem, aspects such as equipment balance and especially load circuitry have been either relaxed or enforced through surrogate constraints due to tractability issues. Furthermore, the limitation on the number of RPs allowed to be visited has not been modeled as

part of this problem until now. In our formulation, a composite variable implicitly captures the limitations on the number of RPs that can be visited, local and lane distances, and circuitry over the shortest path distance between origin and destination. Since these constraints are already captured in the variable definition, it is not necessary to include them in the mathematical formulation. In addition, since empty repositioning moves can be modeled as additional variables in our formulation, we are able to enforce perfect balance at the nodes in the transportation network. To the best of our knowledge, this has not been achieved by previous work in this area.

A method to generate composites is required for the CVM. We consider two ways: an exact path enumeration algorithm and a method using predefined templates for the routes. The development of an exact feasible path enumeration algorithm for directed acyclic graphs (DAGs) that represent a TL transportation network provides an alternative approach to the literature on constrained shortest path algorithms. The extensive computational testing with the proposed algorithm allows us to identify which restriction between the number of nodes allowed to be visited and the total length of a path represents a greater computational challenge based on the number of feasible paths that are generated.

Furthermore, we use the composite generation method that uses predefined templates to exercise our model and obtain high-quality solutions in reasonable times to several instances of TLRND that are larger than those solved in existing literature. This allows us to solve realistically-sized problem instances using our approach as demonstrated by the analysis of a test case provided by a major TL carrier that operates nationally.

Another contribution of our proposed formulation is that it is flexible to extensions. Given the definition of our variables, we are able to incorporate generalizations of TLRND into

the model without significantly affecting the structure of the formulation and its tractability. We extend our mathematical formulation to solve the integrated problem of strategically designing the relay network and selecting a dispatching method for truckloads served by a mixed fleet dispatching system that combines PtP and RN shipments. To the best of our knowledge, this is the first prescriptive model for the strategic TL relay network design with mixed fleet dispatching (TLRND-MD) problem. Furthermore, additional operational constraints can be incorporated in the CVM formulation of TLRND in a similar way.

Finally, having prescriptive models for TLRND and TLRND-MD allows us to better quantify the benefit of a hybrid configuration for TL transportation that uses PtP and RN dispatching over a RN-only dispatching system. Previous research attempts to compare these two systems, but those comparisons are made based on non-optimal solutions for TLRND-MD.

1.3. Dissertation Overview

The primary focus of my dissertation research is to study the TLRND problem. However, our research work in this area also includes both a related problem associated with our approach to solve TLRND and an extension to the basic CVM formulation proposed for this problem. The latter is intended to add additional realistic aspects of the problem to the original formulation and be able to evaluate alternative implementations of TL relay networks that are more likely to be applicable in practice.

In Chapter 2, we present the development of an exact method based on reverse topological sorting for the enumeration of all feasible paths that satisfy additional side constraints in directed acyclic graphs. We explicitly consider limitations on the number of nodes visited and total path length. The purpose of developing this algorithm was to generate valid routes (i.e., composite variables) for our proposed CVM formulation of TLRND presented in

Chapter 3. Since our algorithm simultaneously considers both side constraints while enumerating paths between an origin node and a destination node in a graph, it overcomes one of the disadvantages of several constrained shortest path and constrained k -shortest path algorithms. Some implementations of these traditional methods require enumerating a large number of paths and checking those paths for every side constraint sequentially, adding to the computational effort required to solve this problem. Also, our algorithm does not require finding an appropriate value of k that will ensure complete enumeration of feasible paths as it is the case with constrained k -shortest path algorithms. Although we determined that our exact algorithm is very efficient for strict values of the side constraints considered, we also discovered that there are some drawbacks associated with the application of this algorithm for the generation of composites, especially for networks with a large number of nodes (i.e., more than 100 nodes in the network). First, the method needs to consider general networks, not just DAGs. While there are methods to transform a general network with loops into a DAG, they represent additional computational effort and cannot be applied without loss of generality. Second, a new network needs to be constructed for every load considered in an instance of TLRND given that every load has a unique pair of origin and destination nodes. Third, our algorithm is particularly challenged by instances with a large number of feasible paths as previously mentioned. Given these limitations and since we also observed high variability in the number of feasible paths that exist across instances of the same type, we decided to focus on a different approach to generate composite variables as described in Chapter 3.

Chapter 3 presents the CVM formulation for TLRND and describes the development of an enumeration-based algorithm for composite generation using predefined templates. Exact solutions obtained using both standard branch-and-cut as implemented in CPLEX and a

customized pricing method for 50 node networks are included in this chapter. We also describe how this model can be used to obtain heuristic solutions for larger problem instances if only a subset of composite variables is generated to solve the problem. The effects of instance size, equipment balance and fixed cost of installation of the RPs are analyzed with respect to the performance of our solution approach and the characteristics of the solutions obtained using both randomly generated problem instances and test case data from a major TL carrier.

In Chapter 4, we extend our model to study a generalization of TLRND where the TL carrier has the option of dispatching using a mixed fleet. In this case, some of the loads are dispatched PtP while the rest are shipped through the network of relay points. Academic researchers and industry practitioners have suggested that mixed fleet dispatching systems are likely to outperform individual methods for TL transportation. However, since no exact approaches exist to design such mixed fleet dispatching systems, there is not a precise understanding of the actual benefits of such configurations in the TL industry. We use our extended formulation to analyze the effect of several design parameters on the solutions obtained by looking at the resulting relay networks and measuring performance metrics for carriers, drivers and shippers. The design parameters evaluated include repositioning costs applied to PtP loads, a maximum proportion of loads allowed to be sent PtP, a minimum volume required to open a RP, and equipment balance desired for loads routed through the RN. Similar to the work in Chapter 3, the heuristic approach developed in this research to solve TLRND-MD is used to obtain solutions for randomly generated data and a test case provided by a major TL carrier.

Finally, Chapter 5 presents the conclusions drawn from our research work along with suggestions for future research directions.

References

- American Transportation Research Institute. (2011). *Critical Issues in the Trucking Industry - 2011*. Arlington, VA: American Transportation Research Institute (ATRI).
- American Trucking Associations. (2007). *Trucking and the Economy*. Arlington, VA: American Trucking Associations (ATA).
- Campbell, J. (2005). Strategic Network Design for Motor Carriers. In *Logistics Systems - Design and Optimization* (pp. 245-278). New York, NY: Springer-Verlag.
- Keller, S.B., and Ozment, J. (1999). Managing Driver Retention: Effects of the Dispatcher. *Journal of Business Logistics*, 20(2), 97-119.
- Morris, F. (2011). *Thousands of Trucking Jobs, But Few Take the Wheel*. Retrieved from <http://www.npr.org/2011/10/31/141791952/thousands-of-trucking-jobs-but-few-take-the-wheel>.
- Rodriguez, J., Kosir, M., Lantz, B., Griffin, G., and Glatt, J. (2000). *The Costs of Truckload Driver Turnover*. Fargo, ND: Upper Great Plains Transportation Institute.
- Taylor, G., Meinert, T., Killian, R., and Whicker, G. (1999). Development and Analysis of Alternative Dispatching Methods in Truckload Trucking. *Transportation Research Part E*, 35, 191-205.
- U.S. Bureau of Transportation Statistics. (2010). *Transportation: 2007 Commodity Flow Survey*.
- Üster, H., and Kewcharoenwong, P. (2011). Strategic Design and Analysis of a Relay Network in Truckload Transportation. *Transportation Science*, 45(4), 505-523.
- Üster, H., and Maheshwari, N. (2007). Strategic Network Design for Multi-Zone Truckload Shipments. *IIE Transactions*, 39, 177-189.

2. Feasible Path Enumeration for Directed Acyclic Graphs Limiting Path Length and Nodes Visited

Path enumeration can lead to a very large number of paths from which only a subset will result after considering side constraints. In this chapter we develop an efficient algorithm that explicitly considers constraints that limit total path length and the total number of nodes visited when enumerating paths for the case of directed acyclic graphs. Computational results are analyzed, and we highlight insights we obtain about the effect of these constraints and network characteristics on the solutions we find. At the end, conclusions are drawn and presented along with directions for future research.

2.1. Introduction and Problem Description

In many problems, it is not sufficient to generate a single path such as the shortest path between an origin and a destination. Instead, many problems require several paths between two nodes in a network to be generated. For instance, a traveler has many options between an origin airport and a destination airport when booking a flight. To make a final decision about the appropriate ticket to buy, a traveler has to consider several aspects such as the number of connections to make, total time of the itinerary and total cost for the flight. However, the number of possible routes is extremely large when looking at the air transportation network of one of the major carriers in the United States. For this reason, it is necessary to limit the number of possible paths that the traveler will have to consider by restricting the number of airports visited and limiting the total cost of the flight in terms of either time, miles or dollars. The traveler will then select a flight from a reduced set of available options.

In general, the explicit enumeration of paths between an origin node and a destination node in a graph requires significant computational effort given that for most realistic

applications, the size of the network gives rise to an extremely large number of paths that need to be generated. Even in cases such as the flight booking example presented above, the generation of a feasible subset of paths between two nodes in a graph is a challenging problem. In this chapter, we consider a problem that simultaneously limits the length of the paths that can be generated and the number of nodes which can be visited. An approach is developed to efficiently generate the set of feasible paths between an origin-destination (O-D) pair in a directed acyclic graph (DAG). Such problems arise in the traffic assignment problem, the flight booking problem presented above and other route choice/optimization problems. We will refer to this problem as path enumeration with nodes visited and length constraints (PENVLC).

Previous research has developed exact and heuristic algorithmic approaches that incorporate restrictions imposed on the enumerated paths between two nodes in a network (see Park and Rilett, 1997 and Park et al., 2002). A commonly used approach to generate these paths is to first enumerate all paths in the network or a very large number of those paths (e.g., k -shortest paths) and later eliminate the paths that do not satisfy the side constraints (van der Zijpp and Fiorenzo Catalano, 2005). This method is impractical for many realistically sized networks, and is particularly inefficient when very few of the paths satisfy the side constraints. In those cases, the number of paths to be eliminated significantly exceeds the number of paths to be kept. Furthermore, such algorithms require computational effort not only to generate the set of all paths, but to check which of these paths satisfy the side constraints. This leads to a procedure that is inefficient (van der Zijpp and Fiorenzo Catalano, 2005). In Section 2.2, we review how constrained shortest path, k -shortest paths, constrained k -shortest paths algorithms and other heuristic approaches have been used for problems related to PENVLC.

In our research, we develop an exact algorithmic approach based on topological sorting that simultaneously limits both the number of nodes visited in a path and the total path distance while enumerating all feasible paths between an O-D pair of nodes in a DAG. A DAG is formed by a collection of connected nodes and arcs such that there is no path that starts at any node i and follows a sequence of arcs that eventually returns to node i (Ahuja et al., 1993). Many real-world problems are not initially represented as DAGs. However, since many applications such as emergency service vehicle routing require alternative paths that avoid visiting a node more than once while moving from origin to destination in the same direction (e.g., north-to-south), a DAG can be used to represent the underlying transportation network. We can transform any directed graph with cycles into a DAG by applying a method such as the heuristic approach developed by Filippovich (1973) which eliminates a minimal number of arcs in the network to avoid cycles. If this method is applied to preprocess the networks we consider, we can then apply our proposed approach to enumerate feasible paths for any network.

Although PENVLC arises in several transportation and communication problems, the primary motivation to study this problem is the routing of loads for a full truckload (TL) carrier that uses a relay network for dispatching (see Taylor et al., 2001, Üster and Maheshwari, 2007 and Üster and Kewcharoenwong, 2011). These TL relay networks essentially divide a load's transportation from O-D into several shorter segments between nodes (i.e., relay points) in the network. At each relay point, drivers or trailers are exchanged and the load continues its transport. This helps to increase the frequency with which drivers can return to their homes given an increased regularity in their routes. This network design and configuration has the potential to reduce driver turnover, as time away from home is cited as the primary reason for the significant turnover rate of over 100% among TL carriers (Taylor et al., 2001). Operating a relay

network requires planners to determine which paths the loads should take through the network. A significant added cost of operating on a relay network is the added circuitry for the loads since they are no longer dispatched directly from origin to destination using point-to-point (PtP) dispatching. For this reason, it is important to limit the distance of the paths that are considered – that is, the paths should not exceed a limit on circuitry relative to the direct O-D distance. In addition, it is important that the route a load takes does not have too many intermediate stops, as this can lead to excessive time to transport the load. This requires us to place a limit on the total number of nodes in a path.

Other real world applications where this algorithmic approach can be used include network interdiction problems and optimal route diversion problems after disruptions. The algorithmic approach presented in this chapter contributes to the literature of path enumeration algorithms with a general method that does not rely on iteratively finding shortest paths and which can be adapted to practical applications where determining alternative feasible paths is required.

The remainder of the chapter is organized as follows. In Section 2.2, we present a literature review on path enumeration. The algorithms developed in this research are presented in Section 2.3. In Section 2.4, we test our algorithms and present computational results. Finally, conclusions and future research are presented in Section 2.5.

2.2. Related Work

Most of the previous research on path enumeration has focused on developing methods for the generation of k -shortest paths, but some work has also considered the constrained k -shortest paths and constrained shortest path problems as illustrated in Figure 1. The two most common approaches used for path enumeration are labeling algorithms and heuristics for arc

penalty or arc elimination. Heuristic methods are generally preferred over exact methods in operational problems such as dynamic in-vehicle routing and other choice path problems where a smaller set of alternative paths is desired in a shorter period of time (see Park and Rilett, 1997 and van der Zijpp and Fiorenzo Catalano, 2005).

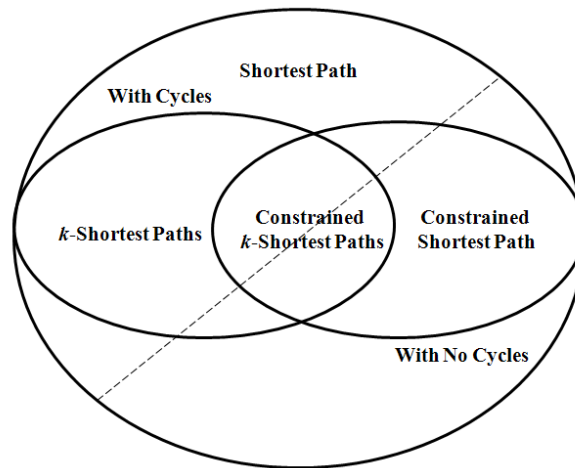


Figure 1. Classification of Path Enumeration Problems.

The k -shortest paths (KSP) problem is a generalization of the shortest path problem that enumerates and selects the k paths with minimum total length that connect a given pair of nodes in a network. It was originally presented by Hoffman and Pavley (1959). Two types of k -shortest paths problems have been studied: one where cycles or loops are allowed in the generated paths and the other which finds paths where cycles or loops do not exist (Yen, 1971).

Dreyfus (1969) introduced one of the early methods to solve the k -shortest paths problem in a network with cycles allowed. This procedure computes the k -shortest paths that exist between an origin node and each of the remaining nodes in the network using a dynamic programming approach. However, the procedure requires generating a large set of shortest paths and then selecting among them by applying side constraints. Eppstein (1998) developed another approach that uses a breadth first search algorithm to enumerate paths starting at a given origin

node and ending anywhere in the network. Eppstein's approach can also be used to enumerate all paths that are shorter than a stated distance threshold but it does not explicitly address a limitation on the number of nodes visited by those paths. The main drawback to these approaches is that the paths generated are allowed to have cycles; this is undesirable for routing problems such as the TL relay network design problem. In addition, determining the appropriate value of k for an O-D pair is difficult since this number has to ensure that all paths that satisfy the side constraints are included. In the remainder of this section we focus on work that considers the generation of loopless paths.

Yen (1971) was the first to present an efficient algorithm for the more complex problem of finding k -shortest simple paths (i.e., paths without cycles). At the time this algorithm was developed, it was a very significant contribution because of its efficiency as compared to existing approaches. Many of the original approaches considered enumerating all possible paths or a considerable number of paths from the origin to the destination node and then sorting them to obtain the k paths with the shortest distances. Such approaches are highly inefficient especially when the size of the network is large and the value of k is relatively small (Yen, 1971). The algorithm developed by Yen (1971) iteratively finds new paths by analyzing small deviations to paths that have already been determined. As such, the new paths generated have more than one node in common with a previously generated path. The remaining nodes are determined by finding the shortest path deviation to the destination node that visits at least a node in the network that is different from the ones already included in the original path. No side constraints are taken into account when generating the k -shortest paths with this iterative procedure. Some modifications and extensions to classical algorithms for KSP have been studied in the path enumeration literature such as the work of Katoh et al. (1982), and Hadjiconstantinou and

Christofides (1999) with undirected graphs; and Martins and Pascoal (2003) and Carlyle and Wood (2005) with directed graphs. More recently, other algorithms based on replacing paths while iteratively generating the k -shortest simple paths in a network have been presented by Hershberger et al. (2007).

There are a few difficulties associated with the application of k -shortest paths algorithms for path enumeration when considering one or more side constraints. First, similar to complete enumeration, it is necessary to filter the set of k -shortest paths based on the side constraint(s), increasing the computational effort required to obtain the set of feasible paths. Second, it is difficult to set the value of k for an O-D pair in order to enumerate all feasible paths. Finally, the application of regular k -shortest paths algorithms typically results in the generation of paths that are very similar, which is not always desired when developing alternative routes for choice path set problems such as the emergency vehicle routing problem or in-vehicle dynamic routing (see Park and Rilett, 1997 and van der Zijpp and Fiorenzo Catalano, 2005). To overcome the difficulties associated with the use of regular k -shortest paths algorithms, Park and Rilett (1997) present a heuristic approach and Park et al. (2002) develop an exact labeling method for the generation of k -reasonable paths for the emergency vehicle routing problem (i.e. k paths that satisfy more than one side constraint and are as dissimilar as possible).

More recently van der Zijpp and Fiorenzo Catalano (2005) proposed a method that directly generates feasible paths while considering side constraints. They developed an algorithmic approach to enumerate k -shortest paths while imposing restrictions that avoid generating overly circuitous paths or paths that overlap considerably. This proposed approach was developed based on Lawler's approach (Lawler, 1972 and Lawler, 1976) of finding a unique shortest path between an O-D pair in the network and then partitioning the set of all remaining

paths into subsets so that the shortest path in each subset can be determined. From the shortest paths generated, the best one is chosen as the second shortest path between the O-D pair of interest. From then on, the set where the shortest path was found is further partitioned and the procedure is repeated considering the previous subsets. While the work of van der Zijpp and Fiorenzo Catalano (2005) provides an approach that is appropriate for constrained k -shortest paths (CKSP) by pruning paths that do not satisfy simultaneous side constraints, the issue related to establishing an appropriate value for k remains. This method is efficient but does not ensure the generation of the complete set of feasible paths given that number of paths is limited by the value of k . Other CKSP algorithms have been proposed by Liu and Ramakrishnan (2001) for a communications problem and Shi (2010) for the design of an automated storage/retrieval system.

When $k = 1$ and at least one additional side constraint is imposed, the problem is known as the constrained shortest path (CSP) problem. Handler and Zhang (1980) use Lagrangian relaxation to propose a dual algorithm to reduce the value of k when a k -shortest path algorithm is used for solving this problem. Aneja et al. (1983) formulate this problem as a special case of the minimum cost-flow problem with side constraints and present an implicit enumeration approach for its solution. More recently, Carlyle et al. (2008) propose a method based on Lagrangian relaxation and near-shortest path enumeration to obtain the shortest path in a directed graph subject to one or more path-weight constraints. Although these methods present alternative solution methods for imposing side constraints, they only provide the shortest path that satisfies those constraints. In our research problem, we must obtain the complete set of feasible paths that satisfies one or more side constraints.

In the next section, we present the algorithmic approach proposed in our research for explicitly considering more than one side constraint simultaneously while generating all feasible

paths between an O-D pair in a directed acyclic graph. The proposed approach is different from previous research given that there is no need for determining a value for k and it is not required to iteratively compute shortest paths while enumerating feasible paths. This represents an advantage in terms of having an efficient exact approach for the solution of PENVLC. In a general sense, our approach fits in the literature of constrained k -shortest paths when no cycles are allowed and the value of k is unknown.

2.3. Algorithms for Path Enumeration in Directed Acyclic Graphs

2.3.1. Notation

We first introduce the notation that is used throughout the presentation of the algorithms developed in this research to solve PENVLC.

Let:

N = set of nodes in the network,

A = set of arcs in the network,

t = destination node,

s = origin node,

d_{ij} = length of the arc between nodes i and j ,

n = maximum number of nodes allowed to be visited in a path from s to t including s and t ,

β = percentage circuitry allowed with respect to shortest path from s to t ,

D = maximum total length (distance) allowed for paths from s to t ,

P = set of permanent nodes,

T = set of pending nodes,

$\text{indegree}(i)$ = indegree for node i ,

$\text{outdegree}(i)$ = outdegree for node i ,

$\text{order}(i)$ = topological order for node i ,

l_i = # of possible paths to destination node t for node i ,

SP_i = shortest path from node s to node i ,

LP_i = longest path from node s to t .

2.3.2. Overview of the Procedure for Feasible Path Enumeration

In this chapter, we present an exact procedure for PENVLC. We explicitly consider a limit on the number of nodes that can be visited and a maximum total length for the enumerated paths.

In Figure 2, we illustrate some of the paths that exist in a directed acyclic graph with nine nodes. In this example, the limit on the number of nodes that can be visited is set as $n = 4$ (i.e., two nodes in addition to the origin and destination) and the maximum percent of allowed circuitry is $\beta = 0.2$. If the shortest path distance between origin node 1 and destination node 9 is 10 units (path 1-4-7-8-9), then we are interested in enumerating all those paths between nodes 1 and 9 that visit up to 4 nodes and have a path length less than or equal to 12 units.

As observed in Figure 2, Path A formed by nodes 1-5-9 satisfies both side constraints and therefore is a valid path that needs to be enumerated. On the other hand, Path B with nodes 1-4-5-8-9 satisfies only the maximum path length constraint since its distance is 12, but violates the limit on the maximum number of nodes that can be visited since five nodes are visited. As a result, this is not a valid path and it should not be enumerated. Path C formed by nodes 1-2-6-9 only satisfies the maximum number of nodes visited constraint but violates the maximum path length allowed, and therefore should not be enumerated since it is not a valid path.

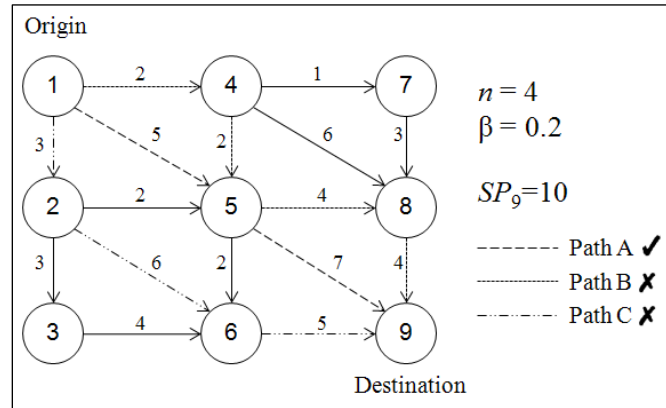


Figure 2. Illustration of Feasible and Unfeasible Paths in a Network.

The procedure for path enumeration requires applying two algorithms in sequence. First, a network reduction algorithm (Algorithm 1) reduces any DAG G with multiple sinks to a DAG G' with a unique sink node t . We apply this since we are concerned only in generating paths from s to t ; all other destination nodes (i.e., nodes with outdegree = 0) can be eliminated without loss of generality. This is helpful in reducing the size of the networks we consider and the computational effort required to solve PENVLC. The resulting network G' is then used to generate feasible paths between the origin s and t using a path enumeration algorithm (Algorithm 3). Algorithm 1 is presented in Section 2.3.3 and Algorithm 3 is presented in Section 2.3.5. Algorithm 2 presented in Section 2.3.4 is necessary to determine the number of paths that satisfy the side constraints and is used to prove the correctness of Algorithm 3.

2.3.3. Algorithm 1: Network Reduction Algorithm

Directed acyclic graphs can have multiple sink nodes (i.e., nodes with no outgoing arcs). However, in PENVLC we consider only a single destination node t . As such, we can eliminate all other sinks and reduce the original network $G=(N,A)$ to a graph $G'=(N',A')$ to enumerate valid paths more efficiently. Algorithm 1 eliminates nodes and arcs that will never be visited on a path from s to t .

2.3.3.1. Network Reduction to a Directed Acyclic Graph with a Unique Sink

The algorithm to obtain G' iteratively eliminates nodes with no outgoing arcs and all the incoming arcs to those eliminated nodes until the graph is left with t as the only sink node. Our algorithm relies on a topological ordering of the nodes in G . A topological ordering is obtained when every node i in G is given a label $\text{order}(i)$ such that for every arc (i,j) in the set of arcs, $\text{order}(i) < \text{order}(j)$ (Ahuja et al., 1993). We are guaranteed to find such an ordering of nodes since a topological ordering of nodes i in G exists if and only if G is a DAG (Ahuja et al., 1993). This is used in Algorithm 1, a search algorithm that uses reverse topological ordering of the DAG to eliminate nodes that will not have a path to destination t . The proposed algorithm is presented in Figure 3.

```

Algorithm 1:
1  begin
2    initialize  $N' := N; A' := A;$ 
3    let  $T := \emptyset;$ 
4    for all  $i \in N'$  do  $\text{outdegree}(i) := 0;$ 
5    for all  $(i,j) \in A'$  do  $\text{outdegree}(i) := \text{outdegree}(i) + 1;$ 
6    for all  $i \in N'$  do
7      begin
8        if  $\text{outdegree}(i) = 0$  then  $T := T \cup \{i\};$ 
9      end;
10   while  $T \neq \emptyset$  do
11     begin
12       select  $i \in T;$ 
13       if  $i \neq t$  then
14          $T := T \setminus \{i\};$ 
15          $N' := N' \setminus \{i\};$ 
16         for all  $(j,i) \in A'$  do
17           begin
18              $A' := A' \setminus (j,i);$ 
19              $\text{outdegree}(j) := \text{outdegree}(j) - 1;$ 
20             if  $\text{outdegree}(j) = 0$  then  $T := T \cup \{j\};$ 
21           end;
22         else
23            $T := T \setminus \{i\};$ 
24       end;
25   end;

```

Figure 3. Pseudocode for Algorithm 1: Network Reduction Algorithm.

The algorithm starts by computing the outdegree (number of outgoing arcs) for each node in the network (see lines 4-5 in Figure 3). Then, it adds all nodes with outdegree equal zero to the set of pending nodes T which is composed of all of the sinks in the network (see lines 6-9 in Figure 3). At every iteration of the algorithm, we select a node i in the set of pending nodes and if it is different than the destination node t , we proceed to eliminate it from the graph along with all of its incoming arcs $(j,i) \in A'$ (see lines 12-18 in Figure 3). At the same time, the outdegree of every node j is updated to reflect the elimination of one of its outgoing arcs (see line 19 in Figure 3). If the outdegree of a node j reduces to zero (i.e., node j becomes a new sink), node j is included in the set of pending nodes T (see line 20 in Figure 3). The algorithm stops when the set of pending nodes is empty, and results in a network G' where node t is the unique sink.

2.3.3.2. Proof of Correctness of Algorithm 1

To prove the correctness of the proposed Algorithm 1 for network reduction, we develop the following Proposition 1. Also, Theorem 1 establishes that no paths from node s to t are eliminated by the network reduction algorithm. Since no paths from s to t are eliminated by Algorithm 1, the resulting network can be used for the efficient enumeration of feasible paths in a further step of our proposed procedure.

Proposition 1: Given a destination t in a directed acyclic graph $G=(N,A)$, G can be reduced to a directed acyclic graph $G'=(N',A')$ where t is the only node $i \in N'$ with $\text{outdegree}(i) = 0$.

Proof: Since G is a DAG, it does not contain any arcs (j,i) that produce cycles. G' is formed by a subset of the arcs included in G , and is therefore acyclic as well. Also, since by construction we iteratively eliminate all nodes $j \in N'$: $j \neq t$ that have $\text{outdegree}(j) = 0$ (see lines 11-24 in Figure 3), G' has a unique sink t with $\text{outdegree}(t) = 0$. Therefore G' is a DAG with a unique sink t .

Theorem 1: The reduction of $G=(N,A)$ to $G'=(N',A')$ does not eliminate any paths from s to t .

Proof: Deleting all arcs $(j,i) \in A'$ each time node i is eliminated from the graph reduces $\text{outdegree}(j)$ by one for all such nodes j . Since a node j with an outgoing arc $(j,t) \in A'$ has $\text{outdegree}(j) > 0$, it will never be a candidate for elimination in the network reduction algorithm (Algorithm 1). Therefore, none of the arcs $(k,j) \in A'$ will be eliminated from the graph since node j remains part of the network during all iterations. Next, assume (p,q) is on a path from s to t and that node q will never be eliminated from G' . Since $\text{outdegree}(q) > 0$ at all iterations, arc (p,q) will never be eliminated from G' and $\text{outdegree}(p) > 0$ during all iterations. Therefore, by induction none of the arcs (p,q) on paths to the destination t will be removed and no paths between the origin s and the destination t will be eliminated. The resulting graph G' contains all paths between the origin s and the destination t .

2.3.4. Algorithm 2: Counting Paths

The second algorithm proposed in this research is not part of the procedure for path enumeration, but is necessary to determine the number of paths that exist in the reduced graph G' between s and t . This is important in order to prove the correctness of the Algorithm 3 for the enumeration of feasible paths in the network.

2.3.4.1. Determining the Number of Paths in the Network

The proposed algorithm to count the number of paths from any node to the destination t in G' uses a search algorithm to iteratively update a label l_i that is the number of possible paths from a given node i to node t . Similar to Algorithm 1, this algorithm uses a reverse topological ordering to determine the order in which nodes that have a path to the destination node t are visited in order to update their corresponding label.

The algorithm presented in Figure 4, starts by computing the outdegree for each node in the reduced network G' and initializes the label $l_t = 1$ (see lines 4-5 and 7 in Figure 4). Since there is only one possible way to get to node t from t , node t is assigned to the set of pending nodes T (see line 6 in Figure 4). Since t is the only sink node that exists in the reduced network, all other nodes have at least one path to t . At every iteration of the algorithm, we select a pending node i and update the label l_i that represents the number of paths that exist between node i and the destination t (see lines 10-13 in Figure 4). Next, we proceed to eliminate node i from the graph along with all of its incoming arcs $(j,i) \in A''$, where A'' is the set of remaining arcs in the reduced network G'' (see lines 14-16 in Figure 4). The outdegree of every node j is updated to reflect the elimination of one of its outgoing arcs (see line 17 in Figure 4). If the outdegree of a node j reduces to zero, node j is included in the set of pending nodes T (see line 18 in Figure 4). The algorithm stops when the set of pending nodes is empty.

| | |
|---------------------|---|
| <i>Algorithm 2:</i> | |
| 1 | begin |
| 2 | initialize $N'' := N'$; $A'' := A'$; |
| 3 | let $T := \emptyset$; $P := \emptyset$; |
| 4 | for all $i \in N''$ do $l_i := 0$; $\text{outdegree}(i) := 0$; |
| 5 | for all $(i,j) \in A''$ do $\text{outdegree}(i) := \text{outdegree}(i) + 1$; |
| 6 | let $T := T \cup \{t\}$; |
| 7 | let $l_t := 1$; |
| 8 | while $T \neq \emptyset$ do |
| 9 | begin |
| 10 | select $i \in T$; |
| 11 | let $P := P \cup \{i\}$; $T := T \setminus \{i\}$; |
| 12 | $N'' := N'' \setminus \{i\}$; |
| 13 | if $i \neq t$ then $l_i := \sum_{k:(i,k) \in A''} l_k$; |
| 14 | for all $(j,i) \in A''$ do |
| 15 | begin |
| 16 | $A'' := A'' \setminus (j,i)$; |
| 17 | $\text{outdegree}(j) := \text{outdegree}(j) - 1$; |
| 18 | if $\text{outdegree}(j) = 0$ then $T := T \cup \{j\}$; |
| 19 | end; |
| 20 | end; |
| 21 | end; |

Figure 4. Pseudocode for Algorithm 2: Path Counting Algorithm.

2.3.4.2. Proof of Correctness of Path Counting Algorithm

We prove the correctness for Algorithm 2 using Propositions 2 to 4 and Theorem 2 that are presented below.

Proposition 2: In a directed acyclic graph G' where t is the only node $i \in N'$ with $\text{outdegree}(i) = 0$, there is always at least one node j with $(j,t) \in A'$ such that $\text{outdegree}(j) = 1$.

Proof: During topological ordering of the nodes in a directed acyclic graph G' , whenever a node i is assigned a topological order label at a given iteration, node i has only outgoing arcs (i,k) that point to unassigned nodes k that will be given higher topological order labels in subsequent iterations. We know that t will be the last node assigned a topological order label since it is the only node with outdegree equal to zero. Therefore, the second to last node j included in the topologically ordered list has only one outgoing arc (j,t) that points to the destination t . As such, outdegree of the second to last node is always 1.

Proposition 3: For a directed acyclic graph G' , while $|N \setminus \{P \cup T\}| > 1$, $|T| \geq 1$.

Proof: By induction. From *Proposition 2*, there is at least one node j with $(j,t) \in A'$ that has $\text{outdegree}(j) = 1$. Since at the start of the procedure $P = \emptyset$ and $T = \{t\}$, node t is moved from T to P and $\text{outdegree}(j)$ is reduced by one. Since now $\text{outdegree}(j) = 0$ and $|N \setminus \{P \cup T\}| = |N| - 1$, node j is added to the set of pending nodes T and $|T| = 1$.

Suppose after n iterations that $|T| \neq 0$ and $|N \setminus \{P \cup T\}| > 2$. Three cases exist. In case 1, $|T| > 1$ and no node $k \in N \setminus \{P \cup T\}$ with $\text{outdegree}(k) = 0$ exists. In this case, at the end of iteration $n+1$ an element of T is removed and added to set P which implies $|N \setminus \{P \cup T\}| > 2$ and $|T| \geq 1$. In case 2, $|T| > 1$ and at least one node $k \in N \setminus \{P \cup T\}$ with $\text{outdegree}(k) = 0$ exists. In this case, at the end of iteration $n+1$ one element of T is removed and at least one new element is added to T which implies $|N \setminus \{P \cup T\}| > 1$ and $|T| > 1$. In case 3, $|T| = 1$. In this case, when

node i with $\text{outdegree}(i) = 0$ is the only element in the set of pending nodes T , $G''=(N'',A'')$ is a DAG with a unique sink i . From *Proposition 2* we know that since $\text{outdegree}(i) = 0$, there exists at least one node j with $(j,i) \in A''$ that has $\text{outdegree}(j) = 1$. Therefore, at the end of iteration $n+1$, $\text{outdegree}(j) = 0$ and j is included in T which implies $|N \setminus \{P \cup T\}| > 1$ and $|T| \geq 1$.

Proposition 4: For a directed acyclic graph G' , Algorithm 2 terminates in a finite number of steps.

Proof: From *Proposition 1*, since there are no cycles in the network G' , each node i will be added to the set of pending nodes T only once when $\text{outdegree}(i) = 0$. Since there are a finite number of nodes in the network, the procedure will terminate in a finite number of steps.

Theorem 2: At the end of Algorithm 2, the value of the label l_i for each node i in G' equals the number of possible paths from node i to destination node t .

Proof: By induction. For any node i with $(i,t) \in A'$ such that $\text{outdegree}(i) = 1$, there is only one possible path from node i to destination node t . As such, $l_i = 1$ for all such nodes i . Suppose at iteration n , we know the set of permanent nodes P and pending nodes T . For each node $j \in P$, we know the values of l_j , which are the number of paths from each permanent node j to the destination node t . Next, suppose that at iteration $n+1$, we arbitrarily select a node $k \in T$. Since $k \in T$, all arcs (k,j) are connected to nodes $j \in P$. The only possible ways to get to the destination t from node k must be on these outgoing arcs (k,j) . As such, each path from node k to node t must immediately visit one of the permanent nodes $j \in P$. Since there is only one possible way to visit a permanent node j from k through the arc (k,j) , and given that we know the number of paths from a node j to t is l_j , the number of paths from node k to t that contain (k,j) is l_j . Since we know all outgoing arcs $(k,j) \in A'$, the total number of paths from node k to destination node t is given by $l_k = \sum_{j:(k,j) \in A'} l_j$.

2.3.5. Algorithm 3: Enumerating Paths

The second step of our proposed methodology is the application of an algorithm that simultaneously considers limits on the distance traveled and the number of nodes visited while generating arrays of labels at each node. These arrays of labels called states will later be used to construct the set of all feasible paths between s and t .

2.3.5.1. Path Enumeration with Side Constraints

Algorithm 3 is presented in Figure 5 and considers a reverse topological ordering of the nodes in the reduced DAG obtained from Algorithm 1 to iteratively generate arrays of labels called *states* for every node in the graph. Each of these states k contains information to build a path from node i to destination t and is only included in a set K_i if the algorithm determines that such state corresponds to a feasible path in the network (i.e., a path that satisfies both side constraints). The set of states K_s generated at the origin s is used to construct the set of all feasible paths between s and t . A significant difference between our proposed algorithm and most of the existing work in the literature of k -shortest paths and constrained k -shortest paths is that it is not necessary to compute a shortest path at each iteration of the algorithm.

The algorithm begins by computing the indegree and outdegree for each node in the network (see lines 2-3 in Figure 5) so that a topological ordering of the nodes in graph G' can be determined (see lines 4-18 in Figure 5). This ordering of the nodes in G' is used to obtain shortest paths between origin s and every other node in the network when all arc costs are temporarily set to 1. This allows us to create a label called “degrees of separation” for each node i (denoted by δ_i) which represents the minimum number of nodes needed to reach node i from the origin s (see lines 19-24 in Figure 5).

After completing a basic feasibility check of the problem using the degrees of separation information for node t , the shortest path between origin s and destination t is obtained (see lines 25- 31 in Figure 5). The length of the shortest path is then used to establish a bound on the maximum total length of paths, D , by considering the percentage of circuitry β allowed on paths from s to t . Also, Algorithm 3 determines the longest path distance that exists between the origin s and the destination t (LP_t) (see lines 33-38 in Figure 5). When appropriate, this latter value is used to check the feasibility of the paths generated and to determine the lengths of the resulting feasible paths.

States k are comprised of a counter k , a label p_{ik} (path slack) that represents the remaining distance that can be traveled on a path from the origin to node i to complete the path represented by state k , a label v_{ik} that represents the number of nodes visited on a path from node i to t represented by state k , and a label u_{ik} that represents the immediate successor node on the path from i to t represented by state k (see lines 39-73 in Figure 5). At every iteration of the search procedure, we select a node i in the set of pending nodes and recursively generate states for that node based on existing state information for successor nodes that are already in the set of permanent nodes P . Both side constraints are enforced to determine whether a recently generated state should be eliminated or included in the set of valid states at node i denoted by K_i . We accept a new state at the current node i if the number of nodes visited and the remaining path slack p_{ik} still allow a feasible path from origin to destination. After states are generated, we update the outdegrees of the affected nodes, and update the set of pending nodes T . The generation of states stops when the set of pending nodes is empty after creating the set of states for node s denoted K_s .

In the last portion of the algorithm (see lines 74-90 in Figure 5), the set of states K_s is used to construct the feasible paths in the network between s and t . For every state k , the length of the path is determined using the path slack information (p_{sk}) included as one of the fields of the state. Successor node (u_{ik}) and number of nodes visited (v_{ik}) information contained in each state k is later used to determine the sequence of nodes i visited in a given path from node s to t .

2.3.5.2. Complexity of Path Enumeration Algorithm

Recall that Algorithm 3 has three main sections: topological ordering, generating states, and reconstructing paths using state information. The complexity of topological ordering is $O(|A|)$. To generate states, the complexity is $O(|A||K|)$. Since $|K|$ is not known in advance, we rely on Algorithm 2 to obtain an upper bound on $|K|$, $|N||A|$. We therefore conclude that the generation of states requires $O(|N||A|^2)$. Finally, the section of Algorithm 3 that builds feasible paths from information obtained from valid states requires $O(|K|^n)$. This is equivalent to $O(|N|^n|A|^n)$. The term $O(|N||A|^2)$ will never be higher than $O(|N|^n|A|^n)$ given that in our motivating problem we will always have $n > 2$ since for every load there is an origin and a destination, making the value of n at least 2. As a result the computation complexity for Algorithm 3 is given by the portion of the code with the highest computational complexity: $O(|N|^n|A|^n)$. Although the worst case is an exponential time for the execution of our algorithm, in practice values of n will not be very high in order to avoid excessive handling of the loads, especially when the distance between origin and destination is not very large.

2.3.5.3. Proof of Correctness of Path Enumeration Algorithm

Theorem 3 and Propositions 5 through 8 are presented below. Together they are used to prove the correctness of Algorithm 3 presented above.

Theorem 3: Algorithm 3 enumerates all paths from origin node s to destination node t in a directed acyclic graph G' when side constraints on the number of nodes visited and the total length of the path are not enforced.

Proof: For any node $i \in N'$ with $(i,t) \in A'$ such that $\text{outdegree}(i) = 1$, there is only one possible path from node i to the destination node t . As such, only one state k is created for each node i that represents the path from node i to t . Furthermore $|K_i| = l_i = 1$.

Suppose at iteration n , we know the set of permanent nodes P and pending nodes T . For each node $i \in P$, we know the values of l_i , which are the number of paths from each permanent node i to the destination node t , and we have also generated l_i states $k \in K_i$. Each of these states k represents a path from i to t and is included in set K_i . Next, suppose that at iteration $n+1$, we arbitrarily select a node $j \in T$. Since $j \in T$, all arcs $(j,i) \in A'$ are connected to nodes $i \in P$. The only possible ways to get to the destination t from node j is on these outgoing arcs (j,i) . As such, each path from node j to node t must immediately visit one of the permanent nodes i and follow one of the feasible paths represented by a state $k \in K_i$. In this way, one new state k' is created at node j which is derived from an existing state k at node i . Therefore, since side constraints on the number of nodes visited and the total length of the path are not enforced, for each outgoing arc $(j,i) \in A'$ there are l_i new states created at node j . Since we know all outgoing arcs $(j,i) \in A'$ and the sets of states K_i at each node i , by induction we know that the states k created at node j represent the set of all paths from node j to t . Furthermore, $|K_j| = l_j = \sum_{i:(j,i) \in A'} l_i$.

The procedure is repeated until node s is made permanent and set K_s is generated. Since $|K_s| = l_s = \sum_{i:(s,i) \in A'} l_i$, the elements of the set K_s represent all paths from origin node s to destination node t in a directed acyclic graph G' when side constraints on the number of nodes visited and the total length of the path are not enforced.

| | | |
|---|----|--|
| <i>Algorithm 3:</i> | 46 | while $T \neq \emptyset$ do |
| 1 begin | 47 | begin |
| 2 for all $i \in N'$ do $\text{indegree}(i) := 0$; $\text{outdegree}(i) := 0$; $\text{order}(i) := 0$; $\delta_i := \infty$; $SP_i := \infty$; $lp_i := 0$; | 48 | select $i \in T$; |
| 3 for all $(i,j) \in A'$ do $\text{indegree}(j) := \text{indegree}(j) + 1$; $\text{outdegree}(i) := \text{outdegree}(i) + 1$; $d'_{ij} := 1$; | 49 | let $P := P \cup \{i\}$; $T := T \setminus \{i\}$; |
| 4 let $T := \emptyset$; $\text{next} := 0$; | 50 | if $i \neq t$ then |
| 5 for all $i \in N'$ do | 51 | for all $(i,j) \in A'$ do |
| 6 if $\text{indegree}(i) = 0$ then $T := T \cup \{i\}$; | 52 | begin |
| 7 while $T \neq \emptyset$ do | 53 | for all $k \in K_j$ do |
| 8 begin | 54 | begin |
| 9 select $i \in T$; | 55 | let $p' := p_{jk}$; |
| 10 let $T := T \setminus \{i\}$; | 56 | let $v' := v_{jk}$; |
| 11 $\text{next} := \text{next} + 1$; | 57 | $k := k + 1$; |
| 12 $\text{order}(i) := \text{next}$; | 58 | $p_{ik} := p' - d_{ij}$; |
| 13 for all $(i,j) \in A'$ do | 59 | $v_{ik} := v' + 1$; |
| 14 begin | 60 | $u_{ik} := j$; |
| 15 $\text{indegree}(j) := \text{indegree}(j) - 1$; | 61 | let $k := \{k, p_{ik}, v_{ik}, u_{ik}\}$; |
| 16 if $\text{indegree}(j) = 0$ then $T := T \cup \{j\}$; | 62 | if $v_{ik} + \delta_i > n$ or $p_{ik} < 0$ then |
| 17 end ; | 63 | $E := E \cup \{k\}$; |
| 18 end ; | 64 | else |
| 19 let $\delta_s := 0$; | 65 | $K_i := K_i \cup \{k\}$; |
| 20 for $r = 1$ to $ N' $ do | 66 | end ; |
| 21 begin | 67 | end ; |
| 22 select $i \in N' : \text{order}(i) = r$; | 68 | for all $(j,i) \in A'$ do |
| 23 for all $(i,j) \in A'$ do $\delta_j := \min\{\delta_j, \delta_i + d'_{ij}\}$; | 69 | begin |
| 24 end ; | 70 | $\text{outdegree}(j) := \text{outdegree}(j) - 1$; |
| 25 if $\delta_t < n$ then | 71 | if $\text{outdegree}(j) = 0$ then $T := T \cup \{j\}$; |
| 26 let $SP_s := 0$; | 72 | end ; |
| 27 for $r = 1$ to $ N' $ do | 73 | end ; |
| 28 begin | 74 | for all $k \in K_i$ do |
| 29 select $i \in N' : \text{order}(i) = r$; | 75 | begin |
| 30 for all $(i,j) \in A'$ do $SP_j := \min\{SP_i, SP_i + d_{ij}\}$; | 76 | let $\text{length}(k) := p_{ik} - p_{sk}$; |
| 31 end ; | 77 | let $j := u_{sk}$; |
| 32 let $D := SP_t (1 + \beta)$; | 78 | $\text{path}(k) := \{s, j\}$; |
| 33 for $r = 1$ to $ N' $ do | 79 | if $j \neq t$ then |
| 34 begin | 80 | select $k' \in K_j : p_{jk'} = p_{sk} + d_{sj}$ and $v_{jk'} = v_{sk} - 1$; |
| 35 select $i \in N' : \text{order}(i) = r$; | 81 | while $u_{jk'} \neq t$ do |
| 36 for all $(i,j) \in A'$ do $lp_j := \max\{lp_i, lp_i + d_{ij}\}$; | 82 | begin |
| 37 end ; | 83 | let $i := j$; |
| 38 let $LP_t := lp_t$; | 84 | $j := u_{ik'}$; |
| 39 let $P := \emptyset$; $T := \emptyset$; | 85 | $k'' := k'$; |
| 40 for all $i \in N'$ do $K_i := \emptyset$; | 86 | $\text{path}(k) := \text{path}(k) \cup \{j\}$; |
| 41 let $k := 1$; | 87 | clear k' ; |
| 42 let $T := T \cup \{t\}$; | 88 | select k' at $\in K_j : p_{jk'} = p_{ik''} + d_{ij}$ and $v_{jk'} = v_{ik''} - 1$; |
| 43 let $p_{ik} := \min\{D, LP_t\}$; $v_{ik} := 1$; $u_{ik} := \emptyset$; | 89 | end ; |
| 44 let $k := \{k, p_{ik}, v_{ik}, u_{ik}\}$; | 90 | end ; |
| 45 $K_i := K_i \cup \{k\}$; | 91 | end ; |
| | | end ; |

Figure 5. Pseudocode for Algorithm 3: Path Enumeration with Side Constraints.

Proposition 5: Pruning a state k at node i when $v_{ik} + \delta_i > n$ does not eliminate any paths with $\leq n$ nodes visited.

Proof: By definition, a path is infeasible if and only if $v_{ik} + \delta_i$ is greater than to the maximum number of nodes allowed to be visited in the path between nodes s and t that contains node i . This is since v_{ik} represents the number of nodes on the path from i to t for state k and δ_i represents the minimum number of nodes to reach node i from the origin. Therefore, pruning a state k at node i when $v_{ik} + \delta_i > n$ does not eliminate any paths with $\leq n$ nodes visited.

Proposition 6: Pruning a state k at node i when $p_{ik} < 0$ does not eliminate any paths of length less than or equal to D .

Proof: By definition, p_{ik} represents the remaining distance that can be traveled on a path from the origin to node i to complete the path represented by state k . Since d_{ij} is non-negative for all $(i,j) \in A'$, if $p_{ik} < 0$, a path between nodes s and i will exceed the remaining path slack. Therefore, pruning a state k at node i when $p_{ik} < 0$ does not eliminate any paths of length less than or equal to D .

Proposition 7: The path enumeration algorithm terminates in a finite number of steps for a directed acyclic graph G' .

Proof: Since there are no cycles in graph G' , each node will be added to the set of pending nodes T only once when the respective outdegree is equal to zero. Since there are a finite number of nodes in the network, the procedure will terminate in a finite number of steps.

Proposition 8: The algorithm enumerates all paths between s and t that satisfy a limit on the number of nodes visited and a maximum total length in a directed acyclic graph G' .

Proof: From *Theorem 3*, we know that the algorithm is able to enumerate all paths between s and t in a directed acyclic graph G' when we relax side constraints on the number of nodes visited and the total length of the path. Also, *Proposition 5* shows that only paths that exceed the limit on the number of nodes visited from s to t are eliminated by the algorithm and *Proposition 6* shows that only paths that exceed the maximum total length allowed on a path from s to t are eliminated by the algorithm. Therefore, when we consider these side constraints simultaneously, the algorithm leaves us with all paths between s and t that satisfy a limit on the number of nodes visited and a maximum total length in a directed acyclic graph G' .

2.4. Computational Results

2.4.1. Experimental Design

To test the performance of our procedure to solve PENVLC, we developed an experiment using the motivating context of the TL relay network design problem. Our experimental design accounts for different aspects of relay network design such as the size of the market served by TL carriers and the existing road infrastructure that is used to provide service. In addition, we establish limits on the number of nodes visited (n) and circuitry (β) permitted between O-D pairs in the network. The selected factors and their respective levels are presented in Table 1.

Table 1. Experimental Design for PENVLC.

| Factors | Levels |
|----------------------------------|--|
| Size of region | Small (600 x 600 miles ²), Large (1,000 x 1,000 miles ²) |
| Arc density | Sparse (10%), Dense (45%) |
| # of nodes | 50 (only small region), 100, 200, 400 (only large region) |
| max # of nodes visited | 3, 4, 5, 10 |
| max allowed percentage circuitry | 0.1, 0.5, 1.0 |

The purpose of this experiment is to test both the performance of our algorithm, and to explore the characteristics of the solutions we obtain as a function of varying system parameters. The performance of the algorithms is reported in terms of CPU time and characteristics of the paths that are generated such as the average number of nodes visited and the average circuitry of the generated paths.

The range of values selected for circuitry allowed is intended to test the performance of the path enumeration algorithm in a broad range of situations. However, it is important to note that in the context of relay network design, a value of $\beta=1$ is very high since this can produce feasible routes that are up to twice as long as the shortest path distance between an O-D pair. In practice, this is not desirable given the significant costs of circuitous routes. It is more likely that a TL carrier would want to have a very strict limit on circuitry which makes a value of $\beta=0.1$ a more reasonable value to restrict the number of feasible paths to be enumerated. In terms of the value of n , since every load shipped between two different nodes located reasonably far apart must visit at least one relay point, we chose to use $n=3$ as the smallest possible value for the number of nodes allowed to be visited (i.e., one intermediate node is visited in addition to the origin and destination). On the other hand, a value of $n=10$ is more likely in cases where the distance between the origin and the destination nodes is extremely large. In the case where the O-D pair is not as distant, a smaller value of n is more reasonable since $n=10$ is a very relaxed bound on the number of nodes allowed to be visited. According to the discussion on the computational complexity of our algorithm in Section 2.3.5.2, we expect a better performance in terms of CPU time for instances where the value of n is lower.

Based on preliminary testing, a time limit of 60 minutes was imposed for each instance in our experimental design. During our initial testing, most of the instances were solved well below this value while the worst case execution required a running time close to 45 minutes to enumerate all feasible paths in a small-region dense network with 200 nodes. A more detailed analysis of the conditions that would require the algorithm to run for extended periods of time in excess of our stopping criterion is presented in Section 2.4.3.4 when we analyze the results for networks with higher arc density. In those cases where the algorithm is stopped prematurely, partial information is collected to analyze the total number of states created and the number of valid states obtained up to that point as well as the location in the network where the algorithm stopped.

2.4.2. Random Network Generation

Random DAGs with the desired network topologies described in Table 1 were generated using the layer-by-layer method proposed by (Tobita and Kasahara, 2002). In our implementation, the number of layers is set equal to the number of nodes needed and every node in the network is assigned to a different layer. This ensures that if an arc exists between a node in layer a (node a) and a node in layer b (node b), then there is no path from node b to node a . The probability of an existing arc between two layers is set so the desired arc density in the network is achieved.

The location of each node is determined by generating uniform random values between 0 and 1 for x -coordinates and y -coordinates and scaling them to the length of the region. The corresponding arc distances were computed using the Euclidian norm. Ten random networks (instances) for each combination of region size and arc density were constructed for a given

network size determined by the experimental design (i.e., 50, 100 and 200 nodes for networks in the smaller region and 100, 200 and 400 nodes for networks in the larger region). These random networks were then tested for all values of n and β given in Table 1.

2.4.3. Results

2.4.3.1. Effect of the Network Reduction Algorithm on Test Problems

We first look at the effect of Algorithm 1 – our network reduction algorithm – on the test instances that were generated for computational testing. Table 2 shows the average number of nodes eliminated and the corresponding average number of arcs removed by Algorithm 1 for all test cases under study. The values presented in Table 2 are averages for the ten replications generated for each test case. It is clear that networks with both arc density and network size strongly influence the number of nodes and arcs that are eliminated by our algorithm.

Table 2. Nodes and Arcs Eliminated by Network Reduction Algorithm (Algorithm 1).

| Nodes | Small-Region Sparse | | Small-Region Dense | | Large-Region Sparse | | Large-Region Dense | |
|-------|-------------------------------|------------------------------|-------------------------------|------------------------------|-------------------------------|------------------------------|-------------------------------|------------------------------|
| | Avg. # nodes eliminated | Avg. # arcs eliminated | Avg. # nodes eliminated | Avg. # arcs eliminated | Avg. # nodes eliminated | Avg. # arcs eliminated | Avg. # nodes eliminated | Avg. # arcs eliminated |
| 50 | 7.90 | 60.50 | 0.00 | 0.00 | --- | --- | --- | --- |
| 100 | 10.20 | 181.80 | 0.20 | 18.10 | 7.30 | 134.00 | 0.10 | 9.00 |
| 200 | 6.90 | 263.20 | 0.20 | 35.30 | 9.80 | 378.60 | 0.20 | 35.80 |
| 400 | --- | --- | --- | --- | 11.40 | 891.70 | 0.00 | 0.00 |

2.4.3.2. Results for Small-Region Sparse Networks

Before looking at our computational results, it is important to mention that based on our initial testing, we implemented slight modifications to Algorithm 3 to reduce total CPU time. One of the modifications removes nodes that do not have a path to the origin from the set of pending nodes. The other modification terminates the algorithm after the origin node has been

made permanent regardless of remaining elements in the set of pending nodes T . This modified Algorithm 3 was applied to all test cases in our experimental design and the results shown correspond to its performance. We implemented our approach using Python 2.6 and ran our computational experiments on a 3.20 GHz Intel® Xeon® processor with 6 GB of RAM.

The first set of computational results is presented for transportation networks that have small region size and sparse arc density. In Table 3 we report the average values for each of the performance metrics obtained over the ten instances with the exception of the values that are italicized. In these cases, one or two instances did not have any paths that satisfy the combination of the number of nodes visited and the total path length constraints. A more detailed analysis of the instances without a solution indicates that for sparse networks, a low value of n is more likely to lead to the infeasibility of PENVLC.

Table 3. Computational Results for Small-Region Sparse Networks.

| | | Nodes | | | | | | | | | | | |
|-----|---------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|
| | | 50 | | | | 100 | | | | 200 | | | |
| n | β | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry |
| 3 | 0.1 | <i>0.042</i> | 1.25 | 2.88 | <i>0.004</i> | <i>0.098</i> | 1.63 | 2.88 | <i>0.019</i> | <i>0.649</i> | 1.88 | 3.00 | <i>0.027</i> |
| | 0.5 | <i>0.024</i> | 1.89 | 2.96 | <i>0.123</i> | <i>0.098</i> | 2.63 | 2.91 | <i>0.104</i> | 0.651 | 4.90 | 3.00 | 0.193 |
| | 1 | <i>0.029</i> | 2.22 | 2.96 | <i>0.220</i> | <i>0.098</i> | 3.25 | 2.94 | <i>0.185</i> | 0.648 | 7.40 | 3.00 | 0.393 |
| 4 | 0.1 | 0.034 | 1.30 | 3.20 | 0.005 | 0.098 | 2.30 | 3.38 | 0.018 | 0.651 | 4.50 | 3.58 | 0.029 |
| | 0.5 | 0.034 | 3.60 | 3.45 | 0.171 | 0.098 | 8.80 | 3.64 | 0.207 | 0.647 | 25.00 | 3.71 | 0.253 |
| | 1 | 0.029 | 7.20 | 3.70 | 0.459 | 0.098 | 17.90 | 3.76 | 0.455 | 0.653 | 72.20 | 3.83 | 0.602 |
| 5 | 0.1 | 0.024 | 1.30 | 3.20 | 0.005 | 0.098 | 2.60 | 3.44 | 0.022 | 0.652 | 5.90 | 3.77 | 0.032 |
| | 0.5 | 0.029 | 5.30 | 3.68 | 0.209 | 0.100 | 15.30 | 4.09 | 0.259 | 0.674 | 57.20 | 4.26 | 0.289 |
| | 1 | 0.034 | 14.30 | 4.16 | 0.564 | 0.103 | 52.10 | 4.40 | 0.568 | 0.727 | 307.00 | 4.54 | 0.691 |
| 10 | 0.1 | 0.028 | 1.30 | 3.20 | 0.005 | 0.099 | 2.60 | 3.44 | 0.022 | 0.665 | 6.00 | 3.79 | 0.033 |
| | 0.5 | 0.029 | 5.60 | 3.71 | 0.211 | 0.105 | 18.60 | 4.22 | 0.268 | 0.782 | 90.80 | 4.69 | 0.312 |
| | 1 | 0.026 | 18.10 | 4.39 | 0.598 | 0.147 | 111.40 | 4.98 | 0.626 | 4.340 | 1,323.70 | 5.72 | 0.767 |

From the results in Table 3, it is clear that the number of feasible paths increases as we relax the number of nodes visited and the total path length constraints by increasing the values of

n and β respectively. Furthermore, we can observe in Figure 6 for networks with 50 nodes that when the maximum allowed circuitry value is more restrictive ($\beta=0.1$), the increase in the number of feasible paths for higher values of n is not as significant as in the case where the value of β is not as restrictive ($\beta=1.0$).

Similarly, when the number of nodes visited constraint is more restrictive ($n=3$), the increase in the number of feasible paths for higher values of circuitry allowed is not as significant as in the case where the value of n is higher. In Figure 7 it can be observed that when the number of nodes visited constraint is relaxed to allow up to 10 nodes, the difference in average number of feasible paths for different values of β becomes significant. The same results are even more evident for networks with higher number of nodes. The effects of both side constraints on the average number of paths generated in the cases with 100 and 200 nodes can be observed in Figure 8. An analysis of variance completed in Minitab® indicates that both factors, n and β , are significant with p-values of 0.004 and 0.001 respectively for a level of significance of 0.05. As it is also observed in Figure 8, the number of nodes in the network has a significant effect on the number of nodes that are enumerated.

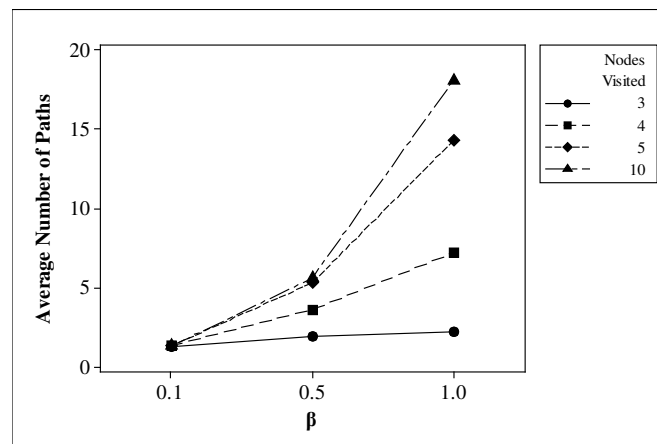


Figure 6. Effect of Circuitry on Small-Region Sparse Networks (50 Nodes).

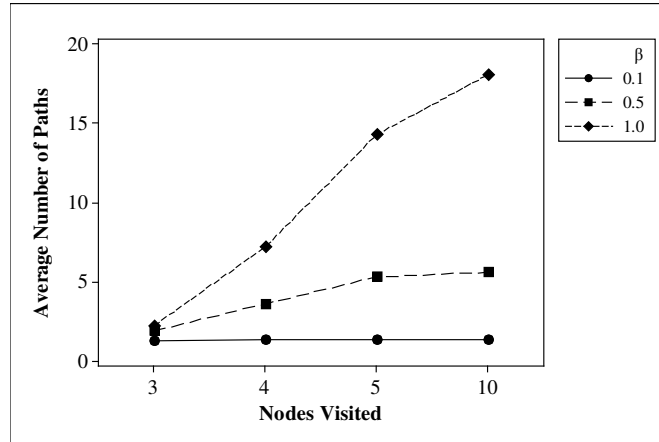


Figure 7. Effect of Nodes Allowed in a Path on Small-Region Sparse Networks (50 Nodes).

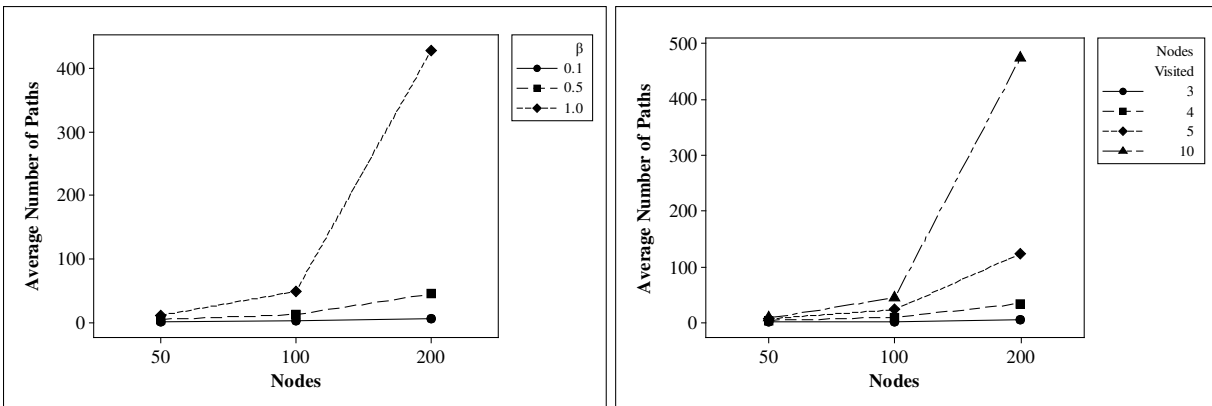


Figure 8. Effects of Circuitry and Nodes Allowed in a Path on Small-Region Sparse Networks.

We observe in Table 3 that the value of β seems to have a significant impact on the number of nodes visited in the enumerated feasible paths – that is, the paths tend to visit fewer nodes regardless of the value of n when the value for circuitry allowed is very restrictive ($\beta=0.1$). We also observe that an increase in the number of nodes allowed to be visited seems to have a significant effect on the average number of nodes visited in feasible paths only on those cases where the maximum allowed circuitry is high ($\beta=1$).

Looking at the efficiency of our proposed algorithmic approach, we observe that for larger networks with 100 and 200 nodes the total CPU time required is proportional to the

number of feasible paths that are enumerated (Table 3). For smaller networks (50 nodes), there is no significant difference between total CPU times across all values of n and β since the number of paths that are enumerated is very small. In all cases, the average total CPU time is under one second with only one exception.

In addition to total CPU time, we recorded other time metrics such as the CPU time needed to setup the problem (i.e., CPU time needed for Algorithm 1 and lines 1-45 of Algorithm 3), the CPU time needed to generate states within the enumeration algorithm (lines 46-73 of Algorithm 3) and the CPU time required to reconstruct the paths obtained (lines 74-99 of Algorithm 3). Overall, for small-region sparse networks, the most significant element of the total CPU time was the time required to setup the problem. Both the CPU time to generate states and to build paths are proportional to the number of paths that are enumerated. We observe that the setup CPU time is similar for networks of the same size, but increases with network size. We also observed that the fraction of the total CPU time that corresponds to setting up the problem becomes less significant for networks with more nodes, decreasing from more than 85 percent of the total CPU time in networks with 50 nodes to less than 45 percent in networks with 200 nodes.

2.4.3.3. Effect of Increasing the Size of the Service Region

Table 4 shows the computational results for large-region sparse networks. Overall, we observe similar results for both large and small regions in terms of major trends observed. However, it is very important to note that the variability on the number of feasible paths increases significantly as the size of the network and the values of n and β increase. A clear indication of this behavior can be observed on the metrics obtained when considering the values

$n=10$ and $\beta=1$ for networks with 200 and 400 nodes. For 200 node networks, the number of feasible paths obtained with the algorithm varies significantly across the different replications ranging from 6 to 14,100. In the case with 400 node networks, the average values presented in Table 4 (shown with an asterisk) actually correspond to nine replications completed within the stopping criterion set for our algorithm. Here, we encountered the first instance for which the algorithm was not able to obtain a solution after 60 minutes due to an extremely large number of feasible paths.

Table 4. Computational Results for Large-Region Sparse Networks.

| | | Nodes | | | | | | | | | | | |
|-----|---------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|
| | | 100 | | | | 200 | | | | 400 | | | |
| n | β | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry |
| 3 | 0.1 | 0.101 | 1.33 | 2.85 | 0.014 | 0.632 | 3.25 | 2.90 | 0.025 | 4.925 | 2.38 | 2.94 | 0.028 |
| | 0.5 | 0.101 | 2.20 | 2.88 | 0.129 | 0.631 | 7.25 | 2.94 | 0.147 | 4.944 | 6.50 | 2.98 | 0.193 |
| | 1 | 0.102 | 3.10 | 2.88 | 0.278 | 0.632 | 7.56 | 2.94 | 0.267 | 4.980 | 10.25 | 2.99 | 0.396 |
| 4 | 0.1 | 0.102 | 2.20 | 3.31 | 0.023 | 0.638 | 7.89 | 3.49 | 0.033 | 5.023 | 4.40 | 3.52 | 0.026 |
| | 0.5 | 0.102 | 7.40 | 3.46 | 0.179 | 0.640 | 44.30 | 3.75 | 0.238 | 5.053 | 26.20 | 3.79 | 0.274 |
| | 1 | 0.102 | 15.00 | 3.55 | 0.420 | 0.642 | 85.70 | 3.86 | 0.505 | 5.035 | 79.70 | 3.89 | 0.613 |
| 5 | 0.1 | 0.103 | 2.50 | 3.43 | 0.023 | 0.654 | 13.20 | 3.87 | 0.034 | 5.045 | 5.90 | 3.74 | 0.033 |
| | 0.5 | 0.104 | 17.20 | 3.84 | 0.201 | 0.687 | 168.80 | 4.33 | 0.271 | 5.200 | 67.80 | 4.18 | 0.296 |
| | 1 | 0.106 | 47.90 | 4.05 | 0.502 | 0.758 | 556.90 | 4.60 | 0.630 | 5.474 | 365.70 | 4.58 | 0.715 |
| 10 | 0.1 | 0.103 | 2.60 | 3.47 | 0.024 | 0.714 | 16.20 | 3.94 | 0.035 | 5.134 | 6.40 | 3.76 | 0.033 |
| | 0.5 | 0.114 | 37.00 | 4.16 | 0.217 | 2.259 | 534.00 | 4.93 | 0.298 | 6.730 | 212.80 | 4.50 | 0.307 |
| | 1 | 0.327 | 226.40 | 4.67 | 0.561 | 42.477 | 3,187.00 | 5.78 | 0.729 | 11.171* | 531.33* | 5.28* | 0.767* |

Looking at the actual CPU times presented in Table 4, it is interesting to note that the majority of average times are still below 1 second except in cases where the variability increases significantly for networks with 100 and 200 nodes. However, increasing the size of the network to 400 nodes causes a considerable increase in total CPU time to values close to 5 seconds and above. In most cases, setup CPU time still represents a significant portion of the total CPU time, but as the number of feasible paths increases there is a significant increase in the fraction of total

CPU time that corresponds to the time required to generate states and build paths. In fact, CPU time for building the paths becomes the more important portion of the total CPU time for those cases with a very high number of paths enumerated which demonstrates that the complexity of our approach is driven by this portion of the algorithm as discussed in Section 2.3.5.2.

2.4.3.4. Effect of Increasing Arc Density

In a different set of computational tests, we substantially increased the arc density of the networks. We found that high values of n and β result in a very significant increase in the number of feasible paths that need to be enumerated relative to the lower density instances we considered. We also observe longer run times for our algorithm due to the proliferation of valid states. For example, assume that node i in a network with 200 nodes has outgoing arcs to nodes j and k . If the number of valid states at nodes j and k are 1 million and 800 thousand respectively, the algorithm has to analyze a total of 1.8 million possible states at node i . Assuming that only 60 percent of those states are valid, the algorithm will have to analyze at least 1.08 million states at any of the predecessor nodes to node i in addition to all other corresponding valid states. In general, the time required to generate states will continue to increase as the algorithm moves closer to the origin node (i.e., for those nodes with lower topological order). In our experimentation, we were able to solve all instances of small-region dense networks with 50 nodes, but some replications exceeded the time limit of 60 minutes for networks with 100, 200 and 400 nodes for small-region and large-region dense networks. In addition to this, we also observed high variability in the number of paths generated for instances solved when the value of $n \geq 5$. To illustrate the variability observed, Table 5 presents all instances for large-region dense networks with 200 nodes and values of $n=10$ and $\beta=0.1$, noting which of those we were unable to

solve within our 60 minute time limit. We observe a significant variation among the instances that were completely solved with valid states that range from 4 to 304,828 and feasible paths ranging from 1 to 319. It is also clear that the instances that were solved in a short period of time have a much lower number of total and valid states as opposed to those instances that were not solved within 60 minutes. Furthermore note that the instances that were not solved actually were not even close to termination since the final node visited was in all cases several nodes from the origin.

Table 5. States in Solved and Unsolved Instances for Large-Region Dense Networks.

| | Replication | CPU time (sec) | # Paths | # Total States | # Valid States | Final Node |
|---------------|-------------|----------------|----------|----------------|----------------|------------|
| $N = 200$ | | | | | | |
| $n = 10$ | 1 | 4.808 | 99 | 374,632 | 10,428 | 1 |
| $\beta = 0.1$ | 2 | 3.108 | 12 | 20,820 | 337 | 1 |
| | 3 | Stop | Unsolved | 14,635,144 | 957,280 | 55 |
| | 4 | 3.034 | 2 | 1,376 | 19 | 1 |
| | 5 | Stop | Unsolved | 14,728,754 | 574,461 | 39 |
| | 6 | 3.018 | 1 | 341 | 4 | 1 |
| | 7 | 47.512 | 197 | 3,205,634 | 107,991 | 1 |
| | 8 | Stop | Unsolved | 14,682,095 | 663,748 | 29 |
| | 9 | 217.415 | 319 | 7,520,374 | 304,828 | 1 |
| | 10 | 23.790 | 276 | 2,006,095 | 48,045 | 1 |

Table 6 and Table 7 present the computational results found for small-region and large-region dense networks respectively. Due to the reduced number of instances completed and the high variability observed on the values obtained, we present only solutions for $n=3$ and $n=4$ for both small-region and large-region dense networks. Other results are not included since the algorithm solved on average only five out of ten replications within our 60 minute time limit. These results suggest that the proliferation of states discussed at the beginning of this section is a

significant issue for the unsolved instances. For this reason, increasing the time limit would not really help us to obtain results that can be analyzed with a central tendency measure such as the average due to the significant variability observed between the different replications for these cases when $n \geq 5$ and the number of nodes in the network is large.

Table 6. Computational Results for Small-Region Dense Networks.

| | | Nodes | | | | | | | | | | | |
|-----|---------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|
| | | 50 | | | | 100 | | | | 200 | | | |
| n | β | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry |
| 3 | 0.1 | 0.070 | 3.90 | 2.68 | 0.026 | 0.416 | 15.70 | 2.77 | 0.026 | 3.057 | 29.40 | 2.96 | 0.035 |
| | 0.5 | 0.072 | 11.10 | 2.88 | 0.195 | 0.419 | 38.20 | 2.96 | 0.197 | 3.067 | 83.70 | 2.99 | 0.200 |
| | 1 | 0.072 | 19.10 | 2.93 | 0.437 | 0.418 | 55.80 | 2.98 | 0.382 | 3.065 | 123.80 | 2.99 | 0.381 |
| 4 | 0.1 | 0.074 | 6.70 | 3.01 | 0.032 | 0.494 | 62.80 | 3.38 | 0.038 | 3.908 | 192.50 | 3.77 | 0.051 |
| | 0.5 | 0.080 | 36.60 | 3.48 | 0.254 | 0.596 | 447.60 | 3.72 | 0.280 | 5.306 | 1,812.20 | 3.94 | 0.302 |
| | 1 | 0.091 | 120.30 | 3.72 | 0.610 | 0.730 | 1,173.90 | 3.85 | 0.599 | 7.418 | 5,113.30 | 3.97 | 0.611 |

Table 7. Computational Results for Large-Region Dense Networks.

| | | Nodes | | | | | | | | | | | |
|-----|---------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|-----------------|--------------|--------------|----------------|
| | | 100 | | | | 200 | | | | 400 | | | |
| n | β | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry | CPU total (sec) | Avg. # paths | Avg. # nodes | Avg. circuitry |
| 3 | 0.1 | 0.415 | 9.00 | 2.68 | 0.029 | 3.037 | 23.90 | 2.82 | 0.030 | 23.300 | 48.50 | 2.92 | 0.037 |
| | 0.5 | 0.418 | 25.30 | 2.90 | 0.203 | 3.050 | 64.30 | 2.86 | 0.162 | 23.467 | 132.10 | 2.98 | 0.202 |
| | 1 | 0.417 | 37.30 | 2.94 | 0.400 | 3.067 | 96.30 | 2.88 | 0.345 | 23.493 | 190.00 | 2.99 | 0.398 |
| 4 | 0.1 | 0.464 | 30.90 | 3.08 | 0.036 | 3.729 | 171.90 | 3.45 | 0.042 | 39.568 | 737.20 | 3.67 | 0.053 |
| | 0.5 | 0.539 | 259.30 | 3.53 | 0.265 | 4.799 | 1,413.40 | 3.64 | 0.249 | 74.105 | 6,252.60 | 3.90 | 0.299 |
| | 1 | 0.625 | 708.40 | 3.75 | 0.601 | 6.443 | 3,869.00 | 3.70 | 0.534 | 117.350 | 15,579.10 | 3.96 | 0.619 |

Comparing the results for dense networks with those for sparse networks of the corresponding size, we observe that the average number of feasible paths increases significantly due to the increase in arc density. However, it is interesting to note that the average total CPU times are still below 1 second for networks with 50 and 100 nodes. A more detailed analysis of CPU times reveals that the setup CPU time for dense networks is not as significant as in the previous cases with sparse networks. This is because networks with higher arc density do not

rely on the network reduction algorithm (Algorithm 1) as previously discussed in Section 2.4.3.1; therefore the time required to set up and reduce the network is a small fraction of the total CPU time for the algorithm. In terms of the quality of the generated paths, the same general results previously observed for sparse networks are obtained when looking at the values for average number of nodes visited and average circuitry of the feasible paths. We again observe that the value of β seems to have an impact on the number of nodes visited with paths that have fewer nodes regardless of the value of n , especially when $\beta=0.1$. In addition, increasing the value of n increases the average number of nodes visited as the value of circuitry allowed gets higher.

2.4.3.5. Relevance of Computational Results

The computational results obtained indicate that the proposed approach is a significant contribution in terms of solving PENVLC. The large size of the networks as well as the high values set for the side constraints imposed do not allow us to solve the problem within a 60 minute time limit. Higher values for n and β would make this even more apparent. However, in the absence of our algorithm, we would need to enumerate all paths in the network, and then sort through them to perform a feasibility check; this would only cause the time and computational effort required to solve this problem to increase.

Our proposed algorithm allows us to generate only the paths which are feasible in a matter of seconds for reasonable values of n and β . Even in those instances in which we have very generous values of n and β , the proposed method would prove to be more efficient to enumerate the complete set of paths satisfying the side constraint imposed since only the feasible ones are obtained and no post-processing is necessary.

2.5. Conclusions and Future Research

Our research contributes to the path enumeration literature by developing an exact approach to generate feasible paths for large instances of PENVLC. The proposed algorithmic approach avoids enumerating all paths and sorting through them to obtain the set of paths that satisfy the side constraints imposed. This represents a considerable advantage in terms of efficiency, particularly when a small proportion of paths satisfy the side constraints (e.g., small values of n and β). In addition, our approach avoids setting values for k , a significant drawback of using existing k -shortest path algorithms for PENVLC. Furthermore, by simultaneously considering both side constraints, it also overcomes one of the disadvantages of several constrained shortest path and constrained k -shortest path algorithms that require checking paths for every side constraint in a sequential manner.

The computational results obtained in our research indicate that as the number of nodes allowed to be visited increases, the number of feasible paths that need to be enumerated increases as well. A similar observation is made in terms of allowing more circuitry for the paths (i.e., increased circuitry leads to a larger number of feasible paths). In both cases, larger CPU times are required as the number of feasible paths grows. Our experiments also indicate that for small sparse networks, highly restricting the number of nodes visited (i.e. a low value of n) may actually result in infeasible instances of PENVLC. Our proposed algorithm is more efficient than regular methods in determining the infeasibility of the problem since it explicitly considers the side constraints as opposed to generating a large number of paths and post-processing them trying to find a valid path that does not exist. We also found that arc density has significant

effect on the number of feasible paths that exist in a network, and consequently on the CPU time required to obtain these paths. This is more evident for higher values of n and β .

In our computational experiments, we observed that network topology also plays a significant role in the variability found across replications, especially for networks with a large number of nodes (e.g. more than 100 nodes). We observed for large dense networks cases with only one feasible path and others with thousands of them. This results in some instances that can be solved in a few seconds while others of similar type require an extended period of time to solve. Our algorithm is particularly challenged by those instances with a large number of feasible paths. However, we note that the number of feasible paths in a network is mostly driven by the bounds on nodes visited (n) and path length (β). Appropriate values for these parameters depend upon the specific application for which this algorithm is used. In our motivating problem, low values of β and n are likely to exist in practice. From our experiments, most instances of these characteristics are solved very quickly (i.e., on average less than 12 seconds). As a result, our proposed method is adequate for PENVLC problems generated in the context of the TL relay network design problem.

There are a number of promising extensions to the PENVLC algorithm we propose. While our algorithm is designed for DAGs, it would be interesting to extend the approach to general networks with cycles since many transportation problems may be best modeled by a general network. Another promising area of future research relates to extending the proposed algorithm to include additional side constraints other than the ones considered in PENVLC via the labels used to represent valid states. In the context of the TL relay network design problem, some periodic service requirements may be enforced such as the need to stop periodically for

fuel or maintenance in route between origin and destination. These requirements would impose constraints to require that a node be visited within a specific time or distance threshold (e.g., cannot travel more than 500 miles without stopping for fuel). Another potential side constraint corresponds to the need to satisfy some sort of service guarantee (e.g., deliver a load in a given time window). The addition of a time dimension in this problem results in an interesting challenge that is worth considering as future research.

Other transportation problems such as the emergency vehicle routing problem focus on generating paths that are sufficiently different so that a real alternative route can be used between a particular O-D pair. This type of requirement is also observed when risk is incorporated in the definition of alternative routes against disruptions in a transportation network. Incorporating a limitation on path similarity is not straightforward, but it would make it relevant for a wider range of practical problems that currently rely on heuristic methods. This would require defining the appropriate modifications to the labels used and to the strategy for generating valid states so that the set of feasible dissimilar paths is efficiently obtained. This is another area that we would certainly like to study in the future. The work done in the current research is a solid step towards the development of exact algorithms that can be applied efficiently in real world transportation problems that require enumerating paths that satisfy several additional constraints.

References

- Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice Hall.
- Aneja, Y.P., Aggarwal, V., and Nair, K.P. (1983). Shortest Chain Subject to Side Constraints. *Networks*, 13, 295-302.

- Carlyle, W.M., and Wood, R.K. (2005). Near-Shortest and K-Shortest Simple Paths. *Networks*, 46(2), 98-109.
- Carlyle, W.M., Royset, J.O., and Wood, R.K. (2008). Lagrangian Relaxation and Enumeration for Solving Constrained Shortest Path Problems. *Networks*, 52(4), 256-270.
- Dreyfus, S.E. (1969). An Appraisal of Some Shortest-Path Algorithms. *Operations Research*, 17(3), 395-412.
- Eppstein, D. (1998). Finding the k Shortest Paths. *SIAM Journal of Computing*, 28(2), 652-673.
- Filippovich, V. (1973). Transforming a Cyclic Directed Graph into an Acyclic Graph. *Cybernetics and Systems Analysis*, 9(2), 348-351.
- Hadjiconstantinou, E., and Christofides, N. (1999). An Efficient Implementation of an Algorithm for Finding K Shortest Simple Paths. *Networks*, 34, 88-101.
- Handler, G. Y., and Zang, I. (1980). A Dual Algorithm for the Constrained Shortest Path Problem. *Networks*, 10, 293-310.
- Hershberger, J., Maxel, M., and Suri, S. (2007, November). Finding the k Shortest Simple Paths: A New Algorithm and Its Implementation. *ACM Transactions on Algorithms*, 3(4), 45:1-45:19.
- Hoffman, W., and Pavley, R. (1959). A Method for the Solution of the Nth Best Path Problem. *Journal of the Association for Computing Machinery (ACM)*, 6, 506-514.
- Katoh, N., Ibaraki, T., and Mine, H. (1982). An Efficient Algorithm for K Shortest Simple Paths. *Networks*, 12, 411-427.
- Lawler, E. L. (1972). A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem. *Management Science*, 18(7), 401-405.
- Lawler, E. L. (1976). *Combinatorial Optimization: Networks and Matroids*. New York, NY: Holt, Rinehart & Winston.
- Liu, G., and Ramakrishnan, K.G. (2001). A*Prune: An Algorithm for Finding K Shortest Paths Subject to Multiple Constraints. *Proceedings IEEE INFOCOM*, 743-749.
- Martins, E.Q., and Pascoal, M.M. (2003). A New Implementation of Yen's Ranking Loopless Paths Algorithm. *4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1, 121-133.

- Park, D., and Rilett, L.R. (1997). Identifying Multiple and Reasonable Paths in Transportation Networks: A Heuristic Approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1607, 31-37.
- Park, D., Sharma, S.L., Rilett, L.R., and Chang, M. (2002). Identifying Multiple Reasonable Alternative Routes: Efficient Vector Labeling Approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1783, 111-118.
- Shi, N. (2010). K Constrained Shortest Path Problem. *IEEE Transaction on Automation Science and Engineering*, 7(1), 15-23.
- Taylor, G.D., Whicker, G.L., and Usher, J.S. (2001). Multi-Zone Dispatching in Truckload Trucking. *Transportation Research: Part E*, 37, 375-390.
- Tobita, T., and Kasahara, H. (2002). A Standard Task Graph Set for Fair Evaluation of Multiprocessor Scheduling. *Journal of Scheduling*, 5(5), 379-394.
- Üster, H., and Kewcharoenwong, P. (2011). Strategic Design and Analysis of a Relay Network in Truckload Transportation. *Transportation Science*, 45(4), 505-523.
- Üster, H., and Maheshwari, N. (2007). Strategic Network Design for Multi-Zone Truckload Shipments. *IIE Transactions*, 39, 177-189.
- van der Zijpp, N., and Fiorenzo Catalano, S. (2005). Path Enumeration by Finding the Constrained K-shortest Paths. *Transportation Research: Part B*, 39, 545-563.
- Yen, J. Y. (1971). Finding the K Shortest Loopless Paths in a Network. *Management Science*, 17(11), 712-716.

3. Composite Variable Model for Truckload Relay Network Design

Driver retention is a significant problem for full truckload carriers that operate using a point-to-point dispatching method. As an alternative, dispatching under a network configuration of relay points can be used to reduce driver tour lengths and consequently improve driver retention. We present a new mathematical formulation for strategic relay network design that minimizes total costs while considering important operational constraints such as load circuitry and driver tour length within the variable definition. This formulation allows us to find very high quality solutions in reasonable times for largely-sized problem instances. Computational results for generated instances and a test case problem provided by a major carrier are presented along with a discussion of areas for future research.

3.1. Introduction

Trucking is the predominant mode of land transportation of freight in the United States, accounting for 71.3% of total freight expenditures (U.S. Bureau of Transportation Statistics, 2010). One of the major challenges faced by truckload (TL) carriers is that of driver retention; typical truckload carriers experience an annual turnover of more than 100% (Campbell, 2005). This is very costly since carriers must invest significant resources in recruiting, training, and retaining drivers. Industry estimates that this costs approximately \$8,200 per driver. This adds up to a significant expense for TL carriers, on the order of \$2.8 billion annually (Rodriguez et al., 2000). Also, the American Transportation Research Institute (2011) reports that driver turnover has not been placed below the sixth spot on the list of top concerns for the trucking industry since 2005, and has been in the top three spots in five of the last seven years. In fact, the

importance of retaining drivers is even more evident now that there is an estimated shortage of 125,000 drivers (Morris, 2011).

TL carriers typically move loads point-to-point (PtP); that is, loads are picked up at their origin and moved via a single driver to their destination. For example in a PtP network, if a driver delivers a load to Fayetteville, AR, the carrier must either find a load leaving Fayetteville or have the driver travel to a nearby city to pick up another load. These loads can be difficult to identify due to imbalanced freight networks. Carriers must construct *tours* – a series of truckload movements that begin and end at a driver's base terminal – that allow drivers to return home. These tours are difficult to construct, and often leave drivers on the road for two or three weeks at a time. The time that drivers spend away from their homes is one of the main causes of driver turnover for TL carriers (Campbell, 2005).

Contrast this with less-than-truckload (LTL) carriers' operations. LTL carriers move smaller shipments than TL carriers. As a result, these shipments must travel via a series of hubs where freight is sorted and consolidated. This means the drivers' trips are shorter since they go from one hub to another. Since freight is moved primarily between hubs, drivers return home much more frequently; in some cases, nightly. Turnover for LTL carriers is only around 10% (Campbell, 2005).

Motivated by the differences in driver retention between LTL carriers and TL carriers, TL carriers are considering constructing a network of *relay points* (RPs). This would be referred to as a *relay network*, and would serve as an alternative to traditional PtP dispatching. Using a relay network, freight would be transported from its origin through a series of relay points to its destination. This would essentially mimic LTL operations, except that no freight would be

handled at the relay points; these points would simply serve as points in the network for drivers and loads to be exchanged (Campbell, 2005).

Relay points introduce a fundamental tradeoff that must be considered – the cost of the relay network must be balanced against improved service and a reduction in driver turnover. Relay networks increase cost for two primary reasons. First, carriers must pay to open and operate relay points. Second, circuitry is introduced since traveling through a network of RPs is not as efficient as direct PtP transportation. However, these costs must be balanced against the increase in driver retention since a relay network configuration allows drivers to return to their home domiciles more frequently. In addition, there are service implications for RPs. Counterintuitively, a load's transit time can be reduced under a relay configuration because loads can travel continuously; since loads can be exchanged at RPs, they need not be inhibited by regulations that limit hours of service that would be in place if a single driver moved a load PtP (Taha and Taylor, 1994).

This chapter introduces a new mathematical model for the truckload relay network design (TLRND) problem. There are a number of contributions to the literature associated with this work. First, we develop a prescriptive composite variable model (CVM) to solve TLRND. We describe how this model can be used to obtain both exact and heuristic solutions for TLRND, depending upon the way in which the composite variables are generated. Second, our CVM is able to explicitly capture operational constraints such as the number of RPs visited and limits on route circuitry for the truckloads that to this point have been acknowledged as important by researchers in this area, but relaxed or replaced by surrogates in the previously-published studies on prescriptive models for TLRND. We also explain how our CVM can be easily extended to

incorporate additional operational constraints. Third, we exercise our model to obtain high-quality solutions to several instances of TLRND. We explore the tradeoff between the solution quality and the time required to obtain these solutions to offer insights into the performance of our model. Finally, our model adds to the growing literature on CVMs. Although a number of CVMs have been used to model and solve a number of problems, particularly in the areas of transportation and logistics, there is not yet a unified body of literature in this area or a predominant set of guiding principles to develop CVMs. As a result, each new CVM is a contribution to the literature and helpful in developing a library of models that can be used by future researchers.

The remainder of this chapter is organized as follows. In Section 3.2, we formally introduce TLRND. In Section 3.3, we present a new mathematical formulation for TLRND that uses composite variable modeling. Section 3.4 describes the computational experiments that we completed to evaluate the performance of our proposed model and the quality of the solutions obtained. Finally, we conclude by highlighting the major findings of our research and discussing areas for future work in Section 3.5.

3.2. Truckload Relay Network Design

As opposed to the significant amount of work completed on freight transportation with consolidation for LTL shipping and express package delivery, relatively few authors have considered network structures for TL planning problems. This type of configuration is not pervasive in TL transportation since truckloads are commonly dispatched PtP. However, the benefits of operating a relay network for TL transportation make this an interesting area for research. Most of the original work in TLRND corresponds to descriptive simulation analyses of

hub-and-spoke networks, multi-zone dispatching systems and other alternative dispatching methods for TL transportation. These studies demonstrated the feasibility of such systems and explored different strategic and operational aspects of TLRND (Taha and Taylor, 1994; Taylor et al., 1995; Taylor et al., 1999; Taylor and Meinert, 2000; Taylor et al., 2001). Only a few studies have considered algorithmic solution approaches to problems on TLRND such as Hunt (1998) and Ali et al. (2002). To the best of our knowledge, the only mathematical modeling formulation for TLRND has been presented in Üster and Maheshwari (2007) and Üster and Kewcharoenwong (2011).

A general conclusion drawn by the previous descriptive studies in this area is that tradeoffs among different performance metrics must be weighed when comparing traditional dispatching methods to alternative network-based configurations for TL transportation. A description of the most important performance metrics for each of three major constituencies – drivers, carrier and customers – and how they are affected by a particular dispatching method is presented in Taylor and Meinert (2000). A general recommendation derived from these previous studies is that TL carriers are best served by operating a partial relay network with some loads being shipped using PtP dispatching to prevent excessive circuitry for some shipments.

A more recent contribution in the TLRND literature is the work of Üster and Maheshwari (2007) and Üster and Kewcharoenwong (2011). Üster and Maheshwari (2007) propose a mathematical formulation for the strategic design of a TL relay network. Their model combines elements from multi-commodity flow network formulations and the hub location problem. The basic model considers restrictions on the length of local and lane driver tours as well as maximum permissible levels of percentage circuitry and load imbalance. They were able to solve

smaller instances of this problem using this mixed-integer programming model, but their formulation is intractable for largely-sized problem instances. For this reason, the authors developed a heuristic procedure to solve a relaxed formulation of the problem without constraints to enforce circuitry limitations and load balance. A construction heuristic was implemented to generate a feasible set of RP nodes that satisfy the local and lane driver tour length constraints. Then, a tabu search procedure was used to improve the initial solution. The heuristic approach was tested with randomly generated problem instances of up to 20 nodes in the network. They were able to obtain optimality gaps of less than 1% for the tabu search in most cases.

Üster and Kewcharoenwong (2011) use a slightly modified mathematical formulation as compared to that presented in the previous paper and develop a Benders' decomposition-based algorithm for solving the problem originally without the circuitry constraints. To overcome inefficiencies in solving the problem with a typical implementation of Benders' decomposition, the authors systematically explored alternatives within an ε -optimal framework such as deriving strengthened Benders' cuts, cut disaggregation techniques and heuristics for improved upper bounds. They also used surrogate constraints in their model to enforce a limitation on circuitry. With this approach, they were able to solve larger problem instances with up to 80 nodes within reasonable solution times with $\varepsilon = 2.0\%$ for the optimality gap. These two last papers provide a significant contribution in terms of defining TLRND, formulating a mathematical model for its solution and analyzing several design aspects for the construction of TL relay networks. However, major TL carriers usually deal with practical problems with hundreds of nodes and thousands of truckloads.

Variations of TLRND also appear in other domains. For example, relay network design problems can be found in telecommunications when strategically designing a fiber-optic network with relays to overcome signal degradation due to attenuation and other factors. The most significant difference between this problem and TLRND is that balance as well as a limitation on the distances between two consecutive nodes in a path are not considered. Cabral et al. (2007) introduce this problem in the context of telecommunication network design and propose a path-based integer programming model with a column generation approach for solving small problem instances. They recommend using a subset of the feasible paths and relay combinations to solve bigger problem instances with very high quality solutions in reasonable CPU times. The use of construction heuristics is also explored in their work. More recently, Konak (2011) presents an alternative formulation for this problem based on set covering constraints to locate relays on the network and designs a genetic algorithm with a specialized crossover operator to generate feasible paths for the commodities. With this approach, problems with up to 160 nodes and 10 commodities are solved within reasonable CPU times.

3.2.1. Problem Definition

Given a set of truckloads to be transported through the network, the TLRND problem minimizes the total cost of opening RPs and routing the truckloads through the network of relay points. In a relay network (RN), each truckload must stop at one or more RPs en route from origin to destination. This configuration would be similar to a hub-and-spoke network, except that no sorting or consolidation of the loads occurs at the RPs. Consider the example shown in Figure 9. In this example, truckload ij is transported from its origin node i through a series of three relay points to its destination j . When all routings are established, all nodes in the network

can be partitioned into *zones* that surround each RP. These represent all origin and destination nodes served by a RP.

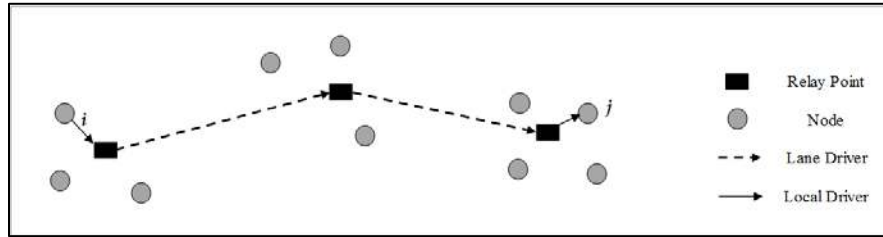


Figure 9. Relay Network for Truckload Transportation.

Under this RN configuration, two types of drivers handle the loads. *Local drivers* move loads within a zone (i.e., from an origin node to a RP or from a RP to a destination node). These local drivers are domiciled at a RP, and return home each night. The distances driven by local drivers cannot exceed a pre-specified limit, γ_1 . *Lane drivers* are responsible for inter-zone movements. These drivers travel longer distances than local drivers, transporting loads from one RP to another. Distance limitations for lane drivers are imposed to ensure federal hours-of-service regulations are satisfied. We refer to the limitation on lane distances as γ_2 .

Based on TL carrier requirements, we impose a limitation on the number of RPs that can be visited on a truckload route, λ . This is to prevent excessive stopping or circuitry, and to ensure that the delivery of loads is timely. To the best of our knowledge, this constraint has not been explicitly considered in previous research though as previously mentioned its importance is validated in discussions with carriers. Moreover, we impose a limitation on the circuitry of loads, β . Assuming that the shortest path that can be taken by a truckload is d , we allow only routes of length $(1 + \beta)d$ or shorter to move that truckload. All of these constraints must be imposed to ensure that the routes are feasible.

In addition, truckload routes and repositioning moves for empty trailers must be selected to ensure that each of the relay points is balanced. This ensures that no node suffers an excessive surplus or deficit of empty trailers and allows the carrier to provide better quality of service to shippers while reducing the number of empty miles driven.

3.3. Mathematical Model for TLRND

Previous models for TLRND are variants of arc-based multicommodity flow network and hub location formulations (Üster and Maheshwari, 2007; Üster and Kewcharoenwong, 2011). These models have difficulty incorporating some of the operational constraints encountered in practice. As a result, traditional formulations are either solved by heuristics, or relax important constraints like circuitry altogether. To overcome these challenges, we propose a composite variable formulation for TLRND. As opposed to variables which represent small, localized decisions, composite variable formulations define the variables in such a way that many of the complicating constraints are captured implicitly within the variable definition. This type of formulation has been used in the past in other integrated planning problems in transportation and logistics such as Armacost et al. (2002) in service network design for express shipping, Cohn and Barnhart (2003) and Lohatepanont and Barnhart (2004) in integrated airline planning, Cohn and Barnhart (2006) in service parts logistics, and Cohn et al. (2007) and Root and Cohn (2008) in integrated planning for small package carriers to name a few.

In this formulation, we define a variable as a feasible route for a truckload; these variables are also referred to as *composites*. Defining our variables in this way allows us to decompose the problem into smaller routing problems. For example, the route shown in Figure 1 would be one example composite. Routes that do not satisfy circuitry limitations, local and lane

distance requirements and limitations on the number of nodes visited are simply not created and therefore cannot be selected by our model. Note that in this way, the challenging operational constraints are embedded implicitly within the definition of our variables. It is important to mention that we also have composites which represent the repositioning of empty trailers; such composites may be required to achieve balanced networks. These routing decisions are then coordinated via a master problem. Given this variable definition, our model is given in the following section.

3.3.1. Composite Variable Model

To model the TLRND problem, we introduce the following notation.

Sets

R = set of composites r ,

T = set of truckloads t ,

N = set of nodes k ,

R_t = set of composites r for truckload t , $R_t \subset R$,

N_r = set of nodes visited by composite r , $N_r \subset N$.

Parameters

c_r = cost of composite r , $\forall r \in R$,

F_k = fixed cost of relay point k , $\forall k \in N$,

b_t = demand for truckload t (in number of loads), $\forall t \in T$,

δ = maximum acceptable percentage load imbalance,

$\theta_{kr} = 1$ if composite r visits relay point k , $\forall k \in N, r \in R$,

0 otherwise,

$\eta_{kr} = -1$ if node k is the origin relay point of composite r , $\forall k \in N, r \in R$,

1 if node k is the destination relay point of composite r ,

0 otherwise.

Variables

$x_r =$ number of composites r used, $\forall r \in R$,

$y_k =$ 1 if a relay point is opened at node k , $\forall k \in N$,

0 otherwise.

Given this notation, our mathematical formulation for TLRND is as follows.

$$\min \sum_{r \in R} c_r x_r + \sum_{k \in N} F_k y_k \quad (1)$$

subject to

$$\sum_{r \in R_t} x_r = b_t \quad \forall t \in T \quad (2)$$

$$\sum_{r \in R_t} \theta_{kr} x_r \leq b_t y_k \quad \forall t \in T, k \in N_r : r \in R_t \quad (3)$$

$$\sum_{r: \eta_{kr} = -1} x_r - \sum_{r: \eta_{kr} = 1} x_r \leq \delta \sum_{r: \eta_{kr} = -1} x_r \quad \forall k \in N \quad (4)$$

$$\sum_{r: \eta_{kr} = 1} x_r - \sum_{r: \eta_{kr} = -1} x_r \leq \delta \sum_{r: \eta_{kr} = 1} x_r \quad \forall k \in N \quad (5)$$

$$x_r \text{ integer} \quad \forall r \in R \quad (6)$$

$$y_k \in \{0,1\} \quad \forall k \in N \quad (7)$$

The objective function (1) minimizes the total cost of routing (i.e., the composites selected) and the fixed cost of opening RPs. Constraint (2) requires routes to be selected for each of the truckloads to be moved through the network. Constraint (3) requires that all RPs

associated with a composite be opened if that composite is used in the solution. Network balance is enforced using constraints (4) and (5) by requiring the difference between the outgoing and incoming flows at each node be less than a limitation on the node imbalance. Note that only one of these constraints will be binding at each node for values of $\delta > 0$. Constraints (6) and (7) require the integrality of all variables. Note that x_r may assume values greater than one to satisfy the demand (i.e., when $b_r > 1$ in constraint (2)), or when multiple empty trailers must be repositioned to achieve balance (constraints (4) and (5)).

3.3.2. Generating Composite Variables

A challenge of composite variable formulations such as the one proposed in Section 3.3.1 is in the large number of variables that must be generated. To generate our variables, we use an enumeration-based procedure where we define *templates*: predefined routing patterns that truckloads can take. Figure 10 shows a template with 4 nodes and 2 RPs (k and l) that is used to determine feasible routes for truckloads with origin in node i and destination in node j where the shortest path distance between i and j is given by SP_{ij} . Since TL carriers do not typically permit movements that visit more than three relay points, it is possible to construct all templates that could be used (Figure 11) and completely enumerate the variables. Empirical testing with our formulation reveals, however, that several of these templates are typically not used in optimal solutions as will be described later in Section 3.4.

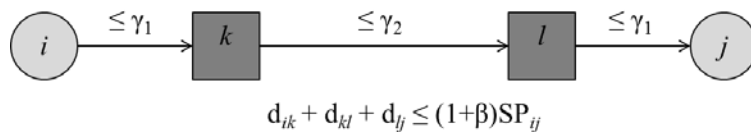


Figure 10. Example Template for Composite Generation.

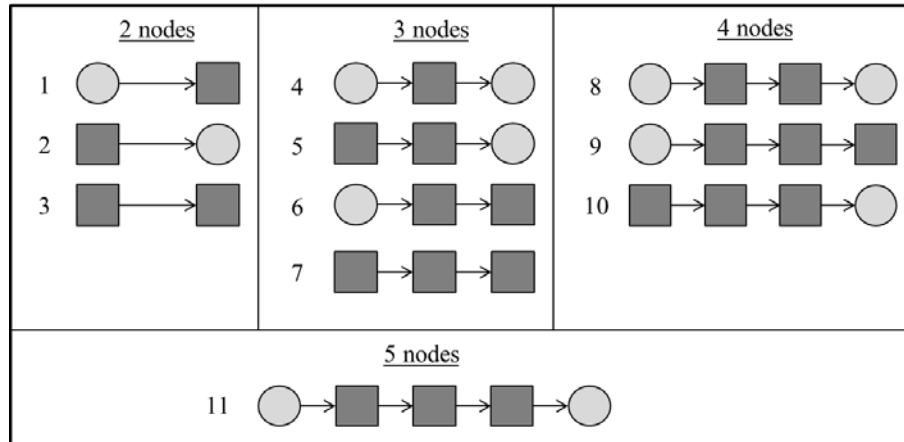


Figure 11. Set of Templates Used for Composite Generation.

To generate composites, we implemented an algorithm to check the feasibility of each of these templates for each truckload/node combination. For all combinations which satisfy operational constraints such as circuitry and local and lane distance limitations, the variables are generated and added to the model. If the feasibility requirements are not satisfied for a given template, then those variables are simply not included. By enforcing these important operational constraints implicitly within the definition of the variables, we do not need to include them as constraints in our mathematical formulation. Additionally, the number of variables that are generated is reduced by considering the dominance of certain templates with respect to others in terms of composite cost. For example, assuming that local movement rates are always less than lane movement rates, templates 1 and 2 presented in Figure 11 dominate template 3 given that each of them creates a composite with a lower transportation cost.

3.3.3. *Omitting Redundant Facility Location Constraints*

Another challenge of this formulation is the large number of constraints, particularly the facility location constraints (3). This led us to consider ways in which we could reduce this number of constraints without sacrificing solution quality. In our initial testing, we observed that

for a non-trivial proportion of truckloads t , only a single composite was generated; let T' represent the set of all such truckloads. For all truckloads $t \in T'$, the routing decisions – including the sequence of relay points k which must be visited – have already been determined; therefore they need not be considered in the model and constraints (2) can be omitted for all $t \in T'$. More importantly when a routing for $t \in T'$ is selected, we know which relay points k must be opened to transport this truckload. If a relay point k is required to transport a truckload $t \in T'$, then all constraints (3) associated with such relay points can be omitted from the model. We found that omitting these constraints significantly reduced the number of constraints in the model, and affected the time required to obtain solutions for the TLRND problem as will be shown in Section 3.4.

3.4. Solution Methodology and Computational Experiments

In this section we present computational results obtained with our proposed formulation for both randomly generated problem instances as well as data provided by a major TL carrier. The algorithm for composite generation and our proposed formulation were implemented using Python 2.6, and all instances were solved using CPLEX 12.1 on a Xeon® 3.2 GHz workstation with 6 GB of RAM.

3.4.1. Generation of Random Instances and Selection of Parameter Values

To test the computational performance and the quality of the solutions obtained using the proposed model presented in Section 3.3.1, we generated random problem instances. In particular, we generated *complete networks* – networks where each node pair is connected by an arc in both directions – of different sizes and truckload density. Although networks that arise in practice tend to be sparser, using complete networks allowed us to test the worst case

performance of our proposed formulation due to the significant number of possible routes that can be generated between any node pair in the network. This is particularly important given the enumeration-based procedure we use to generate our variables described in Section 3.3.2.

One of the objectives of our computational experiments was to assess the effect of network size on the performance of our model. Ten instances of networks with 50, 100 and 150 nodes each were generated by randomly locating uniformly distributed nodes in a squared region of 600 miles \times 600 miles. The size of the region in which our networks were created was selected to represent the geographical area covered by a regional network (e.g., Northeast US region) for a major TL carrier. However, the distances on the arcs between node pairs in our complete networks were computed using the Euclidean norm and do not represent actual over the road distances.

Another goal of our computational testing was to determine the effect of different truckload flow densities (i.e., demand levels) on the performance of our formulation and on the quality of the solutions found. For this reason, truckloads (i.e., origin-destination node pairs) were randomly selected to cover 10%, 20% and 40% of all pairs of nodes in the network; that is, for a network with 50 nodes, we selected 245, 490 and 980 origin-destination (O-D) node pairs to have truckload flows. The actual demand (i.e., number of truckloads shipped between a selected O-D node pair) was determined by randomly generating a uniformly distributed integer between 10 and 20.

In our testing, we assume each node in the network can serve as a RP. The fixed cost of installation of a relay point was set to \$10,000. This value does not represent a considerable capital expense, but it is reasonable in the sense that RP locations are simple facilities where

drivers and loads are exchanged without using expensive equipment and infrastructure. However, we performed sensitivity analysis on the value for this parameter to determine its effect on the quality of the solutions obtained with our formulation. Also, rates per mile were established for both local and lane movements. Local miles driven were charged \$0.80 per mile, while lane movement miles were charged \$1.00 per mile.

To generate feasible routes, we allowed 25% circuitry (β) above the shortest path distance between origin and destination of a given truckload. This is a somewhat generous value considering previous research on TLRND has shown that using truckload routes with 20% circuitry results in a good compromise between company/driver performance and customer service metrics when using a network based dispatching system for TL transportation (Taylor et al., 1999). The limitations on the distances covered by local (γ_1) and lane (γ_2) drivers were set to 150 miles and 600 miles respectively. The latter value accounts for the distance that can be covered in a full work day by a single driver according to current hour-of-service regulations for the industry.

3.4.2. *Exact Solution Method*

In our testing of the proposed model, exact solutions were obtained using CPLEX 12.1 (i.e., a branch-and-cut approach) with model preprocessing enabled and other settings set to default values. To assess the tractability of our model, we initially relaxed the equipment balance constraints (4) and (5) and ran computational experiments using networks with 50 and 100 nodes. Table 8 shows the results obtained with our model after omitting facility location constraints as described in Section 3.3.3. Our customized preprocessing technique resulted in an average reduction on the number of constraints of 10.26% and an average reduction in solution

time of 20.54%. The values presented in Table 8 correspond to averages for 10 replications, except where otherwise noted.

Table 8. Exact Solution Results.

| # Nodes | # Truckloads | # Composites | # Constraints | Solution Value (\$) | # RPs Open | Setup Time (secs) | Solution Time (\$) |
|---------|--------------|--------------|---------------|---------------------|------------|-------------------|--------------------|
| 50 | 245 | 30,812.9 | 3,189.3 | 1,344,387.3 | 22.9 | 38.07 | 0.59 |
| | 490 | 64,224.3 | 6,314.1 | 2,515,949.8 | 27.0 | 78.31 | 1.49 |
| | 980 | 127,840.2 | 11,505.0 | 4,752,742.7 | 31.2 | 165.76 | 2.08 |
| 100 | 990 | 782,406.7 | 25,138.2 | 4,719,260.5 | 39.3 | 1,932.10 | 114.61 |
| | 1,980 | 1,535,146.3 | 49,052.5 | 8,836,048.3 | 50.1 | 5,472.60 | 451.58 |
| | 3,960 | 3,010,949.8* | 98,115.3* | 17,433,227.15* | 60.8* | 16,626.70* | 719.27* |

* CPLEX out of memory in 6 of the 10 replications.

As observed in Table 8, the number of composite variables increases significantly as both the number of nodes and truckload density increase in our problem instances. This has an effect on the tractability of our proposed formulation as observed in the case for problems with 100 nodes and 3,960 truckloads where only 4 replications were solved before CPLEX ran out of memory due to problem size. Based on these observations, it is expected that similar memory issues will also appear when attempting to solve instances with more than 100 nodes given the significant increase on the number of composites used in our formulation. This will prevent us from solving practically sized problem instances of TLRND using this exact approach. For this reason, alternative solution methods for largely sized instances of our problem are explored later in Sections 3.4.2.2 and 3.4.3.

In a similar way, the large number of composites also affects setup time and solution time. Setup time is the time required to generate composite variables and build the mathematical model, while solution time is the time CPLEX takes to solve the problem. From this table we observe that the generation of the composites represents a significantly larger computational effort than the solution of the problem itself. Setup times seem to be mostly driven by the

number of composites and do not increase linearly, while solution times seem to depend on both number of composites and number of nodes and increase in a nonlinear fashion as well. Although solution times do not exceed 12 minutes for networks with 100 nodes, having a huge number of composites produces setup times that exceed 4 hours and 30 minutes when truckload demand is high.

In terms of the solutions found, Table 8 shows that, as expected, the objective function value increases as the size of the network and the truckload demand increase. The proportion of the fixed cost in the total cost for the design of the RN decreases as the size of the problem instance increases. This is because for networks of the same size, the number of RPs that need to be opened does not increase in the same proportion as the truckload density. However, the large number of RPs observed can be explained by the low capital investment needed for the installation of the RPs (i.e., \$10,000 per RP). An analysis of the effect of this value on the solutions found is presented later in Section 3.4.3. Another important observation about the solutions found is that several origin/destination nodes in the network are assigned to more than one existing RP in their vicinity. This has been identified by other researchers as an open area in TLRND optimization. Since our approach is already taking this aspect into account, this can be considered an additional contribution of our research.

3.4.2.1. Strength of Formulation

Because the number of composite variables in our formulation can hamper the tractability of our model using the standard branch-and-cut approach implemented in CPLEX, we explored alternative solution methods for largely-sized instances of TLRND. However, we first analyzed the strength of our formulation to evaluate the potential use of other exact and heuristic solution

approaches. Table 9 shows that the results found using the linear programming (LP) relaxation of our model without balance constraints are very close to the solutions found with the integer program (IP) for networks with 50 nodes. Average and worst case optimality gaps below 1% are an indication that our formulation has the potential to find very high quality solutions for many problem instances even when the integrality of our variables is relaxed. Furthermore, best case optimality gaps of 0.0% indicate that in all cases at least one problem was solved to optimality with the LP relaxation of our model.

Table 9. Strength of Formulation of Composite Variable Model for TLRND (50 Nodes).

| # Truckloads | IP Solution Value (\$) | LP Relaxation Solution Value (\$) | Average Optimality Gap | Best Case Optimality Gap | Worst Case Optimality Gap |
|--------------|---------------------------|--------------------------------------|---------------------------|-----------------------------|------------------------------|
| 245 | 1,344,387.3 | 1,341,862.4 | 0.19% | 0.00% | 0.37% |
| 490 | 2,515,949.8 | 2,512,951.8 | 0.12% | 0.00% | 0.24% |
| 980 | 4,752,742.7 | 4,751,060.8 | 0.04% | 0.00% | 0.08% |

The values presented in Table 9 encouraged us to pursue both exact and heuristic solution methods that take advantage of the very small optimality gaps observed while reducing the memory requirements of our formulation. In particular, we decided to employ an exact pricing method, and a heuristic method based on using a subset of the composite variables to find very high quality solutions in reasonable CPU times. The pricing method is presented next (Section 3.4.2.2) and the heuristic method is presented in Section 3.4.3.

3.4.2.2. Pricing Method for Solving TLRND

Given the tractability issues observed for largely-sized instances of TLRND and the small optimality gaps seen for solutions obtained with the LP relaxation of our model, we tested a branch-and-price framework to solve this problem. Starting with the LP relaxation of our model and using only a subset of the composite variables at the root node of the branch-and-bound tree,

we implemented a pricing method to evaluate the reduced costs of composites not included in the model, and added those that improve the objective function value. This pricing method is also used at the other nodes of the branch-and-bound tree until optimality can be proven. In our implementation of the branch-and-bound tree, we decided to branch on the most fractional variable associated with a relay point (y variables) and used depth-first search to traverse the tree.

The most significant challenge associated with this solution method was determining the appropriate pricing method for the composites. One alternative is to develop a mathematical model for the pricing subproblem. This resulted in a formulation similar to the existing model for TLRND presented in previous literature with balance constraints relaxed (Üster and Maheshwari, 2007; Üster and Kewcharoenwong, 2011). Since there are tractability issues with this model, we decided to use a different approach for the pricing of the composites. Our pricing method uses the templates previously defined for composite generation (Figure 11) to evaluate the reduced cost of composites not included in the reduced model. Let ω_t and ω_{tk}^0 denote the dual values for constraints (2) and (3) respectively. The reduced cost for a given composite is shown in equation (8) where α_{tr} indicates if a composite r corresponds to a truckload t .

$$\bar{c}_r = c_r - \sum_{t \in T} \omega_t \alpha_{tr} + \sum_{t \in T} \sum_{k \in N} \omega_{tk}^0 \theta_{kr} \quad (8)$$

Using the templates, equation (8) is evaluated for each composite not included in the reduced model and whenever the value obtained is negative, the composite can improve the objective function value and is therefore included in the model as a new column. Since our facility location constraint (3) is defined only for those nodes visited by composites associated with a truckload (i.e., $\forall t \in T, k \in N_r : r \in R_t$), the addition of new composites may also result in the generation of new rows in the model if new nodes are included in routes for a given

truckload. For this reason, our solution methodology can be classified as a branch-and-price-and-cut approach.

Although this approach solves the memory issues by reducing the size of the problems and allows us to provide an optimality guarantee for the solutions obtained, it introduces a significant computational burden in terms of the CPU time required to obtain an optimal solution. Table 10 shows the performance of this exact method for networks with 50 nodes. The values in parentheses below the averages shown in this table represent the changes with respect to the values previously obtained for networks with 50 nodes (Table 8). It is important to note the significant reduction in the number of composite variables generated with this exact approach (i.e., more than 86% reduction) and how this reduction in problem size helps to alleviate the memory issues previously observed. However, the savings in terms of the reduction in the number of composites and the associated setup times are completely lost with the significant increase in solution times for the algorithm. Although the number of nodes required in the branch-and-bound tree is relatively small (i.e., between 5 and 12 on average), using the templates to evaluate reduced costs for the composites represents a considerable computational effort similar to completing several setups during the pricing of the composites. This does not allow us to find optimal solutions for largely-sized instances of TLRND in reasonable CPU times. Even though we were able to solve problems with 50 nodes and 980 truckloads in just over 10 minutes, one instance with 100 nodes and 990 loads was stopped after it ran for 16 hours without obtaining the optimal solution. For this reason, we developed the heuristic method that is presented in Section 3.4.3, and used this method to solve the remaining problem instances.

Table 10. Pricing Method Results (50 Node Networks).

| # Truckloads | # Initial Composites | # Final Composites | # Constraints | Setup Time (secs) | # B-and-B Tree Nodes | Solution Time (secs) |
|--------------|-----------------------|-----------------------|----------------------|--------------------|----------------------|------------------------|
| 245 | 3,968.4 (-86.94%) | 4,155.7 (-86.29%) | 2,357.7 (-26.02%) | 11.67 (-69.22%) | 8.4 | 176.53 (+35,118.0%) |
| 490 | 8,098.8 (-87.29%) | 8,492.5 (-86.68%) | 4,713.5 (-25.27%) | 23.17 (-70.24%) | 11.1 | 449.82 (+36,118.7%) |
| 980 | 16,271.1 (-87.16%) | 16,755.0 (-86.77%) | 9,369.6 (-18.08%) | 46.22 (-71.81%) | 5.4 | 549.85 (+30,311.1%) |

3.4.3. Heuristic Method

Since our testing revealed that a significant number of composites hampers the tractability of our formulation, we developed a heuristic procedure to solve largely-sized instances of TLRND. In this heuristic method, we restrict our consideration only to a subset of the templates originally created to generate composites. To select an appropriate subset of the templates to be used in our heuristic approach, we analyzed the usage of each template type in the optimal solutions found for networks with 50 and 100 nodes presented in Section 3.4.2. The respective usages across all instances considered are presented in Table 11. As observed in this table, there are four template types that are rarely used – types 7, 9, 10 and 11. Based on these observations, our heuristic method only considers composite variables generated using the remaining templates shown in Figure 11 (i.e., types 1, 2, 3, 4, 5, 6 and 8) which only generate truckload routes with up to two relay points.

While omitting templates does not allow us to guarantee that we will obtain optimal solutions, our testing indicated that the solution quality suffered either very little or not at all while the performance of the model improved significantly, as shown in Table 12. The values presented between parentheses below the averages shown in this table correspond to the variations with respect to the exact solution values previously obtained for networks with 50

nodes presented in Table 8. We observe that using a subset of the composites results in very high quality solutions for 50 node networks with average and worst case optimality gaps well below 1%. Even more, optimal solutions were obtained in at least half of the replications for each truckload density used.

Table 11. Usage of Composites in Optimal Solutions by Template Type.

| Template Type | # Composites Used | Percentage of Total |
|---------------|-------------------|---------------------|
| 1 | 3,063 | 4.89% |
| 2 | 4,969 | 7.93% |
| 3 | 844 | 1.35% |
| 4 | 14,374 | 22.93% |
| 5 | 2,981 | 4.76% |
| 6 | 2,835 | 4.52% |
| 7 | 7 | 0.01% |
| 8 | 33,578 | 53.56% |
| 9 | 10 | 0.02% |
| 10 | 8 | 0.01% |
| 11 | 21 | 0.03% |
| Total | 62,690 | 100% |

Table 12. Heuristic Method Results (50 Node Networks).

| # Truckloads | # Composites | # Constraints | Solution Value (\$) | # RPs Open | Setup Time (secs) | Solution Time (secs) | Worst Case Opt. Gap | Optimal Replications |
|--------------|-----------------------|----------------------|--------------------------|----------------|--------------------|----------------------|---------------------|----------------------|
| 245 | 4,203.8 (-86.15%) | 2,197.2 (-31.18%) | 1,344,707.3 (+0.24%) | 23.0 (+0.1) | 3.64 (-90.40%) | 0.22 (-61.73%) | 0.15% | 5 out of 10 |
| 490 | 8,579.7 (-86.52%) | 4,334.9 (-31.35%) | 2,516,115.6 (+0.007%) | 27.1 (+0.1) | 7.25 (-90.69%) | 0.33 (-79.89%) | 0.50% | 6 out of 10 |
| 980 | 17,232.4 (-86.39%) | 8,307.9 (-30.12%) | 4,752,952.7 (+0.004%) | 31.2 (-) | 14.41 (-91.22%) | 0.43 (-79.30%) | 0.22% | 6 out of 10 |

Our results show that the slight increase in objective function value is observed in conjunction with a significant reduction in problem size and CPU times. Most of the benefits of our heuristic method are obtained by reducing the number of composites generated (> 86% reduction) which significantly reduces the setup times (> 90% reduction). Solution times are also reduced significantly, though they vary more than when all template types are generated.

In terms of solution quality, there is no noticeable change in the number of RPs that are opened relative to the exact solutions previously found. The relatively high number of RPs that are selected by the model can be explained in part by the low fixed cost to open RPs in our experiments. However, another reason for this is the service requirements that affect the feasibility of the network design. In particular, when truckloads originate or terminate at isolated nodes in the network, RPs must be opened to accommodate such truckload. Local and lane distance limitations can also force the model to open more than one RP in the same general area. This behavior was evident when we varied the fixed cost of installation of the RPs. We observed that an increase of 100 times in magnitude for the fixed cost only produced an average reduction in the number of open RPs of 15%. This indicates that feasibility of the routes plays a significant role in the quality of the solutions found with our model given the limitations on local and lane distances and circuitry.

To complement our analysis of the solutions obtained, we recorded the average and maximum circuitry observed for the selected truckload routes and the truckload usage of the open RPs. For networks with 50 nodes, the average circuitry of the selected truckload routes is 3.43%, while the maximum circuitry observed is 24.98%. This indicates that while the majority of the truckloads are assigned routes with very low circuitry, there are a few with very circuitous routes that are close to the maximum allowable circuitry. Most of these outliers correspond to truckloads with origin and destination nodes that are not very apart in distance but since they are not selected as RPs need to be routed through at least one other node in the network. These truckloads are actually good candidates for PtP dispatching. With respect to RP usage, we observe that although a few RPs handle a significant number of truckloads, most of the selected

RPs handle only a moderate to reduced number. This is evidence that a non-trivial number of RPs are underutilized and mostly exist for feasibility reasons.

The large number of required RPs and the existence of highly circuitous routes for relatively close O-D truckload pairs reinforce some of the conclusions presented in previous research related to TLRND that indicate that operating a hybrid system would be a more appropriate approach for the implementation of a network structure for TL transportation dispatching. In a hybrid system, some of the truckloads would be dispatched using the relay network while the rest would be sent PtP. In this way, a compromise between increased tour regularity (and consequently reduced driver turnover) and reduced cost of installation and operation of the dispatching system is likely to be obtained. Further thoughts on this are presented when future research is discussed.

3.4.3.1. Effect of Instance Size

Using our heuristic approach, we were able to solve bigger instances of TLRND that were previously intractable. We are now able to obtain solutions for all ten replications of networks with 100 nodes and 3,960 truckloads requiring an average running time of less than 5 minutes for each instance. This can be contrasted to the approximately 4 hours and 50 minutes previously required for each replication by the exact method when solving only four out of ten instances as described in Section 3.4.2. Moreover, we can use the heuristic to solve even larger problem instances of TLRND with 150 nodes and 8,940 truckloads (40% truckload density) in less than 45 minutes on average. Figure 12 shows the effect of instance size in the results obtained and the performance of our solution method. The values presented in Figure 12 are averages obtained for ten replications of networks with 50, 100 and 150 nodes.

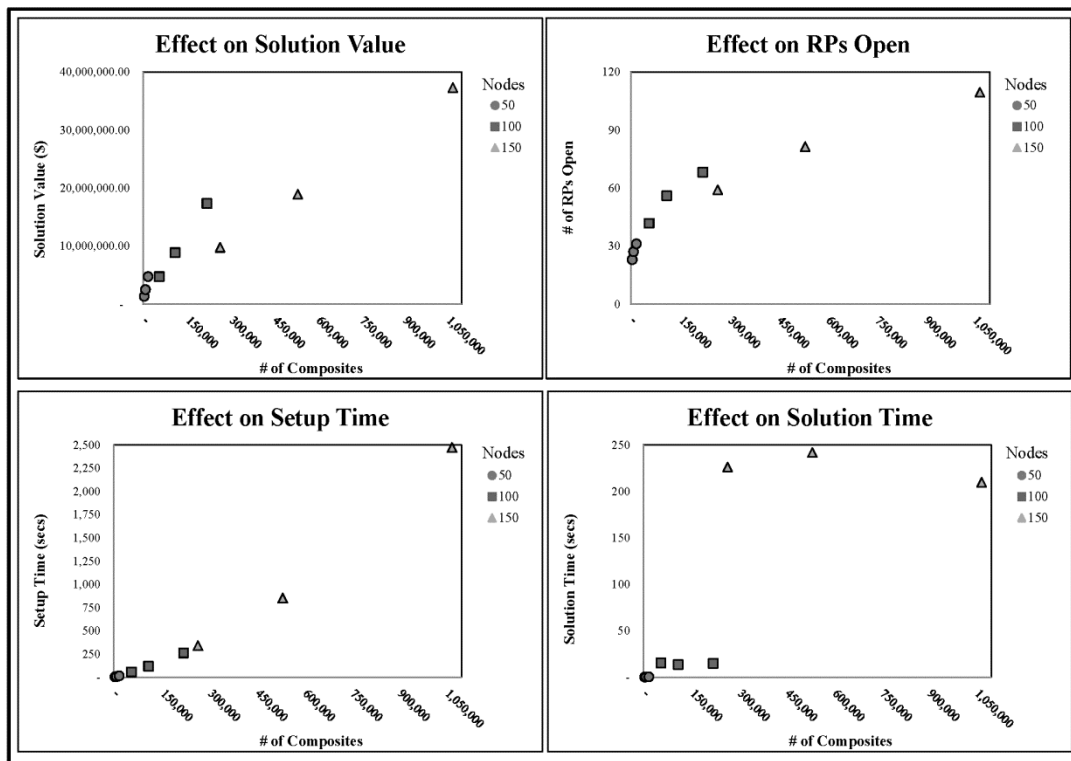


Figure 12. Effect of Instance Size on Solution Quality and Model Performance.

As shown in Figure 12, solution value and number of RPs open increase as the number of composites increases for networks of the same size (i.e., as truckload density increases). The size (i.e., number of nodes) of the network also seems to have an effect in these two quantities, especially for larger networks. It is also evident that setup time is directly affected by the number of composites in the model. There is an exponential growth in setup times as the number of composites increases regardless of the size of the network. On the other hand, solution times for the model seem to be affected only by the number of nodes in the network. Networks with 150 nodes are all solved in less than 5 minutes regardless of the number of loads. However, these instances take significantly longer to solve than instances that occur in networks with 50 and 100 nodes; these are solved in less than a minute. The significantly higher solution times for

largely-sized networks can be explained by the fact that the model has to make decisions for several more nodes than before. The variability observed in solution times for networks of the same size is attributed to our customized pre-processing technique explained in Section 3.3.3. We observed that in some cases our pre-processing of the model helps to significantly reduce the size of the problem while in some other cases the effect is minimal.

3.4.3.2. Effect of Equipment Balance

Up to this point, the equipment balance constraints (4) and (5) have been relaxed from our formulation of TLRND. This actually helped us to obtain significant insights about the performance of our model. For instance, we discovered the memory issues that resulted when generating large numbers of composites and also determined that including only a subset of composites did not significantly impact solution quality. These findings allowed us to come up with the pricing method described in Section 3.4.2.2 and the heuristic approach presented in Section 3.4.3. However, we now consider the effect of balance in the quality of the solutions found and the performance of our model by incorporating again the previously relaxed constraints in our formulation.

We tested networks with 50, 100 and 150 nodes considering values of 30% and 0% for the maximum allowed percentage of imbalance (δ). A value of $\delta = 0$ means that we are requiring perfect balance at all nodes in the network. In all cases, the addition of the balance constraints increases the number of rows in our model by two times the number of nodes in the network. This is still manageable for our approach. Also, we adapted our composite generation algorithm to create composites for deadhead (i.e., unloaded truck) movements in addition to the truckload composites. This results in an increase in the number of variables in the model. Fortunately, the

inclusion of these composites did not affect the tractability of our model. Furthermore, they also allowed us to solve problems with perfect balance which have not been previously solved in the TLRND literature due to feasibility reasons. This allows us to better assess the actual cost of operating a TL relay network since those empty movements required to provide service in the network are also included when computing the solution value.

Table 13 shows the results obtained in our testing using our heuristic approach and setting an optimality gap of 1.0% in CPLEX. The comparisons with the values observed without balance are shown between parentheses underneath the averages presented in this table for problems with $\delta < 1$. As observed in this table, achieving balance in the network comes at a cost both in terms of the solution value and computational effort. However, the increase in solution value is not very significant, exceeding 1% in only a single case. However, the solutions obtained typically increase the number of RPs opened, particularly when there are more nodes in the network. Additionally, the model selects different routes and RPs in those cases to help balance the network with a minimal number of empty trailer repositioning moves. Thus, the small increment in the total cost of the network design. With respect to CPU times, there is an increase in setup times due to the fact that we have to generate feasible routes for the empty trailer movements. Since the number of these additional composites is the same regardless of the truckload volume considered in a problem instance for networks of the same size, the effect of this addition in the setup time decreases as the truckload density increases. Finally, note that incorporating equipment balance has a significant effect in the solution times for the model, especially for networks with 100 and 150 nodes. However, as observed in this table, it only

takes less than 13 minutes on average to solve perfect balance problems with 150 nodes and 8,940 truckloads.

Table 13. Results with Equipment Balance Enforced.

| Nodes | Imbalance Allowed (δ) | # Truckloads | Solution Value (\$) | # RPs Open | Setup Time (secs) | Solution Time (secs) | Average Imbalance | |
|-------|--------------------------------|--------------|--------------------------|--------------------------|----------------------|------------------------|----------------------|-------------------|
| 50 | 0.3 | 245 | 1,348,673.2 (+0.30%) | 23.0 (-) | 4.79 (+31.51%) | 0.80 (+375.40%) | 0.23 (-30.19%) | |
| | | 490 | 2,518,189.8 (+0.08%) | 27.3 (+0.2) | 9.01 (+24.33%) | 1.25 (+452.11%) | 0.22 (-23.40%) | |
| | | 980 | 4,754,359.6 (+0.03%) | 31.0 (-0.2) | 17.45 (-21.00%) | 0.86 (+169.83%) | 0.18 (-14.81%) | |
| | 0.0 | 245 | 1,374,934.1 (+2.26%) | 23.6 (+0.6) | 4.77 (+31.07%) | 1.26 (+508.57%) | 0.00 (-100.00%) | |
| | | 490 | 2,534,493.9 (+0.73%) | 28.1 (+1.0) | 9.00 (+24.19%) | 1.64 (+474.96%) | 0.00 (-100.00%) | |
| | | 980 | 4,773,788.4 (+0.44%) | 31.5 (+0.3) | 17.47 (+21.15%) | 1.00 (+194.78%) | 0.00 (-100.00%) | |
| | 100 | 0.3 | 990 | 4,735,510.4 (+0.34%) | 41.6 (+2.3) | 67.92 (+21.27%) | 15.55 (+240.45%) | 0.23 (-24.94%) |
| | | | 1,980 | 8,866,278.2 (+0.34%) | 53.0 (+2.8) | 138.60 (+18.01%) | 10.30 (+111.17%) | 0.20 (-23.08%) |
| | | | 3,960 | 17,356,404.3 (+0.37%) | 68.1 (+6.8) | 300.98 (+14.80%) | 10.79 (+51.84%) | 0.17 (-21.97%) |
| 0.0 | | 990 | 4,752,037.6 (+0.69%) | 42.0 (+2.7) | 68.08 (+21.54%) | 46.05 (+946.01%) | 0.00 (-100.00%) | |
| | | 1,980 | 8,883,005.3 (+0.53%) | 54.8 (+4.6) | 138.61 (+18.02%) | 53.49 (+979.84%) | 0.00 (-100.00%) | |
| | | 3,960 | 17,354,380.9 (+0.36%) | 67.6 (+6.3) | 300.69 (+14.68%) | 38.46 (+442.42%) | 0.00 (-100.00%) | |
| 150 | | 0.3 | 2,235 | 9,774,214.6 (+0.23%) | 58.9 (+3.1) | 396.97 (+16.84%) | 156.39 (+341.71%) | 0.21 (-18.69%) |
| | | | 4,470 | 18,978,074.4 (+0.57%) | 87.3 (+13.4) | 964.16 (+13.28%) | 91.03 (+94.21%) | 0.21 (-14.91%) |
| | | | 8,940 | 37,317,107.6 (+0.47%) | 109.6 (+19.5) | 2,684.44 (+8.56%) | 82.77 (+19.35%) | 0.16 (-33.01%) |
| | 0.0 | 2,235 | 9,811,407.2 (+0.62%) | 63.8 (+8.0) | 397.53 (+17.00%) | 437.66 (+1,263.52%) | 0.00 (-100.00%) | |
| | | 4,470 | 19,002,880.2 (+0.70%) | 88.7 (+14.8) | 965.20 (+13.41%) | 496.78 (+851.14%) | 0.00 (-100.00%) | |
| | | 8,940 | 37,184,483.5 (+0.15%) | 95.1 (+5.2) | 2,676.02 (+8.47%) | 690.53 (+872.09%) | 0.00 (-100.00%) | |

3.4.4. Test Case from a Major TL Carrier

In addition to the computational results obtained for randomly generated instances of TLRND, we were able to test our proposed formulation using real data from J.B. Hunt, one of

the nation's largest TL carriers. We consider a network of origin and destination nodes that encompasses the eastern half of the United States. Figure 13 shows the location of the 623 origin and destination nodes included in this area, hereafter called the East U.S. Network. Each of these nodes marks the location of the centroid for the latitude and longitude coordinates of origin/destination points that exist in a three-digit level ZIP code area.

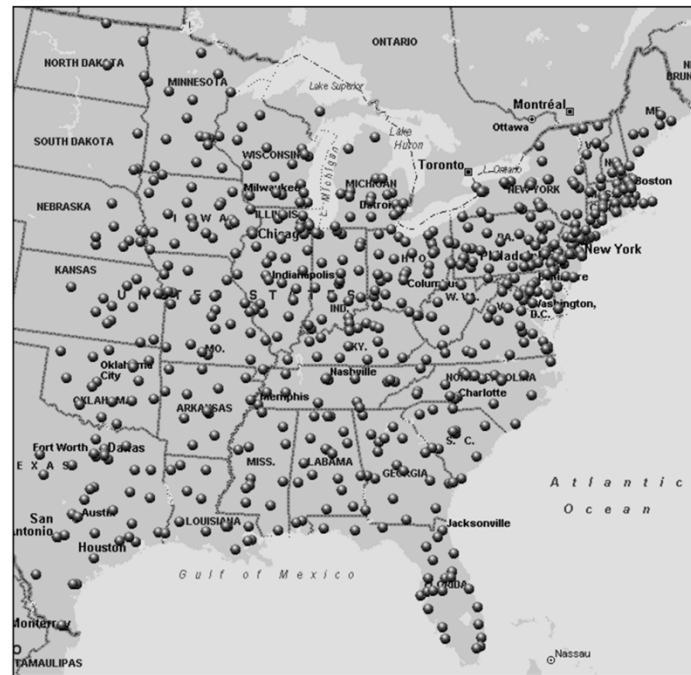


Figure 13. East U.S. Network for a Major TL Carrier.

In addition to the node information, we also obtained truckload and road distance information for some of the arcs in this network from the carrier. After consolidating and analyzing the truckload information, we decided to use all truckload origin-destination pairs with five or more truckloads. The resulting 1,529 O-D pairs included in this subset account for 77.5% of the total truckload volume in the East U.S. Network. We used MapPoint to obtain distances between pairs of nodes in the network that were not included in the arc set provided by the

carrier. This allowed us to consider a denser network that more closely resembles the U.S. Highway System in the eastern part of the country (i.e., a network with 21.25% arc density).

Similar to our testing with randomly generated instances of TLRND, we assumed that each node in the network is a potential RP. We considered fixed costs of \$10,000, \$100,000, \$200,000, \$500,000 and \$1,000,000 to assess the effect of RP installation costs on the characteristics of the solutions found. The rates per mile for both local and lane movements were established using estimates provided by the carrier. Local miles were charged \$1.00 per mile, while lane miles were charged \$1.30 per mile. We maintained a limitation of 25% for the allowable circuitry (β) for truckloads when enumerating feasible routes and the limitations on the distances covered by local (γ_1) and lane (γ_2) drivers were set to 225 miles and 450 miles respectively. The value of γ_2 represents the distance that can be covered in a full work day by a single driver when including an intermediate rest according to current hours-of-service regulations.

We used the heuristic method described in Section 3.4.3 to solve this test case with equipment balance constraints enforcing perfect balance (i.e., $\delta = 0$). A preliminary assessment of the feasibility of the problem determined that considering our templates and the limitations enforced on circuitry and local and lane distances no composites exist for 143 of the 1,529 O-D pairs with truckloads. We can assume that the demand for these 143 O-D pairs can be dispatched direct point-to-point and design the relay network for the remaining 1,386 truckloads with five truckloads or more.

Table 14 shows the results obtained for different values of fixed cost of installation of the RPs after setting an optimality gap of 1.0% in CPLEX. From the values observed in Table 14, it

is clear that the number of RPs that are open is only affected by the fixed cost of installation when this amount is less than \$100,000. This means that in many cases RPs are open only due to feasibility requirements for the truckload routes. The increase from 116 to 117 RPs that are open when increasing the fixed cost from \$500,000 to \$1 million can be attributed to CPLEX stopping after finding the first integer solution that is within the specified optimality gap. As observed in Table 14, the portion of the total cost that corresponds to loaded miles and empty miles to be driven does not vary significantly. This is another indication that the basic design of the relay network remains the same regardless of the fixed cost of installation of the RPs. Figure 14 shows an illustration of the solutions with the most and least RPs open. The circles around the nodes represent open RPs and their size is proportional to the truckload volume handled at the RPs. It can be seen that many RPs only handle a small number of truckloads and in other cases there are more than two RPs that are open in the same general area. These results enforce the idea that a hybrid system where some of the truckloads are dispatched direct point-to-point while others are sent through the relay network is a very appealing alternative for a dispatching system in TL transportation. The reason for this is that having a mixed fleet dispatching system would allow the carrier to benefit from the reduction in driver turnover since most of the demand would be routed through the relay network while reducing costs of operation by sending only a portion of the truckloads directly from origin to destination.

Table 14. Heuristic Method Results for East U.S. Network (623 Nodes and 1,386 Truckloads).

| Fixed Cost (\$) | Solution Value (\$) | # RPs Open | Cost Loaded Miles (\$) | Cost Empty Miles (\$) | Avg. Circuity | Avg. RP Usage | Setup Time (secs) | Solution Time (secs) |
|-----------------|---------------------|------------|------------------------|-----------------------|---------------|---------------|-------------------|----------------------|
| 10,000 | 19,986,727 | 136 | 18,124,691 | 502,036 | 4.54% | 488.65 | 11,509.67 | 989.38 |
| 100,000 | 30,729,878 | 118 | 18,339,008 | 590,870 | 5.61% | 545.11 | 11,580.90 | 1,156.96 |
| 200,000 | 42,575,465 | 118 | 18,355,652 | 619,814 | 5.63% | 545.28 | 11,547.63 | 1,192.89 |
| 500,000 | 77,080,987 | 116 | 18,382,760 | 698,227 | 5.75% | 560.97 | 11,591.93 | 1,307.86 |
| 1,000,000 | 135,973,174 | 117 | 18,372,212 | 600,962 | 5.70% | 554.63 | 11,606.78 | 1,045.40 |

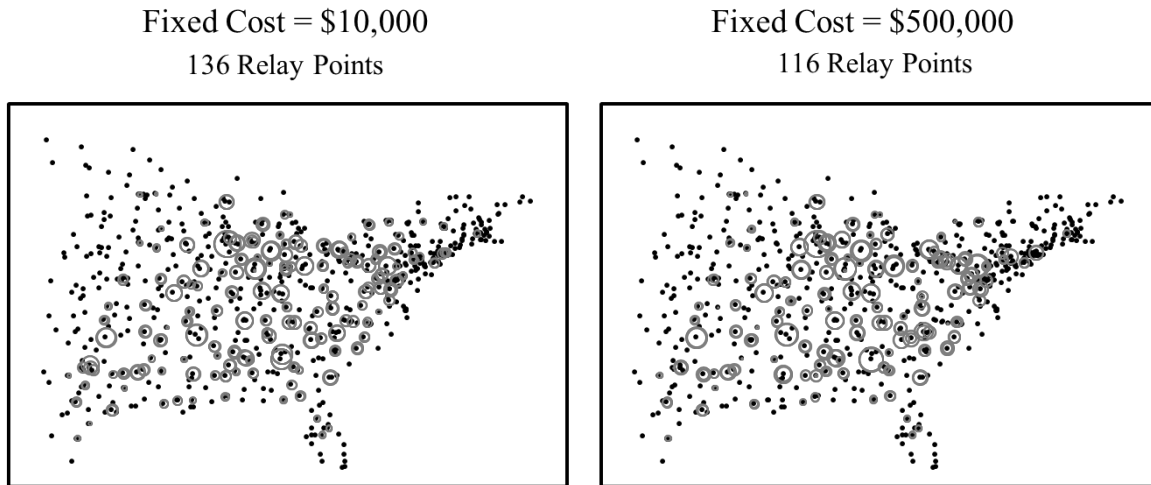


Figure 14. Test Case Solutions for Different Fixed Costs of Installation.

Finally, it is important to note that even though this test case represents a very large problem instance for a major TL carrier (i.e., more than 2.2 million variables and 60,364 constraints), our heuristic method is able to obtain a solution in reasonable time, particularly for a strategic network design problem. As observed in our previous testing, the huge number of composites affects significantly the setup time for the model. While solution times are in turn slightly affected by the magnitude of the fixed cost of installation of RPs. In this case, setup time is close to 3 hours and 15 minutes while CPLEX takes between 16 and 23 minutes to solve the test case instance for different values of the fixed cost of installation of RPs.

3.5. Conclusions and Future Research Directions

In this chapter, we presented a composite variable model for TLRND that implicitly captures difficult operational constraints such as truckload circuitry and distance limitations for drivers within the definition of our variables. This new modeling approach provides a stronger formulation for TLRND than existing models in the literature. Exact solutions can be obtained for small and medium sized problem instances using standard branch-and-cut as implemented in

CPLEX. However, given the very large number of composite variables needed in our proposed formulation, we developed both an exact pricing method and a heuristic method to solve largely sized problem instances of TLRND. Although our exact pricing method provides a guarantee of optimality, its main drawback is that the pricing of the composites takes a significant amount of time and makes the use of this method impractical for medium and largely sized problem instances. On the other hand, our computational testing showed that the solutions found with a heuristic method that uses a subset of the composites are very high quality with average optimality gaps below 1% for networks with 50 nodes. As a result, we were able to efficiently solve largely sized problem instances of TLRND with up to 150 nodes using the heuristic approach. In addition, we successfully used our heuristic method to obtain solutions for a problem with real truckload data provided by a major TL carrier that operates nationally and obtain insights about relay network design and TL dispatching.

The computational performance of our proposed formulation is significantly affected by the number of composite variables in the model. Once the parameters for circuitry and local and lane distances are established, the number of composites depends on the number of nodes in the network and the demand for truckloads. The large number of composites directly affects the time needed to setup our model which includes generating our variables using templates and also building the mathematical model. This results in setup time being a very significant portion of the total time needed to solve the TLRND problem. On the other hand, solution times seem to be mostly affected by the number of nodes in the network given that for bigger instances there are more opportunities to locate RPs. Also, solution times are significantly affected by enforcing equipment balance at the RPs given the additional difficulty associated with selecting nodes and

routes to balance the network. We additionally observed that solution times are variable due to the effect of the customized pre-processing method that we implemented to reduce the size of the model.

The relaxation of the equipment balance constraints in our formulation allowed us to observe that enforcing this important operational restriction comes at a cost in terms of solution value and computational time. The effect of balance on solution cost is not very significant which indicates that the initial design of the relay network actually facilitates attaining balance when enforced. Also, adding composites for empty trailer movements allowed us to better assess the cost of designing a balanced network. However, most importantly, the addition of these constraints also showed us that the tractability of our model is not affected by the addition of other restrictions in our basic network design formulation. The flexibility of our proposed CVM for TLRND will allow us to incorporate extensions and better assess the effect of other realistic aspects of TL dispatching when designing the relay network.

Looking at the solutions found with our proposed formulation, it is interesting to observe that the model selects a high number of nodes to be relay points. Analysis of the solutions obtained when varying this fixed cost led us to observe that many RPs are selected for feasibility reasons in both randomly generated and real data problem instances. Even very high fixed costs do not result in a significant reduction in the number of open RPs. In addition, there is evidence of the existence of a very few number of highly circuitous routes for truckloads with origin and destination nodes that are not very distant apart. Both of these observations indicate that if the network is to be setup with less RPs and reduced additional miles to avoid excessive installation and operation costs then some of the truckloads should still be dispatched point-to-point as

opposed to through the relay network. The development of a prescriptive model for the strategic design of the relay network when using this mixed fleet dispatching system is an interesting area for future research. In the past, the determination of which truckloads should be dispatched PtP has been completed a priori or after finding a solution for the TLRND problem. A mathematical model that integrates this tactical decision would be very helpful in the process of designing a TL dispatching system that would better exploit the tradeoff between improved quality of life for the drivers and additional costs of operation, thus making a network configuration for TL dispatching more applicable in practice.

A direct extension of our research is the analysis of the allocation of drivers to RPs and their assignment to truckloads based on the design of the dispatching system. Although previous studies have shown that relay configurations make it significantly easier to return drivers to their home domiciles due to increased tour regularity, existing models do not consider this driver balance explicitly. Instead, this operational problem is left to the carriers. Some of the challenges associated with obtaining driver balance relate to enforcing hours-of-service regulations for each work day, and limitations on how long drivers can be away from their home domiciles. Actually, considering driver balance explicitly will help to finally assess from an economic perspective the value of a relay network configuration for TL transportation as compared to the existing PtP dispatching system that is used in the industry.

Finally, based on our experience with composite variable modeling for this problem, we would like to continue exploring other integrated decision problems in truckload transportation and logistics that would benefit from a stronger formulation in order to obtain very high quality solutions for realistically-sized problem instances found in practice.

Acknowledgments

We gratefully thank J.B. Hunt Transportation Services for providing data for the test case presented in this research. We also acknowledge the work of Sergio Maldonado who completed an empirical analysis of the factors affecting the TLRND problem for his undergraduate honors thesis. The insights produced were very helpful as we completed our experiments. Finally, we thank Dr. Chase Rainwater for reviewing our proposed formulation and providing helpful suggestions.

References

- Ali, T., Radhakrishnan, S., Pulat, S., and Gaddipati, N. (2002). Relay Network Design in Freight Transportation Systems. *Transportation Research Part E*, 38, 405-422.
- American Transportation Research Institute. (2011). *Critical Issues in the Trucking Industry - 2011*. Arlington, VA: American Transportation Research Institute (ATRI).
- Armacost, A., Barnhart, C., and Ware, K. (2002). Composite Variable Formulations for Express Shipment Service Network Design. *Transportation Science*, 36(1), 1-20.
- Cabral, E., Erkut, E., Laporte, G., and Patterson, R. (2005). The Network Design Problem with Relays. *European Journal of Operational Research*, 180(2), 834-844.
- Campbell, J. (2005). Strategic Network Design for Motor Carriers. In *Logistics Systems - Design and Optimization* (pp. 245-278). New York, NY: Springer-Verlag.
- Cohn, A., and Barnhart, C. (2003). Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions. *Operations Research*, 51(3), 387-396.
- Cohn, A., and Barnhart, C. (2006). Composite-Variable Modeling for Service Parts Logistics. *Annals of Operations Research*, 144, 17-32.
- Cohn, A., Root, S., Wang, A., and Mohr, D. (2007). Integration of the Load-Matching and Routing Problem with Equipment Balancing for Small Package Carriers. *Transportation Science*, 41(2), 238-252.

- Hunt, G. (1998). *Transportation Relay Network Design*. Atlanta, GA: Ph.D. Thesis, Georgia Institute of Technology.
- Konak, A. (2011). Network Design Problem with Relays: A Genetic Algorithm with Path-Based Crossover and a Set Covering Formulation. *European Journal of Operational Research*, doi:10.1016/j.ejor.2011.11.046.
- Lohatepanont, M., and Barnhart, C. (2004). Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. *Transportation Science*, 38(1), 19-32.
- Morris, F. (2011). *Thousands of Trucking Jobs, But Few Take the Wheel*. Retrieved from <http://www.npr.org/2011/10/31/141791952/thousands-of-trucking-jobs-but-few-take-the-wheel>.
- Rodriguez, J., Kosir, M., Lantz, B., Griffin, G., and Glatt, J. (2000). *The Costs of Truckload Driver Turnover*. Fargo, ND: Upper Great Plains Transportation Institute.
- Root, S., and Cohn, A. (2008). A Novel Modeling Approach for Express Package Carrier Planning. *Naval Research Logistics*, 55, 670-683.
- Taha, T., and Taylor, G.D. (1994). An Integrated Modeling Framework for Evaluating Hub-and-Spoke Networks in Truckload Trucking. *Logistics and Transportation Review*, 30(2), 141-166.
- Taylor, G.D., Harit, S., and English, J.R. (1995). Hub and Spoke Networks in Truckload Trucking: Configuration and Operational Concerns. *Logistics and Transportation Review*, 31(2), 209-237.
- Taylor, G.D., and Meinert, T.S. (2000). Improving the Quality of Operations in Truckload Trucking. *IIE Transactions*, 32, 551-562.
- Taylor, G.D., Meinert, T.S., Killian, R.C., and Whicker, G.L. (1999). Development and Analysis of Alternative Dispatching Methods in Truckload Trucking. *Transportation Research Part E*, 35, 191-205.
- Taylor, G.D., Whicker, G.L., and Usher, J.S. (2001). Multi-Zone Dispatching in Truckload Trucking. *Transportation Research: Part E*, 37, 375-390.
- U.S. Bureau of Transportation Statistics. (2010). *Transportation: 2007 Commodity Flow Survey*.
- Üster, H., and Kewcharoenwong, P. (2011). Strategic Design and Analysis of a Relay Network in Truckload Transportation. *Transportation Science*, 45(4), 505-523.

Üster, H., and Maheshwari, N. (2007). Strategic Network Design for Multi-Zone Truckload Shipments. *IIE Transactions*, 39, 177-189.

4. Mixed Fleet Dispatching in Truckload Relay Network Design Optimization

A network of relay points where drivers and trailers are exchanged can be used to transport full truckloads from origin to destination. This relay network system allows full truckload carriers to reduce driver tour lengths and improve driver retention as compared to the traditional point-to-point dispatching method that is used by the industry. Previous research suggests that combining a relay network and point-to-point dispatching into a hybrid system using a partitioned fleet can result in better performance for carriers, drivers and shippers. We propose a mathematical formulation for strategic relay network design and dispatching method selection that incorporates important operational constraints such as load circuitry and driver tour length within the variable definition. High quality solutions for largely-sized problem instances can be obtained in reasonable times using this formulation. Computational results for randomly generated data and a test case from a major carrier are analyzed to develop insights about the effect of a mixed fleet dispatching system on the design of the relay networks and performance metrics of the system.

4.1. Introduction

Driver retention has been a significant problem for full truckload (TL) transportation carriers for a very long time. The American Transportation Research Institute (2011) reports that driver retention has ranked among the top three concerns for the industry in five of the last seven years, and it has never placed lower than sixth during this period. Driver turnover rates for TL carriers have historically exceeded 100%, and while the reported rate for the third quarter of 2011 is 89%, this value is the highest of the last four years (American Trucking Associations, 2011). In fact, the estimated shortage of truck drivers during 2011 rose to 125,000, exceeding

previous estimates (Morris, 2011). In this scenario, driver turnover not only represents a significant expense for the industry in terms of hiring and retention costs which are estimated at an average of around \$8,200 per driver (Rodriguez et al., 2000), but this problem also negatively affects level of service to shippers and results in lost income from demand that cannot be satisfied (Keller and Ozment, 1999). In reality, given that trucking is not a self-contained industry; the shortage of drivers is able to produce negative economic impacts that go far beyond the boundaries of the industry (American Trucking Associations, 2007).

One of the main factors commonly associated with high driver turnover corresponds to the characteristics of the operation of TL carriers that use a direct point-to-point (PtP) dispatching method to satisfy demand. Long-haul drivers who spend a significant amount of time away from home often quit due to a low quality of life perception (Gupta et al., 1996; Lockridge, 2008). In a PtP system, carriers assign a truckload to a single driver who is in charge of transporting it all the way from origin to destination. After the load is delivered, the carrier needs to find a new truckload for the driver with an origin close to the drop-off location in order to minimize empty miles. Finding appropriate backhaul trips for these drivers is a very difficult task, and usually several consecutive repositioning moves are needed to return a driver to his or her domicile. Other transportation industries such as less-than-truckload (LTL) and express package shipping that use a hub-and-spoke configuration for their transportation networks do not suffer from the same level of driver turnover (Taylor et al. 1999). In fact, the turnover rate for LTL carriers does not commonly exceed 10% and the value reported for the last quarter of 2011 is only 7% (American Trucking Associations, 2011).

As freight demand continues to grow and more truck drivers are needed to satisfy this demand, there is need to evaluate alternative dispatching methods as a means to reduce driver tour lengths and improve driver retention. One such alternative dispatching method considers using a network configuration of relay points to allow drivers to exchange trailers and return home more frequently while the truckload continues its transport to its final destination. These TL relay networks essentially divide a load's transportation from origin to destination into several shorter segments between nodes in the transportation network. This helps to increase regularity in the routes for the drivers since now most movements occur between fixed relay points (RPs) (Üster and Maheshwari, 2007; Üster and Kewcharoenwong, 2011). In a general sense, the configuration of a relay network (RN) is somewhat similar to a hub-and-spoke network, except that no sorting or consolidation of truckload freight is required. Also, the strategic design of a RN needs to consider unique operational constraints that make it different from a regular hub-and-spoke network. Figure 15 shows a partial relay network for truckload transportation. A review of the literature on TL relay network design is presented in Section 4.2.

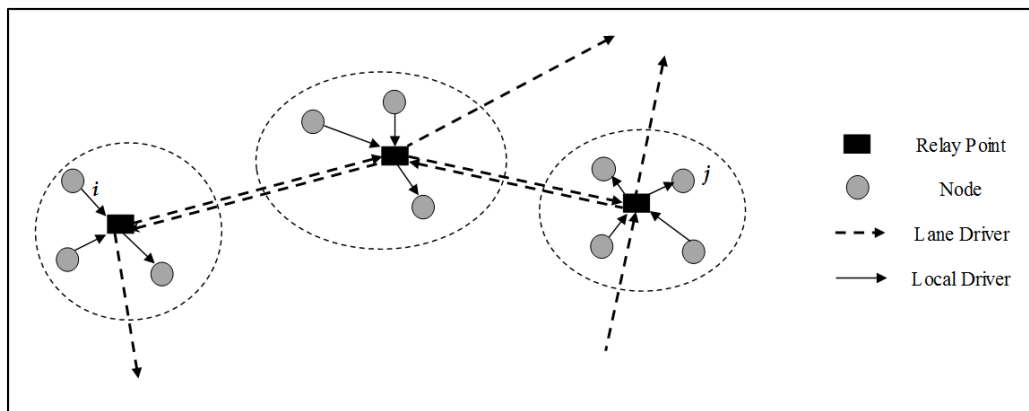


Figure 15. Relay Network for Transportation of Truckloads.

Previous work on the TL relay network design (TLRND) problem has shown that this alternative dispatching method has significant potential to improve drivers' jobs. However, these previous studies also suggest that combining the two methods, PtP and RN, into a hybrid system using a divided fleet will result in a better performance for carriers, drivers and shippers. Interestingly, as described in Section 4.2, most of the literature focuses on the use of simulation and heuristic approaches for solving this and other mixed fleet dispatching system design problems. A general observation from previous work in this area recognizes that hybrid systems seem to consistently outperform the individual methods for freight transportation by truck, but since no exact approaches exist to design such mixed fleet dispatching systems there is not a precise understanding of the actual benefits of such configurations in the TL industry.

In this chapter, we propose a mathematical formulation to solve the integrated problem of strategically designing the relay network and selecting the appropriate dispatching method for truckloads served by a mixed fleet of trucks in a hybrid system that combines PtP and RN dispatching. We call this problem strategic TL relay network design with mixed fleet dispatching or TLRND-MD. The contributions of this work are as follows:

- We develop a prescriptive mathematical model for TLRND-MD using a composite variable model (CVM) formulation. To the best of our knowledge, this is the first prescriptive model that incorporates the selection of the dispatching method for truckloads within the strategic design of a relay network for TL transportation.
- The model presented in this research successfully integrates strategic and tactical decisions while considering several operational constraints using an approach that is similar to the one presented in Chapter 3, but is different from most previous work in this area. This shows that

this modeling formulation is flexible and it is able to incorporate significant changes in problem scope without significantly affecting the tractability of the model.

- The formulation and the heuristic solution approach developed in this research allow us to obtain very high quality solutions for largely-sized problem instances of this problem in reasonable time. An exact solution approach can be used to optimally solve problems of moderate size.
- We use our model to quantify the benefit of a hybrid configuration for TL transportation that uses PtP and RN dispatching over a RN-only dispatching system through extensive computational testing. The effect of several design parameters is analyzed with respect to performance metrics for carriers, drivers and shippers using randomly generated problem instances and a test case provided by a major TL carrier.

The remainder of the chapter is organized as follows. In Section 4.2, we present a literature review of alternative dispatching methods for freight transportation focusing on mixed fleet systems. A formal problem statement of TLRND-MD and the proposed mathematical formulation for its solution are presented in Section 4.3. In Section 4.4, we introduce our solution methodology. The analysis of the computational results obtained for randomly generated problem instances and test case data from a major carrier is presented in Section 4.5. Finally, conclusions and future research are presented in Section 4.6.

4.2. Alternative Dispatching Methods in Freight Transportation

Traditional dispatching methods used in the TL industry have focused on the reduction of empty miles between drop-offs and pick-ups for single drivers in order to reduce dispatching costs. However, this practice and the fact that carriers often receive long-haul driving job

requests in a random fashion make it very difficult to find an appropriate sequence of jobs for drivers so that they can return to their domiciles frequently. It is estimated that TL drivers typically spend between 14 and 21 days on the road between return trips to their home domiciles (Taylor et al. 1999). This has motivated academic and industry researchers to examine the development of alternative dispatching methods that can improve driver job quality and potentially reduce driver turnover rates.

Considering the relatively low driver turnover rate in the LTL industry, Taha and Taylor (1994) analyzed the use of hub-and-spoke networks for TL transportation using a simulation approach. This was the first in a series of studies that considered using a different configuration to dispatch truckloads and observed the benefits associated with such alternative dispatching systems. In general, they concluded that a network configuration for TL dispatching would reduce driver tour lengths due to the regularization of the duties between the hubs. They observed, however, that this reduction comes at the expense of added circuitry for the loads. This tradeoff needs to be explicitly considered when designing the hub-and-spoke networks, especially since truckload circuitry represents an additional cost for the carriers. In a different study, Taylor et al. (1999) considered the development and analysis of other alternative dispatching methods for a regional implementation. Their simulation experiments showed that among the alternatives considered a zone method with hubs located along its perimeter and a hybrid system incorporating a central hub to the zone with perimeter hubs lead to significant improvements in several performance metrics for carriers and drivers. Moreover, they mention that field data indicates that driver tour lengths for the regional jobs analyzed in this research are on the order of 2 to 4 days.

Building on these findings, several other mixed fleet dispatching systems have been analyzed in the literature. Taylor and Whicker (2002) present a heuristic method and an integer programming approach for selecting routings for truckloads and compare various methods of dispatch in a distributed manufacturing setting when considering private fleets held by manufacturers and retailers. Taylor et al. (2006) considered the combination of traditional PtP dispatching and the use of regional fleets with limited coverage service areas. Taylor et al. (2009) analyzed the use of delivery ‘pipelines’ with local end-of-line dray movements for some high volume lanes and assessed the effect of these shipments on the rest of system employing regular PtP dispatching. Another 3-way hybrid system integrating regional fleets, delivery ‘pipelines’ and PtP dispatching in a concurrent operation was explored by Taylor and Whicker (2008). Finally, Taylor and Whicker (2010) studied the use of an extended regional dispatching system using lanes to connect regions of limited service area as they are integrated with a direct PtP dispatching method for some of the truckloads in the system. A general conclusion of these simulation studies is that although operational tradeoffs exist, mixed fleet dispatching systems always outperform a baseline PtP-only system for several metrics that affect carriers, drivers and shippers. Unfortunately, although some general design rules for these hybrid systems are provided, no indication of what constitutes an optimal design is presented in these studies.

Other types of mixed fleet dispatching systems for freight transportation can be found in the LTL literature. Despite the significant differences between the TL and LTL industries, these studies also show that allowing an alternative method in addition to traditional dispatching is beneficial for improving performance metrics, especially obtaining a reduction of dispatching costs. In this area, Ronen (1997) explored the effect of combining a private fleet with the use of

a common carrier when considering two different objectives: distance minimization and cost minimization. A mathematical model is presented to select between the alternative dispatching methods and it is used to demonstrate that the cost minimization approach is superior when dispatching LTL shipments from a single source using a mixed fleet of trucks. In another study, Liu et al. (2003) developed a heuristic approach for the selection of the dispatching mode for LTL loads in a mixed truck delivery system with both hub-and-spoke and direct shipment between suppliers and customers. Their heuristic based on local search also selects routes for individual loads in both modes of delivery. They concluded that the mixed delivery system is more effective than both pure systems based on savings in total distance required for demand satisfaction. As evidenced in these studies, a systematic analysis of alternative dispatching systems that use a mixed fleet of trucks is justified based on the potential benefits that can be obtained.

4.2.1. Relay Networks for TL Transportation

The use of a relay network for TL transportation has been considered as one alternative dispatching method to alleviate the driver retention problem. Most of the research in this area has been motivated by the initial work of Taha and Taylor (1994). Hunt (1998) and Ali et al. (2001) explored the development of algorithmic and heuristic approaches for determining the number and location of relay points considering shortest path routes for the truckloads in the network. The first mathematical formulation for the strategic design of TL relay networks was presented by Üster and Maheshwari (2007) and it was slightly modified by Üster and Kewcharoenwong (2011). Their basic model is a combination of the multicommodity network flow formulation and the hub-and-spoke problem. Üster and Maheshwari (2007) present a

heuristic approach to solve a formulation of TLRND that relaxes limitations on truckload circuitry and equipment balance. As part of their approach, a feasible set of RPs is generated using a construction heuristic that enforces limitations on the length of lane and local driver tour lengths. Then, a tabu search procedure is used to improve the initial solution. This method is successfully applied in small- to medium-sized problem instances of TLRND. Alternatively, Üster and Kewcharoenwong (2011) developed an algorithm based in Benders' decomposition to solve larger problem instances of this problem. In most of their computational experiments, the authors explicitly consider the limitation on equipment balance, but relax the truckload circuitry constraint which is eventually incorporated through a surrogate later in their experimentation. To enhance the efficiency of their typical implementation of Benders' decomposition, the authors analyzed several alternatives within an ε -optimal framework. Their exact approach incorporates strengthened Benders' cuts, cut disaggregation techniques and heuristics for improved bounds. Relay networks with up to 80 nodes are solved using this approach in reasonable times considering an optimality gap of $\varepsilon=2.0\%$.

In Chapter 3, we present an alternative formulation for the strategic design of TL relay networks. We propose a CVM formulation for this problem where operational constraints such as limitations on circuitry, local and lane driver tour lengths and number of RPs visited by a load are implicitly incorporated within the definition of variables that represent feasible routes for the truckloads. This approach requires a very large number of composite variables which are used in an integer programming model to locate RPs and select routes for the truckloads while minimizing total costs. Another advantage of this modeling approach is that perfect equipment balance can be effectively enforced using this method since deadhead movements can be

incorporated as additional variables in the model, whereas this had not been achieved in previous research. Computational experiments with randomly generated data are used to show that this modeling approach provides a stronger formulation for TLRND. Relay networks with up to 100 nodes and high freight density are solved to optimality using standard branch-and-cut as implemented in a commercial solver. Moreover, high quality solutions are obtained in reasonable times for largely-sized instances with 150 nodes using only a subset of the variables. This heuristic approach provided very small optimality gaps while significantly reducing computational times and was successfully used to solve a network design problem with test case data provided by a major TL carrier.

More recently, Melton and Ingalls (2012) proposed another prescriptive model for locating RPs while minimizing total costs. Their mixed integer program incorporates driver considerations and includes a driver turnover cost in the objective function. The driver turnover percentage is determined based on an estimation of the number of drivers required for each origin-destination (O-D) node pair according to the location of RPs and the resulting number of miles per week per driver. However, this formulation does not explicitly consider empty movements to balance equipment and it is only used to solve a small case study for a single O-D pair.

Although these studies show that relay networks for TL transportation have the potential to improve the quality of TL driving jobs, some of the results presented also seem to suggest that a partial implementation of this alternative configuration along with direct PtP dispatching is likely to result in a system with better performance. Both Üster and Kewcharoenwong (2011) and Chapter 3 point out that for a non-trivial number of truckloads with origin and destination

nodes that are close to each other, highly circuitous routes result from the use of a relay network for dispatching. Figure 16 shows an example of this case when a truckload has origin at node i and it is destined to node j . It is easy to observe that routing this truckload through the relay network is not as efficient as sending it directly from i to j . In Chapter 3, we also found out that several RPs are only opened for feasibility reasons, especially when truckloads have origin or destination nodes in isolated regions of the transportation network. We observed that the amount of traffic at these RPs is significantly lower than in other parts of the network with higher node density. These isolated truckloads are also good candidates for direct PtP dispatching. Having the option of sending them using this method would result in a reduction in the number of RPs in the network.

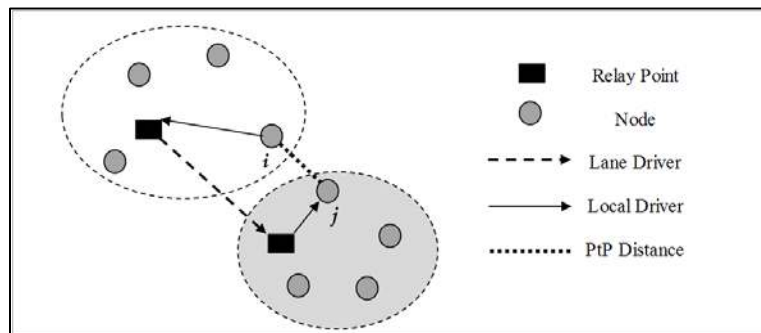


Figure 16. Highly Circuitous Route in a Relay Network for Truckload Transportation.

The design of relay networks for TL transportation using this mixed fleet dispatching system was analyzed by Üster and Kewcharoenwong (2011) using two approaches. The first is an a priori approach in which first some truckloads of reduced distance (i.e., less than 2/3 of the limitation for local movements) are pre-selected for PtP dispatching and then the relay network is designed for the remaining truckloads. The second approach considers designing the relay network for all truckloads and then selecting truckloads with excessive circuitry (e.g. more than

200% circuitry) to be dispatched using PtP dispatching. The authors determined that the a priori approach as implemented in their research provided a greater reduction in total costs with more truckloads being dispatched PtP. However, no sensitivity analysis of the values used for selecting truckloads for direct dispatching in both methods is presented in this research. To the best of our knowledge, no prescriptive model exists in the literature to solve this problem in an integrated manner. The research presented in this chapter is intended to fill this existing need.

4.3. Problem Statement and Mathematical Formulation

4.3.1. *Truckload Relay Network Design with Mixed Fleet Dispatching (TLRND-MD)*

Given a set of truckloads that need to be transported, the TLRND-MD problem minimizes the total cost of opening RPs and routing truckloads either through the relay network or directly PtP from origin to destination. Figure 15 shows an example of a RN for TL transportation and Chapter 3 provides a detailed description of its structure and basic operation. Two types of drivers are required to handle the truckloads that are routed through the RN. *Local drivers* handle the movements from an origin node to a RP or from a RP to a destination node. The distances covered by this type of drivers are limited so that they can return home each night using a pre-specified limit, γ_1 . *Lane drivers* are in charge of the movements between RPs. In this case, a limitation on distances for lane drivers, γ_2 , is imposed to ensure that federal hours-of-service regulations are satisfied in a single movement between two RPs (U.S. Department of Transportation, 2011). Alternatively, distances for drivers who are responsible of PtP loads are not limited in accordance to current practice in the industry.

Additional constraints are imposed based on requirements from TL carriers. In order to prevent excessive handling of the RN loads and be able to ensure timely service to customers, a

limitation is imposed on the number of RPs that can be visited between an O-D pair, λ . We also consider a limitation on circuitry for the truckloads, β , so that the number of additional miles driven in the RN does not represent a considerable expense for the carriers. In this case, if the shortest path distance between origin and destination for a truckload is d , only those routes that are shorter or equal to $(1 + \beta)d$ are allowed as alternatives to move that truckload. Similarly, in order to reduce the number of empty miles driven and ensure that no node in the RN suffers an excessive deficit or surplus of empty trailers, equipment balance must be enforced at the RPs.

The formulation of TLRND-MD is further complicated by two design considerations that are imposed by TL carriers. First, a minimum volume of freight traffic is desired at the RPs in order to justify their installation. Nodes that do not exceed this threshold are not considered as RPs. Second, the proportion of loads that are dispatched direct PtP with respect to the total demand in the system cannot exceed a specified percentage value. This maximum value is set to ensure that the benefits of improved driver retention are actually attained by having most of the driving jobs in regular routes.

4.3.2. *Composite Variable Formulation for TLRND-MD*

In this chapter, we present a CVM formulation for TLRND-MD that extends the original formulation presented in Chapter 3 for the TLRND problem. As in this previous work, our *composite* variables are defined as feasible routes for a truckload when it is dispatched using a RN. Limitations on circuitry, number of RPs visited, and local and lane distances are enforced when creating feasible routes according to the method described in Section 4.4.1. This allows us to incorporate the difficult operational constraints for the design of the RN within the variable definition and decompose this problem into several smaller routing problems. Additionally, we

introduce a set of decision variables that let the truckloads to travel PtP. Using this approach, the selection of dispatching mode for the truckloads and the coordination of the routing decisions for RN loads are handled by the following integer program.

4.3.2.1. Notation

Based on the definition of our composite variables, the following notation is required for the formulation of the CVM for TLRND-MD.

Sets

R = set of composites r ,

T = set of truckloads t ,

N = set of nodes k ,

R_t = set of composites r for truckload t , $R_t \subset R$,

R_k = set of composites r that visit node k , $R_k \subset R$.

Parameters

c_r = cost of composite r , $\forall r \in R$,

F_k = fixed cost of relay point k , $\forall k \in N$,

P_t = cost of dispatching truckload t using PtP dispatching, $\forall t \in T$,

b_t = demand for truckload t (in number of loads), $\forall t \in T$,

δ = maximum acceptable percentage equipment imbalance,

ρ = maximum proportion of truckloads to be dispatched direct PtP,

v = minimum volume (in number of loads) required to open a RP,

$\eta_{kr} = -1$ if node k is the origin relay point of composite r ,

1 if node k is the destination relay point of composite r , $\forall k \in N, r \in R$,

0 otherwise,

$\theta_{kr} = 1$ if composite r visits relay point k , $\forall k \in N, r \in R$,

0 otherwise.

Variables

x_r = number of composites r used, $\forall r \in R$,

$y_k = 1$ if a relay point is opened at node k , $\forall k \in N$,

0 otherwise.

z_t = number of truckloads t sent direct PtP, $\forall t \in T$.

4.3.2.2. Model Formulation

The mathematical formulation for TLRND-MD is as follows.

$$\min \sum_{r \in R} c_r x_r + \sum_{t \in T} P_t z_t + \sum_{k \in N} F_k y_k \quad (9)$$

subject to

$$\sum_{r \in R_t} x_r + z_t = b_t \quad \forall t \in T \quad (10)$$

$$\sum_{r \in R_t} \theta_{kr} x_r \leq b_t y_k \quad \forall t \in T, k \in N, r \in R_t \quad (11)$$

$$\sum_{r: \eta_{kr} = -1} x_r - \sum_{r: \eta_{kr} = 1} x_r \leq \delta \sum_{r: \eta_{kr} = -1} x_r \quad \forall k \in N \quad (12)$$

$$\sum_{r: \eta_{kr} = 1} x_r - \sum_{r: \eta_{kr} = -1} x_r \leq \delta \sum_{r: \eta_{kr} = 1} x_r \quad \forall k \in N \quad (13)$$

$$\sum_{r \in R} \theta_{kr} x_r \geq \nu y_k \quad \forall k \in N \quad (14)$$

$$\sum_{t \in T} z_t \leq \rho \sum_{t \in T} b_t \quad (15)$$

$$x_r \text{ integer} \quad \forall r \in R \quad (16)$$

$$y_k \in \{0,1\} \quad \forall k \in N \quad (17)$$

$$z_t \text{ integer} \quad \forall t \in T \quad (18)$$

The objective function (9) minimizes the total cost of routing truckloads through the RN, the cost of dispatching loads PtP, and the fixed cost of installation of the RPs. Constraint (10) requires that a routing for each truckload is selected either using a RN route (i.e. a composite) and/or dispatching the load directly from origin to destination. In either case, total demand for all truckloads has to be satisfied. Constraint (11) enforces that selected routes can visit relay points in the network only if they are open. Constraints (12) and (13) enforce relay network balance by requiring that a maximum permissible imbalance at each node be satisfied considering the difference between outgoing and incoming flows for RN loads. Note that a value of $\delta = 0$ enforces perfect balance for RN loads at every node in the transportation network; this is the only case in which constraints (12) and (13) will be binding simultaneously. Note that since PtP loads are dispatched using a different fleet, they are not included in these constraints. To account for the balance of PtP loads, we include a repositioning cost into the estimation of the cost of sending a load direct from origin to destination, P_i . In particular, we recognize that additional miles must be traveled to achieve balance in the PtP network. This allows the model to accurately tradeoff the increased travel required to reposition PtP movements and the cost of opening and rebalancing trailers in the RN. Constraint (14) requires the volume at each open RP to meet or exceed a pre-specified threshold, ν . Constraint (15) requires that the proportion of

truckloads dispatched direct PtP with respect to the total demand does not exceed a given limitation set by the TL carrier. Finally, constraints (16), (17) and (18) enforce integrality for all decision variables.

4.4. Solution Methodology

4.4.1. *Generation of Composite Variables*

An enumeration based procedure using *templates* is used to generate composite variables for our formulation of TLRND-MD presented in Section 4.3.2.2. A template is a predefined routing pattern that represents an alternative for moving truckloads from origin to destination through a series of intermediate nodes. Since the number of RPs allowed to be visited in a route is limited to $\lambda=3$ based on TL carrier requirements, we are able to completely enumerate all feasible routes with three or fewer stops that satisfy constraints on circuitry and distances for local and lane drivers. Even with this limitation on the number of RPs that can be visited, the number of composite variables required for our formulation is very large. Figure 17 shows all template types that were considered in this research. Note that no templates with a total of two nodes from which only one is a RP are considered here. The reason behind this is that since these alternatives connect O-D pairs directly over a limited distance, truckloads that would be generated with these templates can be handled by a different fleet than the truckloads that are routed though the RN.

Note that since we enforce the important operational constraints associated with circuitry and distance limitations by checking the feasibility of the routes that are generated with this approach, there is no need to incorporate them as constraints in the mathematical formulation for TLRND-MD.

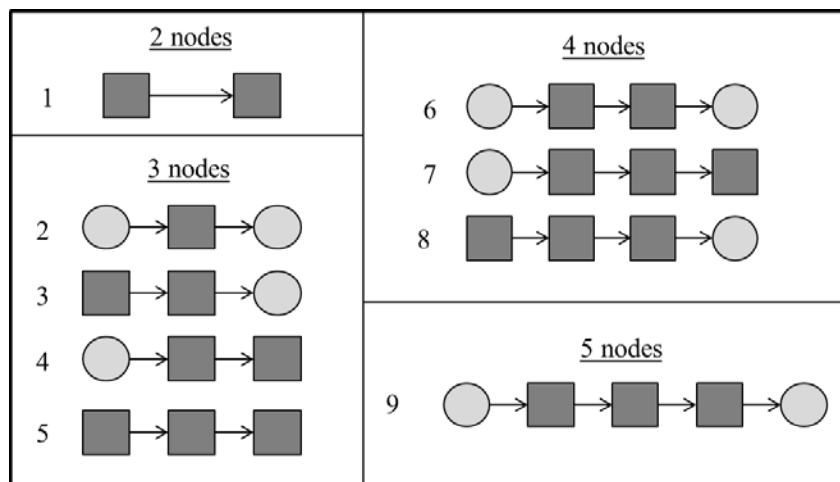


Figure 17. Template Types Used for Composite Variable Generation.

4.4.2. Heuristic Solution Method

Exact solutions for some instances of TLRND-MD can be obtained using standard branch-and-cut as implemented in CPLEX. Preliminary testing with the formulation discovered that the number of composite variables needed significantly affects the tractability of the model. However, this testing also revealed that optimal solutions for this problem do not usually consider composites generated for some of the template types that are presented in Figure 17. This presented an opportunity to reduce the number of variables significantly. Table 15 shows the usage of each template type in the optimal solutions found during our preliminary testing with 50 node networks.

Moreover, an analysis of the truckloads that are dispatched PtP showed that most of these loads traveled short distances. Figure 18 shows how many loads travel PtP and via the RN as a function of the distance between origin and destination nodes.

Table 15. Usage of Composite Variables in Optimal Solutions by Template Type.

| Template Type | # of Composites Used | Percentage of Total |
|---------------|----------------------|---------------------|
| 1 | 481 | 2.72% |
| 2 | 3,339 | 18.88% |
| 3 | 1,639 | 9.27% |
| 4 | 1,594 | 9.01% |
| 5 | 6 | 0.03% |
| 6 | 8,418 | 47.59% |
| 7 | 6 | 0.03% |
| 8 | 6 | 0.03% |
| 9 | 6 | 0.03% |
| Total RN | 15,495 | 87.59% |
| Total PtP | 2,195 | 12.41% |
| Total | 17,690 | 100% |

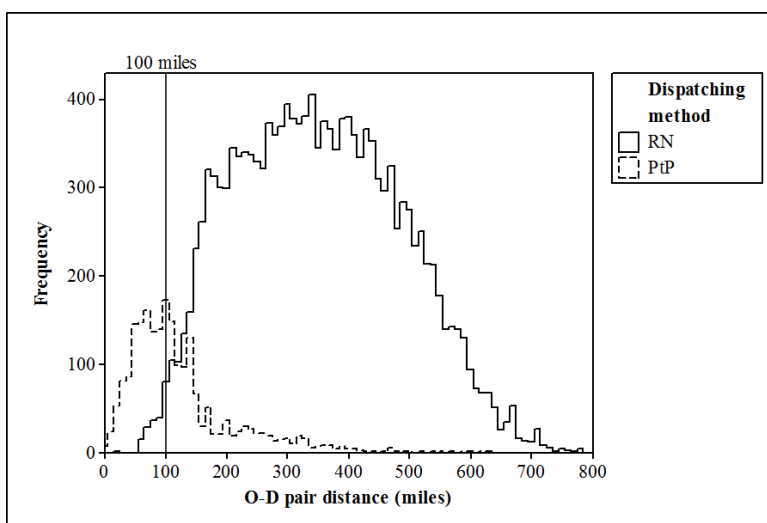


Figure 18. Dispatching Method Usage with Respect to Truckload O-D Pair Distance.

These observations from empirical results allowed us to develop a heuristic approach to obtain high quality solutions to TLRND-MD. The basic idea behind this heuristic approach is that the generation of composite variables is modified so that only a reduced subset of high quality variables is created and added to the model. This reduced model is then solved using CPLEX until a pre-specified optimality tolerance is satisfied. We use a similar approach to the one implemented in Chapter 3 (i.e., elimination of template types that produce variables that are

not commonly used in optimal solutions), but we enhance it by further reducing the number of composites in the model when we avoid creating routes for truckloads that have O-D pair distances that are below a given threshold. The assumption is that these truckloads will be dispatched using the direct PtP method and no alternative routes (i.e., composites) are required for them. From Table 15, it is evident that template types 5, 7, 8 and 9 are rarely used. Based on these findings, the heuristic implemented in this research considers only the remaining templates with two or less RPs that are shown in Figure 17 (i.e. types 1, 2, 3, 4 and 6).

In the same way, as observed in Figure 18, truckloads with O-D pair distances below 100 miles seem to be dispatched primarily using PtP rather than routing them through the relay network; we therefore preprocess the solutions to ensure such loads will travel PtP and avoid generating composite variables for them. Although we are not able to guarantee that optimal solutions will be obtained with this heuristic approach, our computational results showed that we are still able to obtain very high quality solutions while significantly improving the performance of the model. The average and worst case optimality gaps observed for 50 node networks are 0.1% and 0.3% respectively, while there is a significant reduction on the number of composites generated for the model that exceeds 86%. The reduction in problem size considerably improves the performance of the model by reducing total CPU times almost 90%. These improvements allow us to solve largely-sized problem instances in reasonable times. Further results are presented in Section 4.5.3.2.

4.5. Computational Results

Computational testing of our proposed formulation and solution approach for TLRND-MD considers the use of a baseline scenario and seven alternative cases based on modifications

to the original design parameters established. One of the purposes of our computational experiments is to perform sensitivity analysis to observe the effect of parameter changes on the design of the RNs (i.e., number of RPs opened and dispatching method selected for the truckloads) and performance metrics for drivers, carriers and shippers. Different network sizes and freight densities are considered through randomly generated problem instances. Additionally, we compare the results obtained for the mixed fleet dispatching system with those of a pure RN implementation and a PtP-only dispatching system. These results are presented in Sections 4.5.3.1 to 4.5.3.3. Finally, to assess the performance of our proposed formulation in a realistic scenario, we include the analysis of a test case provided by a major TL carrier in Section 4.5.3.4.

4.5.1. Experimental Design

In order to determine the influence of certain design parameters on the solutions obtained for TLRND-MD and the performance of our model, a baseline and several alternative scenarios are defined by varying parameter values. The results obtained for each scenario are averaged over 10 replications of 50, 100 and 150 node test networks that were randomly generated using the approach that is presented later in Section 4.5.2.

The baseline scenario was established to assess the effect of allowing truckloads to be dispatched either PtP or through the RN without enforcing a limitation on the maximum proportion of PtP loads allowed ($\rho=1.0$) or requiring a minimum volume of truckload traffic to open a RP ($v=0$). Additionally, as the results for the mixed fleet dispatching system are to be compared to a RN-only system, perfect balance is required for RN loads ($\delta=0$) and repositioning costs of PtP loads are estimated to be one quarter of the actual cost of transportation between

origin and destination nodes. In this way, transportation costs are inflated 25% for these loads. This 25% increase over the regular rate per mile for lane movements comes from industry estimates of the average percentage of empty miles for TL carriers (Alam et al., 2007). Finally, the fixed cost of installation of a relay point is assumed to be \$10,000. Discussions with a major TL carrier indicate that installing RPs requires very little capital investment. This is because exchanging trailers between drivers at these locations does not require the use of expensive equipment and infrastructure. Besides land cost, RP facilities only require a perimeter fence with a security system in place to prevent the loss of assets. Exact and heuristic results obtained for this scenario are presented in Sections 4.5.3.1 and 4.5.3.2 respectively.

In each of the following scenarios, the value of one of the parameters is modified with respect to the baseline scenario to evaluate the effect of the corresponding parameter on the solutions obtained while the remaining parameters were held constant across all scenarios during our computational experimentation. This sensitivity analysis is presented in Section 4.5.3.3. We explored the effect of repositioning cost for PtP loads by considering values that are lower and higher than the estimated value used in the baseline scenario. We also considered the effects of establishing a limitation on the proportion of loads that are dispatched PtP and the requirement of having a given amount of truckload traffic visit a node in order to open a RP at that location respectively. We also analyzed the effect of allowing some imbalance at the nodes for truckloads that are dispatched through the RPs to assess the effect of enforcing equipment balance. And finally, we studied the effect of a higher fixed cost of installation for the RPs.

For system parameters, we used values similar to those used in Chapter 3 to solve TLRND. A limitation of 25% circuitry (β) above the shortest path distance between origin and

destination nodes was imposed when generating feasible RN routes for a truckload. Similarly, the distances covered by local (γ_1) and lane (γ_2) drivers were limited to 150 miles and 600 miles respectively. Note that changes to these parameters affect the number of composite variables that are generated and consequently have an effect on the tractability of our model. Finally, the rates per mile charged for local and lane movements were set to \$1.0 and \$1.30 respectively.

4.5.2. *Random Network Problem Generation*

Similar to the approach used in Chapter 3, our computational tests used random instances of *complete networks* with 50, 100 and 150 nodes to assess the effect of network size on model performance. These are networks that have arcs connecting every pair of nodes in both directions, and consequently have the highest network density possible. Although practical transportation networks are sparser, these instances allow us to test the worst case performance of our formulation since the number of composites (i.e., feasible routes for the truckloads through the RN) that will be generated using the approach described in Section 4.4.1 is expected to be very large. Nodes are uniformly distributed in a squared area that represents the regional service area of a major TL carrier (i.e., 600 miles \times 600 miles), and distances between node pairs are computed using the Euclidean norm as a surrogate to actual over the road miles. Ten network problems of each size were generated to reduce random effects.

To test the performance of our model with different truckload flow densities, the number of lanes in the network (i.e., O-D node pairs) with freight traffic was also modified. To accomplish this, 10%, 20% and 40% of all O-D pairs were randomly selected to have truckload flows. This means that for a network with 100 nodes, we selected 990, 1,980 and 3,960 O-D node pairs to have truckload flows. Moreover, to represent lanes with different freight volume, a

randomly generated integer number between 10 and 20 was used to establish the actual number of truckloads required between each selected O-D node pair. The latter is similar to the approach used by Üster and Kewcharoenwong (2011).

4.5.3. Results

We implemented our formulation and solution approach using Python 2.6 and solved the computational experiments with CPLEX 12.1 on a 3.20 GHz Intel® Xeon® workstation with 6 GB of RAM.

4.5.3.1. Exact Solution Results

Table 16 shows the results obtained for the baseline scenario with 50 node networks using CPLEX 12.1 (i.e., a standard branch-and-cut method) to solve our CVM formulation for TLRND-MD. As observed in this table, freight density affects the size of the problem by increasing the number of variables and constraints in the formulation. In turn, the increase in problem size directly affects the time required for the solution. Setup times correspond to the time required to generate composites and build the mathematical model that is then solved by CPLEX using the time shown in the last column of this table (i.e., Solution Time). Although problem size negatively affects total CPU time, optimal solutions are obtained for high freight density problems in very reasonable times (i.e., less than 5 minutes in the worst case). The values presented in Table 16 are averages for ten replications of problems with the same freight density.

From this table, note that as freight density increases the number of RPs that are opened increases while the proportion of loads that are dispatched PtP decreases. This is the result of having more traffic in the system and consequently better opportunities for relayed movements.

More importantly, the proportion of loads that are dispatched PtP never exceeds 20% and is almost half of that number for higher freight density problems. This means that most of the demand will be satisfied by RN drivers, and thus driver retention will generally improve for the TL carrier.

Table 16. Exact Results for 50 Node Networks (Baseline Scenario).

| # O-D Pairs | # Composites | # Constraints | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------------|--------------|---------------|---------------------|------------|-------------------------|-------------------|----------------------|
| 245 | 30,743.5 | 3,471.4 | 1,650,991 | 15.1 | 19.58% | 45.83 | 6.07 |
| 490 | 64,077.7 | 6,894.0 | 3,148,072 | 19.9 | 13.65% | 94.22 | 6.42 |
| 980 | 127,566.0 | 13,660.1 | 6,040,305 | 25.6 | 10.43% | 197.27 | 8.87 |

One of the objectives of the present study was to determine the benefit of a mixed fleet dispatching system over RN-only and PtP-only systems. Table 17 shows the comparison of these two other dispatching methods with the mixed fleet system. The values between parentheses are the differences observed with the mixed fleet results.

Table 17. Comparison RN-only and PtP-only Systems with Mixed Fleet Dispatching (50 Nodes).

| # O-D Pairs | Mixed Fleet | | RN-only | | PtP-only | |
|-------------|---------------------|------------|-----------------------|----------------|------------------------|------------|
| | Solution Value (\$) | # RPs Open | Solution Value (\$) | # RPs Open | Solution Value (\$) | # RPs Open |
| 245 | 1,650,991 | 15.1 | 1,694,802 (+2.65%) | 24.0 (+8.9) | 1,913,915 (+15.82%) | N/A |
| 490 | 3,148,072 | 19.9 | 3,164,872 (+0.53%) | 28.7 (+8.8) | 3,884,885 (+23.40%) | N/A |
| 980 | 6,040,305 | 25.6 | 6,018,572 (-0.36%) | 32.5 (+6.9) | 7,734,080 (+28.06%) | N/A |

Results for the pure RN implementation were obtained using the heuristic approach presented in Chapter 3 to solve TLRND, while the solution values for the PtP system were computed considering a repositioning cost of 25% over the lane rate per mile for all truckloads when they are dispatched directly from origin to destination. Note that the mixed fleet system provides significant savings as compared to the PtP-only system, and those savings increase as

freight density increases. This is due largely to the need to reposition the equipment for the next pick-up of a PtP load which adds empty miles to the transportation costs in this system. Repositioning is significantly easier and less expensive in a relay network configuration since loads flow between a limited number of points and can be repositioned more easily. Also, given the savings observed note that the solutions for the mixed fleet dispatching system are still cheaper than using the PtP-only method even when the fixed cost of installation of RPs increases from its baseline of \$10,000 to up to \$17,000, \$37,000 and \$66,000 respectively for problems with 10%, 20% and 40% freight density.

In contrast, although there is no significant difference in the total cost as compared to the RN-only method, using the mixed fleet system results in relay networks with considerably fewer RPs. This reduction in the number of RPs represents other operational advantages to the carriers in terms of managing fewer driver domiciles and increased utilization of each of the RPs that is installed. Note, however, that another byproduct of the mixed fleet system is increased circuitry for the truckloads that travel through the relay network since fewer RPs are open.

Although the mixed fleet dispatching system seems to provide overall benefits to the carriers, it is also important to look at performance metrics for drivers and customers. A measure of importance to drivers is the *average length of haul* defined as the average one-way driving distance covered in a single leg between two nodes. A reduced average length of haul allows the creation of shorter driving duties (i.e., driving jobs that initiate and terminate at a driver's domicile). This means more frequent returns to domiciles for the drivers and consequently, an improvement in quality of life that leads to lower driver turnover. Similarly, a metric that interests shippers is *service time* defined as the estimated time to deliver a truckload from origin

to destination. In a PtP system, service time is estimated by considering the distance between origin and destination for the truckload, average speed for the vehicles, and hours-of-service regulations for the drivers. Given the long distances that are usually covered by TL carriers, some loads are commonly delayed while drivers stop to rest in compliance with federal safety regulations. On the other hand, truckloads that are dispatched on the RN can be immediately transferred to a different driver assuming there are available drivers at the RP and continue to their final destination. It is important to determine expected and worst case service times for RN loads and PtP loads in the mixed fleet system in order to guarantee appropriate service to all customers.

Figure 19 shows the results for average length of haul and Figure 20 presents the average and worst case service times observed for the three alternative dispatching systems under study. Once again, the PtP-only dispatching system shows the worst performance of the three with the highest average length of haul and worst case service times observed. The comparison between the RN-only and mixed fleet dispatching systems shows that there is no significant difference between the metrics obtained for truckloads that are shipped through the relay network. However, an advantage of the mixed fleet system can be observed for those truckloads that are sent PtP. The reduced average length of haul for these loads allows the creation of short duties for the drivers who are responsible for these movements. As a result, even if they are not assigned to relayed movements, it is very likely that they will be able to return home more frequently. Also, average service times for PtP loads in the mixed fleet dispatching system are below the average service times observed for RN loads since many of the truckloads dispatched PtP travel only short distances between origin and destination nodes.

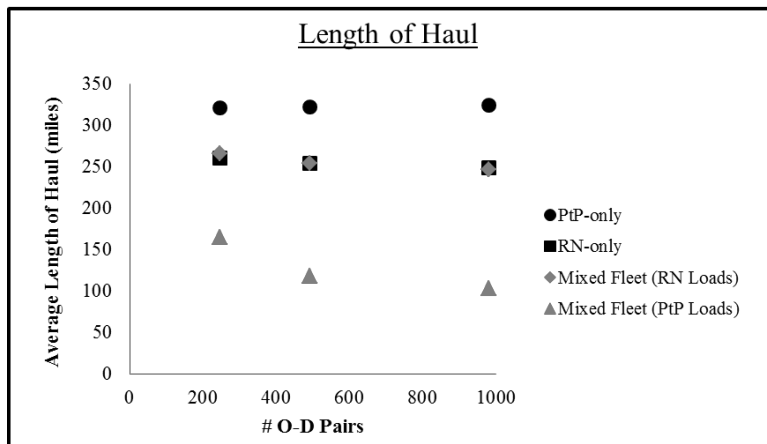


Figure 19. Length of Haul Results for Alternative Dispatching Systems (50 Nodes).

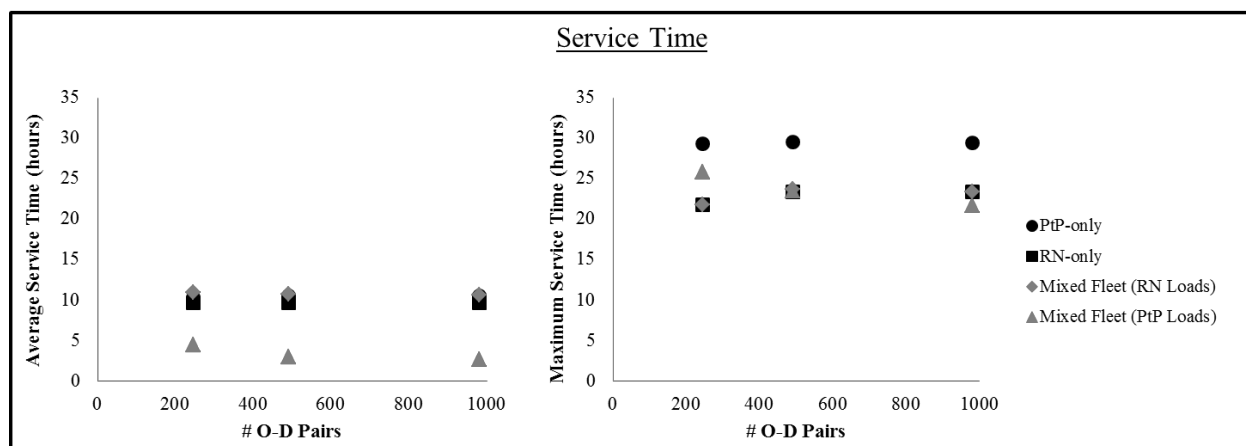


Figure 20. Service Time Results for Alternative Dispatching Systems (50 Nodes).

Finally, it is important to note that the strength of our formulation was analyzed by solving these problem instances using the LP relaxation of our model. The average and worst case optimality gaps obtained were 0.03% and 0.22% respectively. In fact, our formulation proved to be better as freight density increased. This is an indication that the CVM formulation has very good lower bounds even when problem size increases. This allows us to obtain high-quality solutions quickly.

4.5.3.2. Heuristic Solution Results

Since the number of variables ultimately affects our model's tractability, we decided to implement the heuristic method presented in Section 4.4.2 that omits several template types and consequently reduces the number of composite variables in our model to solve the 50 node network instances and test the performance of the heuristic approach in terms of solution quality and efficiency. Table 18 shows the results obtained using the heuristic. The values between parentheses correspond to the differences observed with respect to the exact solutions presented in Table 16. As shown in this table, the significant reduction in the number of composite variables and constraints in the model allows us to obtain very high quality solutions efficiently. The reduction in setup times is a direct result of the reduction in problem size.

Table 18. Heuristic Results for 50 Node Networks (Baseline Scenario).

| # O-D Pairs | # Composites | # Constraints | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------------|-----------------------|----------------------|-----------------------|----------------|-------------------------|--------------------|----------------------|
| 245 | 4,096.2 (-86.68%) | 2,381.7 (-31.39%) | 1,652,401 (+0.09%) | 15.2 (+0.1) | 20.21% (+0.63) | 5.06 (-88.96%) | 2.17 (-64.33%) |
| 490 | 8,351.9 (-86.97%) | 4,628.9 (-32.86%) | 3,151,178 (+0.10%) | 19.7 (-0.2) | 14.50% (+0.85) | 9.58 (-89.83%) | 1.73 (-73.10%) |
| 980 | 16,828.7 (-86.81%) | 9,182.2 (-32.78%) | 6,047,650 (+0.12%) | 25.3 (-0.3) | 11.52% (+1.09) | 18.66 (-90.54%) | 2.17 (-75.49%) |

Based on the performance of the heuristic, we used this method to solve larger problem instances with more nodes and freight traffic to assess the effect of network size and freight density. The results for 100 and 150 node networks are presented in Table 19.

These results show that while the number of composites in the model directly affects setup times, solution times are mostly determined by the size of the networks (i.e., number of nodes). However, setup time is still the most important fraction of total CPU time in most cases. The largest problem instances with 150 nodes and 8,940 O-D pairs with truckload flows are built and solved in less than 55 minutes in the worst case. This is quite acceptable considering that

this is a strategic design problem. It is interesting to note that solution times seem to be higher for problems with the same number of nodes but lower freight density. This may be an indication that as more truckload traffic exists more common routes can be defined and it is easier to design the relay network.

Table 19. Results for 100 and 150 Node Networks (Baseline Scenario).

| Nodes | # O-D Pairs | # Composites | # Constraints | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|-------------|--------------|---------------|---------------------|------------|-------------------------|-------------------|----------------------|
| 100 | 990 | 55,684.7 | 16,170.5 | 5,902,860 | 29.4 | 10.09% | 70.12 | 118.68 |
| | 1,980 | 109,663.5 | 31,688.8 | 11,152,766 | 39.3 | 9.24% | 143.56 | 51.08 |
| | 3,960 | 219,354.3 | 63,079.8 | 21,970,931 | 50.6 | 8.57% | 312.50 | 46.71 |
| 150 | 2,235 | 265,290.2 | 51,505.2 | 12,310,993 | 55.6 | 8.30% | 408.82 | 266.40 |
| | 4,470 | 532,480.1 | 102,724.4 | 23,906,718 | 68.7 | 8.02% | 992.13 | 165.51 |
| | 8,940 | 1,065,642.4 | 204,741.7 | 47,230,483 | 76.9 | 8.00% | 2,722.20 | 253.09 |

Another observation from these results is that the fixed cost of installation of the RPs is not a significant fraction of the total cost. This is clearly the case for problems with 150 nodes and 2,235 O-D node pairs. This indicates that transportation costs account for most of the expense for the carriers. Also, similar to the results obtained for 50 node networks, the number of open RPs increases and the proportion of loads dispatched PtP decreases as freight demand increases for networks of the same size. However, the rate of decrease for the proportion of loads dispatched PtP seems to be decreasing as well with problem size.

Comparing these results to those obtained for RN-only and PtP-only dispatching systems we observed similar results to those obtained for 50 node networks. Both RN-only and mixed fleet systems outperform traditional PtP dispatching, and although the objective function values are very close for the network-based systems, we noticed a greater reduction in the number of RPs required in the mixed fleet dispatching system as the size of the network increases requiring

around 13 fewer RPs in 100 node networks and up to 30 fewer RPs in 150 node networks. Certainly, this represents an advantage for the mixed fleet dispatching system, particularly as the cost of relay point installation increases as discussed later in Section 4.5.3.3.5.

4.5.3.3. *Sensitivity Analysis for Different Carrier Parameters*

The results analyzed so far correspond to the baseline scenario presented in Section 4.5.1. Appendix A shows the results in detail for the scenarios that explore the effects of repositioning costs for PtP loads, the proportion of PtP loads permitted, the minimum volume required to open a relay point, the percentage of imbalance allowed in the relay network, and the fixed cost of installation of RPs. These results are analyzed in the following subsections.

4.5.3.3.1. Effect of Repositioning Cost for PtP Loads

In two alternative scenarios, we considered modifications to the repositioning cost applied to loads that are dispatched PtP to evaluate the effect of this parameter on the solutions obtained and the performance of our formulation. The scenario called *Low PtP Repositioning Cost* tests a repositioning cost that is less than the average percentage of empty miles (10%) for TL carriers while the scenario called *High PtP Repositioning Cost* considers a cost that doubles the baseline cost (50%). A repositioning cost of zero was not used since additional moves to balance equipment for the PtP fleet will always be needed in practice. Table A.1 and Table A.2 in Appendix A show the results for these two scenarios.

In the low PtP cost case, we observed a reduction in solution values and number of RPs that are required in the relay network. However, the reduction in total cost is not significant and never exceeds 2%. A greater effect is observed in the proportion of truckloads that are shipped PtP, especially for 50 node networks and low freight density. This is an indication that with the

lower repositioning cost, some truckloads are better served PtP since circuitry and rebalancing miles can be avoided in the RN. The increase in the proportion of loads moved PtP is less significant for bigger instances since having more truckloads in the system requires routing a higher proportion of truckloads through the RN to obtain balance at the nodes. In terms of the performance of our model, there is a prevalent reduction in solution times which seems to show that it is generally easier to select the dispatching method when the transportation cost for PtP loads is less expensive and closer to the cost for RN loads.

Looking at the results for the high PtP cost case, we noticed a similar behavior but in the opposite direction. Increments in solution values and number of open RPs are observed but they are only marginal (the highest increase in solution value does not exceed 2%). The effect of the higher repositioning costs on solution times is less clear and seems to range from higher CPU times for largely-sized problems with 150 nodes to lower CPU times for smaller instances. These results indicate that even assessing a very high repositioning cost seems to result in a mixed fleet dispatching system that has advantages over a pure RN dispatching method in terms of fewer RPs open and comparable total costs.

Figure 21 shows the effect of varying the repositioning cost for PtP loads from 10% to 50% on solution values and number of open RPs for instances with 100 nodes and 10% freight density. As observed in the graph located on the left side of Figure 21, the savings associated with the mixed fleet system as compared to a RN-only system decrease as the repositioning cost for PtP loads increases. Note that when the repositioning cost is 50% over the cost of transportation, the cost of a mixed fleet system exceeds that of a RN. The main reason for this is that in our approach we pre-process truckloads with distances shorter than 100 miles to be

dispatched PtP in the mixed fleet dispatching system, and thus the transportation cost for these truckloads always exceeds the cost that would be incurred if they are dispatched through the relay network as local movements. On the other hand, it can also be observed in the graph located on the right side of Figure 21 that the hybrid configuration consistently requires fewer RPs in the network than a RN-only system.

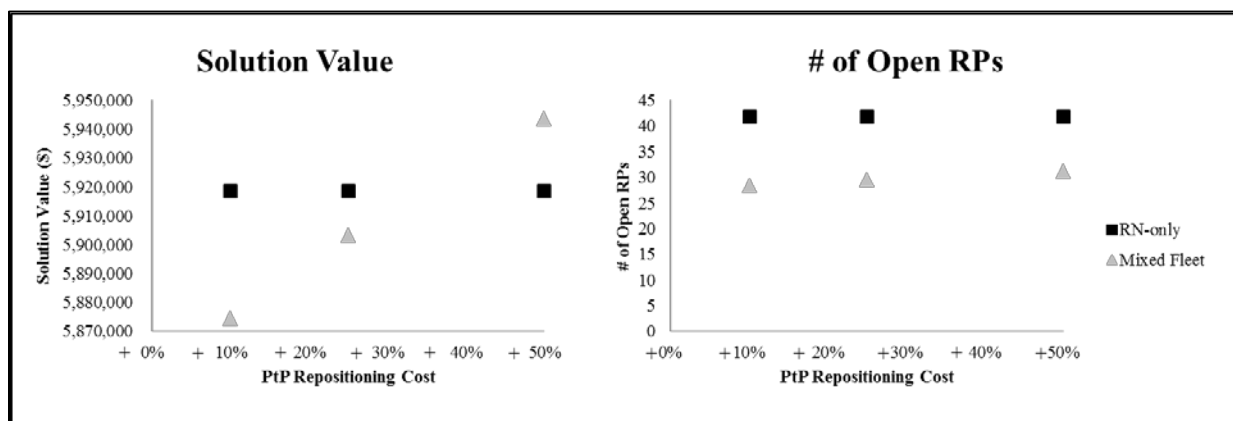


Figure 21. Differences Between RN-only and Mixed Fleet as PtP Repositioning Cost Varies.

4.5.3.3.2. Effect of Maximum Proportion of PtP Loads

One of the parameters considered for the design of a mixed fleet dispatching system is the maximum proportion of loads that are allowed to be dispatched PtP. TL carriers believe that limiting the proportion of loads traveling PtP represents the best compromise between overall driver retention (given that most loads are dispatched through the relay network and shorter tour lengths are possible) and reduced circuitry (i.e., additional miles driven) for the truckloads. In the baseline scenario we did not impose a restriction on this value. For the scenario called *Limited Proportion of PtP Loads*, we used a value of $\rho = 0.1$ to limit the proportion of loads that are sent PtP to 10%. The results obtained are presented in Table A.3 in Appendix A.

The limitation on the proportion of PtP loads in the mixed fleet system proved to be more restrictive for the smaller networks with 50 nodes, especially when freight density is low. In

these cases, the limitation on the proportion of PtP loads results in the need for more RPs to handle more traffic through the RN. However, since the fixed cost of installation of RPs is not a significant portion of the total cost, solution values do not change significantly. The observed worst case increase in solution value just barely exceeds 1%. It is also important to note that there are some infeasible instances for which no solution can be obtained using our heuristic approach. The reason for this is that since we pre-select truckloads with O-D node pair distance below 100 miles, the proportion of PtP loads that will exist in the system is already higher than the limitation imposed by the model. The exact solution approach or a heuristic method without pre-processing of PtP loads will be able to find solutions for these problems.

In terms of the performance of our formulation when this restriction is imposed, we observed that with a few exceptions, solution times are significantly higher than those observed in the baseline case. Note, however, that setup times continue to be the most significant portion of total CPU times and the largest instances are still solved in less than 54 minutes in average. This is just 4 minutes longer than in the baseline case.

4.5.3.3.3. Effect of Minimum Volume Required to Open a Relay Point

A design parameter that interests TL carriers is the establishment of a minimum volume required to justify opening and operating a RP. The installation of a RP is justified as long as the location is expected to handle a given proportion of the total demand in the system. For this reason, we considered values of $v = 2\%$ and $v = 5\%$ of the total truckload demand to require moderate and high truckload volume respectively. A value of zero for this parameter was established in the baseline scenario, so that even locations with low truckload traffic were

allowed to become RPs. Table A.4 and Table A.5 in Appendix A show the results for moderate and high truckload volume required to open a RP in respective order.

In both scenarios, we observed that the limitation becomes restrictive as the size of the network and freight density increase. The moderate volume requirement only starts to affect problems with 100 nodes and 40% freight density. A reduction in the number of open RPs along with a significant increase in solution times are observed for these instances. However, total costs increase only slightly as a result of this constraint. It is clear that as the size of the network increases to 150 nodes, this limitation becomes very restrictive affecting the performance of our formulation and requiring longer solution times. Although instances with 20% freight density are still solved by CPLEX, problems with higher freight density (i.e., 40%) cannot be completely solved and CPLEX stopped due to lack of memory as more nodes in the branch-and-cut tree are needed. Interestingly, we noticed that for the majority of the time required to solve the model, it maintains an initial feasible solution with all truckloads dispatched PtP. When the first mixed fleet solution is found, the optimality gap drops substantially (typically to <1%) and a final solution is found relatively soon after. For the largest instances that we were able to solve, we observed a worst case solution time of 86 minutes on top of the time required to setup the problem.

The same general behavior is observed in the high volume requirement scenario. This means reductions in the number of open RPs and increased solution times are observed for instances in which the constraint becomes restrictive. However, since the limitation is stricter in this case, its effect is more significant and starts to be noticed in smaller instances as compared to the previous case (i.e., fewer RPs and longer solution times are observed for 50 node networks

with 40% freight density). Still, high quality solutions for mixed fleet dispatching designs are able to be obtained for problems with up to 150 nodes and 10% freight density. After the problem is setup, these instances can be solved in less than 4 hours and 15 minutes in the worst case with an average solution time of 2 hours and 40 minutes. Once again, memory issues prevent us from solving larger problem instances with 20% and 40% freight density. In general, these results are an indication that this parameter is very important for the design of mixed fleet dispatching systems of considerable size since it can limit significantly the number of open RPs in the network, and has implications both on the solutions obtained and the tractability of our model.

Figure 22 shows the differences in solution values and number of open RPs between a RN-only system when this limitation is enforced and the mixed fleet dispatching system for problems with 100 nodes and 10% freight density. Note that as the minimum volume required to open a RP increases the savings associated with the mixed fleet system become more apparent as fewer RPs are needed and more loads can be dispatched PtP.

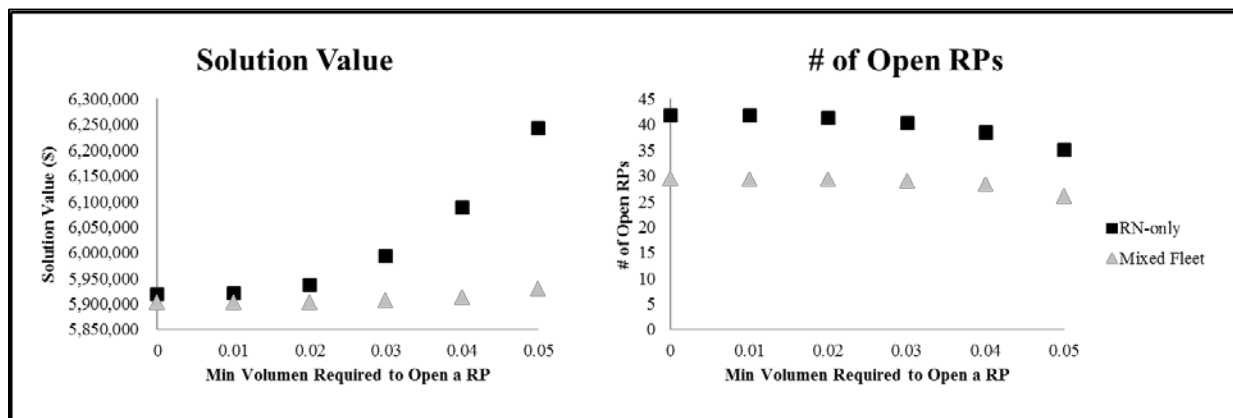


Figure 22. Differences Between RN-only and Mixed Fleet as Minimum Volume Required to Open RPs Varies.

4.5.3.3.4. Effect of Equipment Balance for Relay Network Loads

The baseline scenario requires perfect balance for the loads that are dispatched through the relay network. We decided to relax this limitation and allow some imbalance at the nodes in the network to analyze the effect of this operational constraint on the performance of our model and the solutions obtained. The results for a maximum percentage imbalance allowed of $\delta = 0.3$ are presented in Table A.6 of Appendix A.

From the results obtained, no significant differences are observed for the solution values as compared to the baseline scenario with perfect balance. However, we note a slight increase in the number of RPs that are open in networks with 150 nodes. This increment in the number of facilities in the RN is not accompanied by a significant change in the proportion of loads that are dispatched PtP.

More importantly, we were able to determine that balance comes at a cost in terms of CPU time. As observed in these results, with the exception of the small networks with less freight density, allowing some imbalance at the nodes in the RN results in solution times that are significantly faster than the baseline scenario. Nonetheless, since setup times are once again the most important portion of total CPU time, the overall benefits are not as significant. The average total CPU time for the largest instances is only reduced in 3 minutes, decreasing from 50 minutes in the perfect balance case to 47 minutes when this constraint is relaxed.

The effect of equipment imbalance at the nodes in the relay network in terms of solution values and number of open RPs can be observed in Figure 23. This figure also shows the differences between the mixed fleet dispatching system and a RN-only system as this limitation varies from requiring perfect balance to allowing 100% imbalance. We observe that the savings

associated with the mixed fleet configuration remain consistent across the different levels of allowed imbalance at the nodes.

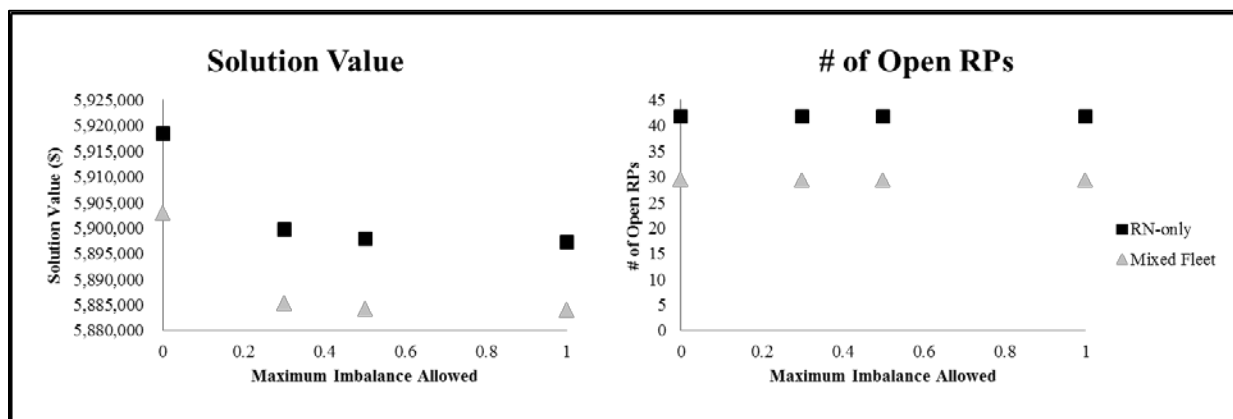


Figure 23. Differences Between RN-only and Mixed Fleet as Maximum Imbalance Allowed Varies.

4.5.3.3.5. Effect of Fixed Cost of Installation of Relay Points

TL carriers that operate nationally have difficulty estimating appropriate installation and operation costs for their facilities given the differences in land and labor costs across regions in the U.S. The value of \$10,000 used in the baseline scenario can be considered relatively low, but is justified by the fact that RPs do not need expensive equipment and infrastructure to operate. However, in this scenario we considered a fixed cost of installation for the RPs that is ten times higher (i.e., \$100,000) to test the performance of our model and determine how this parameter affects the characteristics of the solutions obtained. The results for a higher fixed cost of installation of \$100,000 are presented in Table A.7 of Appendix A.

We observe that the higher fixed cost of installation results in PtP-only solutions for problems with 50 nodes and freight densities of 10% and 20%. In these cases, as the capital investment to open RPs becomes more significant, the reduced truckload volume is insufficient

to justify opening RPs as the expense associated with their installation cannot be outweighed by the reduction in transportation costs.

Solutions with a mixed fleet configuration begin to occur for instances with 50 nodes and 40% freight density. The higher fixed cost of installation results in mixed fleet systems with significantly fewer RPs. This consequently affects the proportion of loads that are dispatched PtP and solution values. The reduced number of RPs becomes more significant as the number of nodes and freight density increase. This reduction goes from 16 to 47 fewer RPs with respect to the baseline case. Solution values increase significantly but never exceed a 25% increase over the baseline scenario. As the instance sizes grow, this objective function increase becomes less significant. In the same way, the increase in the proportion of loads that are dispatched PtP is more significant for smaller instances. In these cases, the proportion of loads dispatched PtP is almost two times higher than in the baseline scenario.

The higher fixed cost of installation also affects solution times significantly. The increase in solution times always exceeds 750% with respect to the baseline scenario as observed for instances with 50 nodes and 40% freight density. As the number of nodes and freight density increase this increase is even more significant. The largest instances solved without running into memory issues are those with 150 nodes and 20% freight density. The worst case solution time observed for these problems is 3 hours and 2 minutes with an average solution time of 2 hours and 46 minutes.

Figure 24 shows that as the fixed cost of installation of the RPs increases the differences between RN-only and mixed fleet dispatching become more apparent from an economic perspective. For a fixed cost of \$100,000, the savings associated with the hybrid configuration

ascend to 20.5% with respect to the pure RN system. In addition, it can be observed that as fixed cost increases the number of open RPs in the RN-only system converges to a value required for feasibility reasons. Since loads can also be dispatched PtP in the mixed fleet system, the number of open RPs continue to decrease as higher fixed costs of installation are considered.

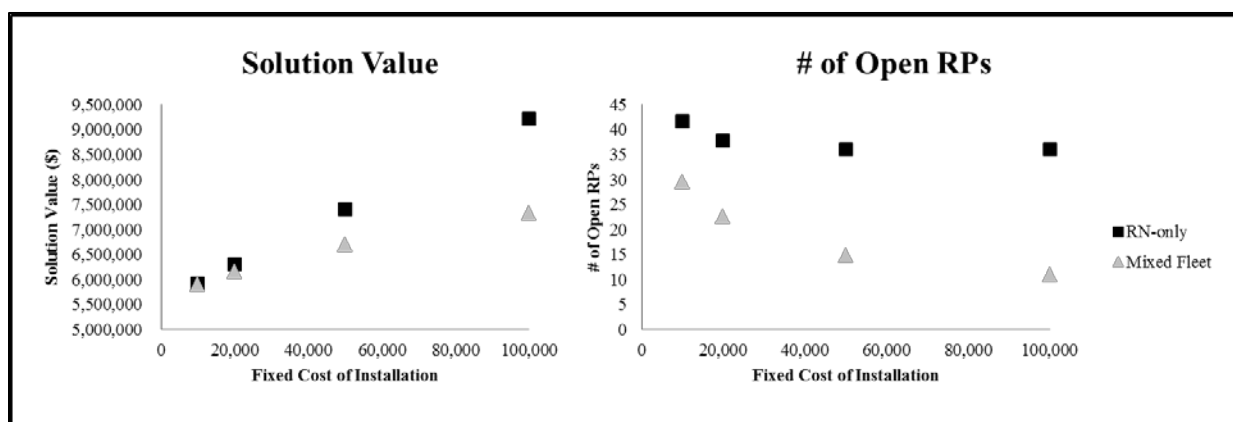


Figure 24. Differences Between RN-only and Mixed Fleet as Fixed Cost of RPs Varies.

4.5.3.3.6. Discussion of Sensitivity Analysis Results

In our computational experiments, we observed that with the exception of minimum volume required to open a RP and fixed cost of installation, changes to the design parameters for TLRND-MD result in solutions with very similar total costs. However, it is important to note that although the solutions obtained are very similar from an economic perspective, they are all different in terms of the configuration of the mixed fleet dispatching system obtained with our model since different nodes are selected as RPs and varying proportions of loads that are dispatched PtP. This highlights the importance of our prescriptive model since very different solutions with very similar solution values can be obtained when making changes to several design parameters of this problem.

Figure 25 shows the general trends observed for the changes in solution characteristics and model performance for each scenario with respect to the baseline scenario. The values

presented in this figure provide an idea of the significance of these changes and a sense of how they affect different instances across different network sizes and freight densities. For example, the low PtP repositioning cost produces a reduction in solution times that is variable across different instances but always less than 2% with respect to the baseline case. This scenario also results in a reduction in the number of open RPs that is consistent across different problems with an average of 2 fewer RPs with respect to the baseline. Also, the change in the proportion of PtP loads is variable across different instances with an increase that is as high as 10.45 percentage points for smaller problems and as little as 0.13 percentage points for the largest problems. Finally, when PtP repositioning costs are low, there is a reduction in solution times that never exceeds 45% with respect to the baseline case. The results for the remaining scenarios can be read in a similar way.

| Scenario | Solution Value | # of RPs | Proportion of PtP Loads | Solution Time |
|--|----------------|----------|-------------------------|---------------|
| Low PtP Repositioning Cost | ↓ < 2% | ↓ ≈ 2 | ↑ 10.45 0.13 | ↓ < 45% |
| High PtP Repositioning Cost | ↑ < 2% | ↑ ≈ 1.5 | ↓ 4.58 0.11 | ↑ < 60% |
| Limited Proportion PtP Loads | ↑ ≤ 1% | ↑ < 8 | ↓ 10.21 0.06 | ↑ < 165% |
| Moderate Vol. Req. to Open a RP | ↑ < 2% | ↓ > 4 | ↑ > 0.4 | ↑ > 210% |
| High Vol. Req. to Open a RP | ↑ < 2.2% | ↓ > 2.5 | ↑ > 1.2 | ↑ > 240% |
| Some Imbalance Allowed | ↓ < 2% | ↑ ≈ 2.2 | ↓ ≈ 0.5 | ↓ < 80% |
| High Fixed Cost of Installation of RPs | ↑ > 20% | ↓ > 15 | ↑ 20.32 7.86 | ↑ > 750% |

Figure 25. Summary of Effects of Design Parameters.

As observed in Figure 25, the most significant changes in solutions obtained and model performance are observed for the scenarios in which we enforce a limitation on the minimum

volume required to open a RP and the scenario with a higher fixed cost of installation. Consequently, TL carriers need to consider these two parameters very carefully when designing mixed fleet dispatching systems.

4.5.3.4. Test Case Results

To test the performance of our model with realistically-sized problem instances, we used test case data provided by J.B. Hunt Transportation Services, one of the largest TL carriers in the United States. Origins and destinations for truckloads in the eastern half of the U.S. were aggregated at the three-digit zip code level resulting in a network with 623 nodes where a node is placed on the centroid of the aggregated origins and destinations. These nodes are connected by a total of 83,734 arcs. This is an arc density of 21.61% which is significantly lower than the arc density of 100% of the complete networks used in our previous experiments. The distances on the arcs are over the road miles between connected nodes in the transportation network. A total of 1,386 O-D node pairs with truckload flows are included in this problem instance. Also, we considered limitations on the distance allowed for local drivers of $\gamma_1 = 225$ miles and distance allowed for lane drivers of $\gamma_2 = 450$ miles. All other problem parameters were not modified in the experimental design. Table 20 shows the results for the baseline scenario of TLRND-MD described in Section 4.5.1. This table also presents the values observed for RN-only and PtP-only dispatching methods to compare them to the mixed fleet dispatching system.

As observed in this table, the mixed fleet dispatching system has the lowest solution value of the three configurations. Similar to the randomly generated instances previously analyzed, the difference is more significant with respect to the PtP-only system that has a total cost that is 22.56% higher. On the other hand, the cost of the pure RN system is only 1.9%

higher than the mixed fleet method. The most noticeable difference between RN-only and mixed fleet dispatching relates to the number of open RPs. Allowing some truckloads to be dispatched PtP results in a considerable reduction of 34 RPs or 25% fewer facilities than in the RN-only alternative. As previously mentioned, this reduction translates into operational advantages for carriers such as fewer drivers' domiciles to manage and increased utilization. This reduction in the number of RPs needed is accomplished by shipping close to 92% of the truckloads through the relay network while the rest is dispatched PtP.

Table 20. Results for Test Case (623 Nodes and 1,386 O-D Pairs with Truckload Demand).

| Dispatching Method | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Average Length of Haul (miles) | Average Service Time (hours) | Maximum Service Time (hours) | Setup Time (secs) | Solution Time (secs) |
|--------------------|---------------------|------------|-------------------------|--------------------------------|------------------------------|------------------------------|-------------------|----------------------|
| Mixed Fleet | 19,613,127 | 102 | 8.42% | 310.15 (RN) 748.62 (PtP) | 18.94 (RN) 29.99 (PtP) | 35.39 (RN) 64.62 (PtP) | 9,894.34 | 1,648.86 |
| RN-only | 19,986,727 | 136 | 0.00% | 314.73 | 19.25 | 35.70 | 11,577.23 | 1,025.67 |
| PtP-only | 24,038,329 | 0 | 100.00% | 638.08 | 25.37 | 64.62 | N/A | N/A |

Average length of haul and service times for the majority of the truckloads in the mixed fleet system (i.e., loads that are shipped through the relay network) are comparable to those in the RN-only method and are considerably better than in traditional PtP dispatching. Unfortunately, the small portion of loads that are dispatched PtP in the mixed fleet system show relatively high average length of haul and service times. In contrast to the previous experiments, there is a higher proportion of direct shipments that cover longer distances. This mostly occurs when a direct shipment is used to satisfy demand at isolated nodes in the network.

In terms of the performance of our formulation of TLRND-MD and the heuristic solution approach developed in this research, setup times continue to dominate total CPU time. The large number of composite variables that are generated (i.e., close to 2 million composites) requires approximately 2.5 hours of setup time to build the model. Fortunately, CPLEX takes less than

28 minutes to solve the problem with an optimality gap of 1.0%. The solution of this test case shows that our approach can be used to obtain high quality solutions for largely-sized problem instances that are likely to appear in practical settings.

4.6. Conclusions

This chapter presents a prescriptive model for the design of a mixed fleet dispatching system for TL transportation. This alternative dispatching method combines direct shipments from origin to destination with the use of a network configuration of relay points. To the best of our knowledge, this is the first prescriptive model for such systems.

We propose a composite variable model that implicitly considers several operational constraints within the definition of variables that represent feasible routes for truckloads through the relay network. The selection of the dispatching mode and routing for the truckloads is handled by an integer programming model that is used to obtain high quality solutions for largely-sized problem instances.

Both TL carriers and researchers have suggested that mixed fleet networks show significant promise and are more likely to be successful in practice than a RN alone. Our results show that from a cost perspective the differences between relay networks and mixed fleet dispatching systems are not significantly different when fixed cost of installation of the RPs is low with an average difference of only around 2% in the best case. However, as the fixed cost of installation increases, mixed fleet dispatching systems show more significant savings over a pure RN system. In addition, mixed fleet network configurations always require fewer relay points which may offer other advantages over RN-only solutions in terms of higher utilization of facilities and fewer domiciles to manage. Note that our methodology can be used to assess

whether these findings are true for other network topologies and demand patterns. In addition, our experiments show that both systems – RN-only and mixed fleet – outperform the traditional PtP-only dispatching system in total costs and other performance metrics of interest such as average length of haul and service times for the truckloads.

Our results suggest that truckloads that have origin and destination nodes not too distant apart are good candidates for PtP dispatching. In the same way, truckloads that either have the origin node or the destination node in an isolated area are best served PtP. As the proportion of truckloads that are shipped directly is usually small, TL carriers can expect to improve the quality of the driving jobs for the majority of their people using this alternative dispatching method while considerably reducing the number of RPs required in the network.

Changes to several design parameters of the mixed fleet systems result in alternative configurations with different RPs and varying proportions of truckloads shipped PtP. However, these changes seem to only minimally affect the solution values (i.e., total costs) obtained. A general observation across different scenarios is that dispatching through the relay network is preferable for a higher proportion of the truckloads as network size and freight density increase.

Also, we determined that the performance of our CVM formulation for TLRND-MD is consistent across several scenarios with the exception of cases where very strict limitations are imposed on the minimum volume required to open a RP and when higher fixed costs of installation for RPs are considered. Extended solution times are required for larger problem instances in these cases. Moreover, the heuristic approach developed in this research was successfully used to obtain a solution for a practical problem provided by a major TL carrier in reasonable time, especially considering that this is an integrated strategic design problem.

There are some challenges that need to be addressed as part of future work such as explicitly incorporating a balance constraint for truckloads that are dispatched PtP instead of using a repositioning cost as a surrogate. Another important area for future research is to assess whether or not mixed fleet dispatching systems can significantly outperform RN-only configurations for certain network topologies, or whether our observations that for low fixed costs of installation of the RPs the cost implications of a mixed fleet dispatching system are not significant extend to all types of network configurations. If secondary performance metrics beyond total cost can be identified to assess the value of mixed fleet dispatching systems, the use of a multi-objective optimization approach may be justified for this problem based on the effect of alternative dispatching systems on metrics that interest not only carriers but also affect drivers and shippers.

In addition, we would like to explore other methods for the generation of the composite variables that are used in our formulation. A column generation approach using a multi-label shortest path algorithm seems to be a viable alternative. This would also allow us to solve problems where more than three RPs are allowed to be visited in a route without significantly increasing the number of variables needed in our model. Alternatively, a method to generate the composites in parallel for different truckloads will help to reduce setup times even further, especially for largely-sized problems.

Finally, driver considerations and timing for the truckloads should be incorporated in order to develop efficient schedules for the drivers. The benefits of this and other alternative dispatching systems in terms of driver retention will only be attainable as long as TL carriers are able to construct driving duties that bring drivers home more frequently.

Acknowledgments

We gratefully thank J.B. Hunt Transportation Services for providing data for the test case presented in this research.

Appendix A

Table A.1. Results for Low PtP Repositioning Cost.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|-------------|------------------------|----------------|-------------------------|----------------------|----------------------|
| 50 | 245 | 1,619,427 (-2.00%) | 12 (-3.2) | 30.66% (+10.45) | 5.05 (-0.18%) | 2.74 (+26.64%) |
| | 490 | 3,121,304 (-0.95%) | 18.2 (-1.5) | 18.14% (+3.64) | 9.59 (+0.18%) | 1.54 (-11.04%) |
| | 980 | 6,006,667 (-0.68%) | 23.5 (-1.8) | 13.73% (+2.21) | 18.68 (+0.15%) | 1.28 (-40.99%) |
| 100 | 990 | 5,874,232 (-0.48%) | 28.3 (-1.1) | 11.20% (+1.11) | 70.24 (+0.17%) | 75.59 (-36.31%) |
| | 1,980 | 11,108,342 (-0.40%) | 38.0 (-1.3) | 9.67% (+0.44) | 143.56 (-) | 36.51 (-28.52%) |
| | 3,960 | 21,895,843 (-0.34%) | 49.6 (-1.0) | 8.78% (+0.21) | 311.47 (-0.33%) | 41.68 (-10.77%) |
| 150 | 2,235 | 12,265,220 (-0.37%) | 54.1 (-1.5) | 8.57% (+0.27) | 408.61 (-0.05%) | 262.11 (-1.61%) |
| | 4,470 | 23,814,251 (-0.39%) | 64.3 (-4.4) | 8.20% (+0.18) | 991.97 (-0.02%) | 159.89 (-3.40%) |
| | 8,940 | 47,085,717 (-0.31%) | 75.4 (-1.5) | 8.13% (+0.13) | 2,723.96 (+0.06%) | 230.97 (-8.74%) |

Table A.2. Results for High PtP Repositioning Cost.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|-------------|------------------------|----------------|-------------------------|----------------------|----------------------|
| 50 | 245 | 1,684,057 (+1.92%) | 17.4 (+2.2) | 15.63% (-4.58) | 5.06 (+0.11%) | 2.87 (+32.66%) |
| | 490 | 3,185,633 (+1.09%) | 21.2 (+1.5) | 12.89% (-1.62) | 9.60 (+0.28%) | 1.69 (-2.32%) |
| | 980 | 6,098,415 (+0.84%) | 27.0 (+1.7) | 10.43% (-1.09) | 18.71 (+0.26%) | 2.38 (+9.43%) |
| 100 | 990 | 5,943,406 (+0.69%) | 31.0 (+1.6) | 9.29% (-0.80) | 70.10 (-0.03%) | 111.78 (-5.82%) |
| | 1,980 | 11,219,937 (+0.60%) | 41.0 (+1.7) | 8.87% (-0.36) | 143.58 (+0.02%) | 35.51 (-30.47%) |
| | 3,960 | 22,089,934 (+0.54%) | 52.6 (+2.0) | 8.32% (-0.25) | 312.18 (-0.10%) | 34.77 (-25.57%) |
| 150 | 2,235 | 12,366,688 (+0.45%) | 54.0 (-1.6) | 8.13% (-0.17) | 408.94 (+0.03%) | 415.61 (+56.01%) |
| | 4,470 | 24,033,820 (+0.53%) | 71.0 (+2.3) | 7.88% (-0.14) | 990.13 (-0.20%) | 168.19 (+1.62%) |
| | 8,940 | 47,465,632 (+0.50%) | 79 (+2.1) | 7.89% (-0.11) | 2,720.51 (-0.06%) | 262.92 (+3.88%) |

Table A.3. Results for Limited Proportion of PtP Loads.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) | # of Infeasible Instances |
|-------|-------------|-------------------------|----------------|-------------------------|----------------------|----------------------|---------------------------|
| 50 | 245 | 1,669,985 (+1.06%) | 21.6 (+6.4) | 10.00% (-10.21) | 4.93 (-2.52%) | 3.72 (+71.77%) | 3 |
| | 490 | 3,174,001 (+0.72%) | 27.4 (+7.7) | 10.00% (-4.50) | 9.41 (-1.78%) | 4.56 (+164.25%) | 2 |
| | 980 | 6,044,016 (-0.06%) | 28.6 (+3.3) | 9.98% (-1.54) | 18.30 (-1.91%) | 4.52 (+107.97%) | 1 |
| 100 | 990 | 5,907,329 (+0.08%) | 30.6 (+1.2) | 9.64% (-0.45) | 70.14 (+0.02%) | 200.65 (+69.07%) | 0 |
| | 1,980 | 11,115,816 (+0.03%) | 39.9 (+0.6) | 9.10% (-0.14) | 143.64 (+0.06%) | 40.81 (-20.10%) | 0 |
| | 3,960 | 21,971,203 (+0.001%) | 50.5 (-0.1) | 8.57% (-) | 312.16 (-0.11%) | 45.33 (-2.95%) | 0 |
| 150 | 2,235 | 12,323,958 (+0.11%) | 57.7 (+2.1) | 8.31% (+0.01) | 408.79 (-0.01%) | 353.18 (+32.58%) | 0 |
| | 4,470 | 23,962,971 (+0.24%) | 73.8 (+5.1) | 7.96% (-0.06) | 990.58 (-0.16%) | 246.20 (+148.75%) | 0 |
| | 8,940 | 47,100,056 (-0.28%) | 74.8 (-2.1) | 8.06% (+0.06) | 2,747.40 (+0.93%) | 477.65 (+188.73%) | 1 |

Table A.4. Results for Moderate Volume Required to Open a RP.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|--------------------|------------------------|-----------------|-------------------------|----------------------|--------------------------|
| 50 | 245 | 1,652,414 (+0.001%) | 15.3 (+0.1) | 20.17% (-0.04) | 5.08 (+0.39%) | 2.82 (+30.59%) |
| | 490 | 3,151,213 (+0.001%) | 19.7 (-) | 14.53% (-0.03) | 9.64 (+0.65%) | 1.65 (-4.40%) |
| | 980 | 6,047,985 (+0.006%) | 25.2 (-0.1) | 11.61% (+0.09) | 18.79 (+0.74%) | 1.82 (-16.26%) |
| 100 | 990 | 5,902,990 (+0.002%) | 29.2 (-0.2) | 10.08% (-0.01) | 71.40 (+1.83%) | 104.77 (-11.72%) |
| | 1,980 | 11,156,163 (+0.03%) | 38.8 (-0.5) | 9.39% (+0.15) | 144.11 (+0.39%) | 50.58 (-0.95%) |
| | 3,960 | 22,001,047 (+0.14%) | 46.4 (-4.2) | 9.00% (+0.43) | 311.57 (-0.30%) | 154.75 (+231.29%) |
| 150 | 2,235 | 12,291,190 (-0.16%) | 45.5 (-10.1) | 9.01% (-0.71) | 411.20 (+0.58%) | 833.83 (+213.00%) |
| | 4,470 | 23,939,619 (+1.38%) | 55.3 (-13.5) | 8.81% (+0.80) | 989.78 (-0.24%) | 2,228.59 (+1,246.48%) |
| | 8,940 ^a | --- | --- | --- | 2,747.40 (+0.93%) | --- |

^a CPLEX out of memory while exploring branch-and-cut tree

Table A.5. Results for High Volume Required to Open a RP.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|--------------------|------------------------|-----------------|-------------------------|----------------------|---------------------------|
| 50 | 245 | 1,653,791 (+0.08%) | 15.0 (-0.2) | 20.57% (+0.36) | 5.07 (+0.18%) | 2.84 (+31.12%) |
| | 490 | 3,153,772 (+0.08%) | 19.2 (-0.5) | 14.79% (+0.29) | 9.57 (-0.10%) | 2.32 (+34.47%) |
| | 980 | 6,065,296 (+0.29%) | 22.6 (-2.7) | 12.74% (+1.22) | 18.79 (+0.69%) | 7.47 (+243.62%) |
| 100 | 990 | 5,928,106 (+0.43%) | 25.9 (-3.5) | 11.26% (+1.18) | 71.48 (+1.94%) | 527.51 (+344.47%) |
| | 1,980 | 11,278,022 (+1.12%) | 28.6 (-10.7) | 11.12% (+1.88) | 143.46 (-0.07%) | 4,553.36 (+8,815.00%) |
| | 3,960 | 22,443,812 (+2.15%) | 29.3 (-21.3) | 11.01% (+2.44) | 310.16 (-0.75%) | 6,387.58 (+13,574.75%) |
| 150 | 2,235 | 12,450,410 (+1.13%) | 28.9 (-26.7) | 10.65% (+2.34) | 410.54 (+0.42%) | 9,651.55 (+3,522.95%) |
| | 4,470 ^b | --- | --- | --- | 987.66 (-0.45%) | --- |
| | 8,940 ^b | --- | --- | --- | 2,720.42 (-0.93%) | --- |

^b CPLEX out of memory while exploring branch-and-cut tree

Table A.6. Results for Some Imbalance Allowed at Nodes for RN Loads.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|-------------|------------------------|----------------|-------------------------|----------------------|----------------------|
| 50 | 245 | 1,633,912 (-1.12%) | 15.1 (-0.1) | 16.90% (-3.31) | 5.06 (+0.06%) | 5.97 (+175.77%) |
| | 490 | 3,133,419 (-0.56%) | 19.5 (-0.2) | 13.88% (-0.62) | 9.56 (-0.16%) | 2.26 (+30.74%) |
| | 980 | 6,025,640 (-0.36%) | 25.0 (-0.3) | 11.13% (-0.39) | 18.68 (+0.12%) | 0.91 (-58.36%) |
| 100 | 990 | 5,885,231 (-0.30%) | 29.2 (-0.2) | 10.16% (+0.07) | 70.08 (-0.06%) | 27.54 (-76.80%) |
| | 1,980 | 11,132,782 (-0.18%) | 39.8 (+0.5) | 9.18% (-0.06) | 143.82 (+0.19%) | 11.92 (-76.65%) |
| | 3,960 | 21,958,981 (-0.05%) | 50.7 (+0.1) | 8.55% (-0.02) | 312.41 (-0.03%) | 12.96 (-72.25%) |
| 150 | 2,235 | 12,323,958 (+0.11%) | 57.7 (+2.1) | 8.32% (+0.02) | 408.95 (+0.03%) | 73.59 (-72.38%) |
| | 4,470 | 23,957,141 (+0.21%) | 74.4 (+5.7) | 7.91% (-0.11) | 990.57 (-0.16%) | 32.35 (-80.45%) |
| | 8,940 | 47,245,013 (+0.03%) | 79.5 (+2.6) | 7.97% (-0.03) | 2,729.60 (+0.27%) | 56.75 (-77.58%) |

Table A.7. Results for High Fixed Cost of Installation of RPs.

| Nodes | # O-D Pairs | Solution Value (\$) | # RPs Open | Proportion of PtP Loads | Setup Time (secs) | Solution Time (secs) |
|-------|--------------------|-------------------------|-----------------|-------------------------|----------------------|--------------------------|
| 50 | 245 | 1,913,915 (+15.83%) | 0.0 (-15.2) | 100.00% (+79.79) | 4.17 (-17.58%) | 5.01 (+131.19%) |
| | 490 | 3,884,855 (+23.28%) | 0.0 (-19.7) | 100.00% (+85.50) | 7.90 (-17.54%) | 9.50 (+450.05%) |
| | 980 | 7,329,919 (+21.20%) | 9.5 (-15.8) | 31.84% (+20.32) | 15.45 (-17.21%) | 18.60 (+755.86%) |
| 100 | 990 | 7,321,420 (+24.03%) | 10.8 (-18.6) | 27.66% (+17.58) | 70.33 (+0.30%) | 1,011.92 (+752.63%) |
| | 1,980 | 13,069,119 (+17.18%) | 14.6 (-24.7) | 18.26% (+9.02) | 143.75 (+0.14%) | 2,843.32 (+5,466.92%) |
| | 3,960 | 24,551,334 (+11.74%) | 19.8 (-30.8) | 13.72% (+5.15) | 309.66 (-0.91%) | 4,253.72 (+9,006.51%) |
| 150 | 2,235 | 14,334,604 (+16.44%) | 16.2 (-39.4) | 16.16% (+7.86) | 408.19 (-0.15%) | 4,297.87 (+1,513.31%) |
| | 4,470 | 26,584,614 (+11.20%) | 22.3 (-46.5) | 12.26% (+4.24) | 1,003.10 (+1.11%) | 9,942.02 (+5,906.82%) |
| | 8,940 ^c | --- | --- | --- | 2,732.72 (+0.39%) | --- |

^c CPLEX out of memory while exploring branch-and-cut tree

References

- Alam, M., Fekpe, E., and Majed, M. (2007). *Freight Analysis Framework 2: Freight Traffic Analysis*. Washington, D.C.: Federal Highway Administration, Office of Freight Management and Operations.
- Ali, T., Radhakrishnan, S., Pulat, S., and Gaddipati, N. (2002). Relay Network Design in Freight Transportation Systems. *Transportation Research Part E*, 38, 405-422.
- American Transportation Research Institute. (2011). *Critical Issues in the Trucking Industry - 2011*. Arlington, VA: American Transportation Research Institute (ATRI).
- American Trucking Associations. (2007). *Trucking and the Economy*. Arlington, VA: American Trucking Associations (ATA).
- American Trucking Associations. (2011). *Trucking Activity Report*. Arlington, VA: American Trucking Associations (ATA).
- Gupta, N., Jenkins, G., and Delery, J. (1996). *Motor Carrier Effectiveness*. Fayetteville, AR: Mack Blackwell Rural Transportation Center: University of Arkansas.
- Hunt, G. (1998). *Transportation Relay Network Design*. Atlanta, GA: Ph.D. Thesis, Georgia Institute of Technology.

- Keller, S.B., and Ozment, J. (1999). Managing Driver Retention: Effects of the Dispatcher. *Journal of Business Logistics*, 20(2), 97-119.
- Liu, J., Li, C.L., and Chan, C.Y. (2003). Mixed Truck Delivery Systems with Both Hub-and-Spoke and Direct Shipment. *Transportation Research Part E*, 39, 325-339.
- Lockridge, D. (2008). Top 10 Reasons Drivers Leave. *Heavy Duty Trucking: Delivering the World of Trucks*, 87(1), pp. 46-51.
- Melton, K., and Ingalls, R. (2012). Truckload Freight Transportation Utilizing Relay Points to Improve Driving Jobs. Orlando: Proceedings of the 2012 Industrial and Systems Engineering Research Conference.
- Morris, F. (2011). *Thousands of Trucking Jobs, But Few Take the Wheel*. Retrieved from: <http://www.npr.org/2011/10/31/141791952/thousands-of-trucking-jobs-but-few-take-the-wheel>.
- Rodriguez, J., Kosir, M., Lantz, B., Griffin, G., and Glatt, J. (2000). *The Costs of Truckload Driver Turnover*. Fargo, ND: Upper Great Plains Transportation Institute.
- Ronen, D. (1997). Alternate Mode Dispatching: The Impact of Cost Minimisation. *Journal of the Operational Research Society*, 48, 973-977.
- Taha, T., and Taylor, G.D. (1994). An Integrated Modeling Framework for Evaluating Hub-and-Spoke Networks in Truckload Trucking. *Logistics and Transportation Review*, 30(2), 141-166.
- Taylor, G.D., DuCote, W.G., and Whicker, G.L. (2006). Regional Fleet Design in Truckload Trucking. *Transportation Research Part E*, 42, 167-190.
- Taylor, G., Meinert, T., Killian, R., and Whicker, G. (1999). Development and Analysis of Alternative Dispatching Methods in Truckload Trucking. *Transportation Research Part E*, 35, 191-205.
- Taylor, G.D., and Whicker, G.L. (2002). Optimization and Heuristic Methods Supporting Distributed Manufacturing. *Production Planning & Control*, 13(6), 517-528.
- Taylor, G.D., and Whicker, G.L. (2008). Concurrent Operation of Mixed Fleet Types in Truckload Trucking. In K. Ellis, R. Meller, M. Ogle, B. Peters, G. Taylor, and J. Usher (Eds.), *Progress in Material Handling Research: 2008* (pp. 638-661). Charlotte, NC: Material Handling Industry.

- Taylor, G.D., and Whicker, G.L. (2010). Extended Regional Dispatch for Truckload Carriers. *International Journal of Physical Distribution and Logistics Management*, 40(6), 495-515.
- Taylor, G.D., Whicker, G.L., and DuCote, W.G. (2009). Design and Analysis of Delivery 'Pipelines' in Truckload Trucking. *Transportation Research Part E*, 45, 255-269.
- Taylor, G.D., Whicker, G.L., and Usher, J.S. (2001). Multi-Zone Dispatching in Truckload Trucking. *Transportation Research: Part E*, 37, 375-390.
- U.S. Bureau of Transportation Statistics. (2010). *Transportation: 2007 Commodity Flow Survey*.
- Üster, H., and Kewcharoenwong, P. (2011). Strategic Design and Analysis of a Relay Network in Truckload Transportation. *Transportation Science*, 45(4), 505-523.
- Üster, H., and Maheshwari, N. (2007). Strategic Network Design for Multi-Zone Truckload Shipments. *IIE Transactions*, 39, 177-189.

5. Conclusion

5.1. Research Conclusions

This dissertation is motivated by an important problem for the TL transportation industry. The excessive driver turnover experienced by large and small TL carriers has motivated industry and academic researchers to explore the development and analysis of alternative dispatching methods to the traditional point-to-point dispatching system. In this context, the use of relay networks for TL transportation has been recognized in previous research as a valid alternative to improve driver retention. This dissertation mainly addresses the need for prescriptive mathematical models and solution approaches to solve largely-sized strategic relay network design problems. These problems are complicated by the need to incorporate operational constraints imposed by TL carriers to overcome some of the challenges of dispatching truckloads through a network configuration of relay points and to effectively improve driving jobs. We contribute to the literature with mathematical formulations and solution approaches that can be effectively used to solve practically motivated instances of TLRND problems.

We first present the development and analysis of an algorithmic approach for the solution of a subproblem related to TLRND. Determining the number and location of relay points in a TL relay network requires the generation of feasible routes for the truckloads. An exact algorithm based on topological sorting is developed for the enumeration of all feasible paths between an origin node and a destination node in a directed acyclic graph. This algorithm uses a labeling approach to simultaneously enforce limitations on total length and number of nodes visited in a path. The approach presented in this research overcomes several shortcomings of existing methods in the path enumeration literature. For example, one class of algorithms

considers the side constraints to eliminate the paths that are infeasible only after all paths in the network have been enumerated. In general, those methods are less efficient especially when networks are large and the number of feasible paths is small. A second class of algorithms can generate the k -shortest paths. This requires the user to set a value for k to determine these paths. As the number of feasible paths is commonly not known in advance, values for k are often set to a very large numbers. This affects the efficiency and reliability of k -shortest path algorithms to obtain all feasible paths in the graph. Our methodology overcomes these limitations. Furthermore, using the approach presented in this research, there is no need to check the feasibility of generated paths considering every side constraint sequentially as is the case for several constrained shortest path and constrained k -shortest path algorithms. This also results in efficiency gains over these latter methods.

Computational tests of the proposed algorithm for PENVLC show that it can be used to obtain feasible paths for O-D pairs in DAGs with different number of nodes, area size and arc density. In general, larger computational times are required to solve PENVLC as the number of feasible paths grows. This number is in part affected by the bounds imposed for the side constraints. The proposed algorithm is very efficient when strict limitations are imposed for both path length and number of nodes allowed in a path, but requires longer computational times as the constraints are significantly relaxed. Our results indicate that relaxing the limitation on number of nodes allowed in a path has a relatively greater effect in the performance of the proposed algorithm. The number of feasible paths also depends on the characteristics of the graphs under study. In this regard, arc density has the most significant effect on the total number of feasible paths over network and area size. The algorithm is particularly challenged by

problems with large dense networks since the number of feasible paths is very large for most of these instances. Still, largely-sized problems can be solved in a few seconds using the algorithm when the limitations imposed by the side constraints are maintained at reasonable values as it is the case for feasible route generation in TLRND.

We then present a composite variable model for TLRND that minimizes total transportation and installation costs for the RPs while implicitly incorporating operational constraints relaxed by previous studies within the definition of the variables. The composites used in our formulation represent feasible routes for the truckloads that are generated considering limitations on circuitry, number of RPs allowed to be visited, and distances for local and lane drivers. In previous research, circuitry limitations had been either relaxed or enforced through surrogate constraints while a limitation on the number of RPs allowed to be visited had not been considered in previous prescriptive models. As the algorithm for PENVLC requires additional extensions to be used for this purpose, we develop an alternative enumeration algorithm using pre-defined templates representing possible routes for the truckloads through the relay network. Based on this variable definition, the proposed integer program for TLRND coordinates the selection of RPs and routes for the truckloads that minimizes total costs. This CVM formulation presents advantages over existing arc-based formulations proposed by previous research in this area that are comparable to those expected from path-based formulations. Better lower bounds obtained with this model determine a stronger formulation with improved tractability for larger problem instances than those solved in previous studies. The most significant challenge for the proposed formulation is the generation of composite variables given that a very large number of them is required to ensure optimality. Exact and heuristic solutions for TLRND are obtained

using this model. The heuristic method developed in this research considers a reduced model with a high quality subset of composite variables. Although optimality cannot be guaranteed with this approach, very high quality solutions (i.e., solutions with less than 1% optimality gap) are obtained very efficiently as demonstrated by comparing these results to exact solutions obtained using standard branch-and-cut as implemented in CPLEX. An alternative exact solution method using a branch-and-price approach was also considered, but proved to become inefficient as size of the problems increased and was therefore not pursued further.

The effects of network size and freight density were analyzed using the heuristic approach through computational tests using randomly generated networks. The results show that more RPs are needed as the number of nodes and truckload demand increase. Interestingly, a significant number of RPs are opened only for feasibility reasons as the fixed cost of installation of RPs does not seem to have a significant effect on the design of the relay networks. Our experiments also show that enforcing balance at the nodes in the network only has a marginal effect on the total cost for the system, although it comes at a cost in computational times. In this regard, since it is possible to generate empty movements between nodes in the network as additional variables, we were able to solve perfect balance problems (i.e., requiring that the incoming flow into a node be equal to the outgoing flow). We have not been able to find other approaches in the literature that achieve perfect balance in TLRND. Additionally, our formulation allows assigning origin-destination nodes to multiple RPs based on the selection of the routes for the truckloads. This effectively covers an open area in strategic design of relay networks as identified by previous studies.

Our experiments reveal that computational times are dominated by the time required to generate the composites and build the mathematical model. This setup time is directly proportional to the number of composite variables in the model. On the other hand, solution times seem to depend mostly on the number of nodes in the network as there are more opportunities to locate RPs when more nodes exist in the network. In general, very short solution times are observed for this strategic planning problem even for largely-sized instances with 150 nodes and 40% freight density (i.e., CPLEX takes an average of less than 12 minutes to solve these problems). Moreover, the suitability of our formulation to solve more realistic problem instances of TLRND is demonstrated by obtaining solutions for a test case problem provided by a major TL carrier in reasonable time (i.e., around three and a half hours for setup and solution) considering the scale of this planning problem.

Later in this dissertation, we show that our proposed formulation is flexible to extensions and that other realistic aspects of this problem can be incorporated into the model without negatively affecting its tractability. We extend the CVM formulation of TLRND to include new decisions for the selection of dispatching mode for truckloads between PtP and RN shipments when the carrier operates a mixed fleet dispatching system. We have not been able to find other prescriptive models for the TLRND-MD problem in the literature. Additional constraints imposed by carriers are also included in this extended model such as a limitation on the proportion of loads that are dispatched PtP and the requirement of a minimum truckload volume to open a RP. Also, a repositioning cost is assessed on the cost of transportation of PtP loads as a surrogate to enforce equipment balance for these movements.

Exact solutions for TLRND-MD are obtained for 50 node networks to show that this remains a strong formulation and its tractability is not affected by the extensions. In addition, these results show that most truckloads that have origin and destination nodes not too distant apart (i.e., less than 100 miles) are better served PtP. This observation is used to develop an enhanced heuristic approach with a priori selection of truckloads to be dispatched PtP. This further reduces the number of composite variables in the model. This heuristic approach is then used to complete computational tests with several randomly generated problem instances of different network size and freight density. Results for a baseline scenario of TLRND-MD allow us to quantify the benefit of a mixed fleet dispatching system over RN-only and PtP-only dispatching systems. It is determined that both mixed fleet and RN-only clearly outperform traditional PtP dispatching in total costs and performance metrics for drivers (i.e., lower average length of haul) and customers (i.e., shorter service times). On the other hand, contrary to the predominant expectation of TL carriers and academic researchers who suggest that a hybrid relay network system would have significant benefits over a RN-only system, our experiments do not show significant differences from an economic perspective between the two alternative dispatching systems when fixed costs of installation of the RPs are low. In these cases, the most significant difference observed between these two methods is the reduced number of RPs in the solutions obtained for the mixed fleet system. Since the proportion of loads that are shipped PtP usually remains below 10%, operating fewer RPs represents other operational advantages for carriers such as higher utilization of their facilities and fewer drivers' domiciles to manage. However, as fixed costs of installation of the RPs increase, the difference between mixed fleet and RN-only systems become more significant from an economic perspective with greater

savings for the hybrid system. Furthermore, an analysis of the results obtained for TLRND-MD confirms the basic idea that truckloads with origin or destination nodes located in isolated areas of the network along with truckloads shipped between locations that are not too far apart are good candidates for PtP dispatching. Still, we note that as network size and freight density increase a higher proportion of truckloads is dispatched through the relay network.

Alternative scenarios for TLRND-MD considering changes to some of the parameters of the model seem to have only minimal effect on the total costs of the resulting configurations that display different RPs and varying proportion of truckloads dispatched PtP. Our experimentation shows that strict limitations on the minimum volume required to open a RP and high fixed cost of installation of the RPs have the most significant effect on the solutions obtained by significantly reducing the number of open RPs and also affect the performance of our formulation by requiring longer solution times. Finally, similar to the performance of our model for TLRND with test case data, a baseline scenario of TLRND-MD can be solved for an instance provided by a major TL carrier in reasonable time using our extended formulation. This shows the value of our research in the search for methods that can be used to obtain solutions for problems that exist in practice.

5.2. Future Research Directions

We are able to identify areas for future work that span across the problems studied in this dissertation and extend to other related planning problems in logistics and transportation.

First, it is important to extend the analysis of the economic and operational differences between a mixed fleet dispatching system and a pure relay network configuration to other

network topologies and demand patterns to determine whether or not the results observed in this research are also applicable to other network configurations.

As determined in this research, the most challenging aspect of our approach to strategically design TL relay networks is the generation of composite variables for the CVM formulations of TLRND and TLRND-MD. We would like to explore other methods for composite generation that will allow us to optimally solve largely-sized instances of these problems without significantly affecting computational times. One of the alternatives is to explore the use of a multi-label shortest path algorithm as part of a column generation approach to identify truckload routes that provide the most benefit to the objective functions of our models without necessarily enumerating all feasible routes and adding them to the models.

Although in this case the algorithm for PENVLC developed in this research cannot be effectively used for composite generation as previously explained, it can still be used in other applications that require enumerating alternative paths that satisfy certain limitations. Additional work is required to extend the applicability of our algorithm from directed acyclic graphs to general networks with cycles as several practical problems are modeled in this way. Furthermore, additional side constraints other than total path length and number of nodes visited can be considered through the labeling approach of our algorithm. An interesting extension in this case would be to explicitly consider limitations related to service and maintenance in which time plays a significant role. Other applications such as telecommunication networks and service network design for airlines, railroads, LTL and express shipping carriers may benefit from this type of algorithms for efficient enumeration of feasible paths.

A direct extension of our research is the analysis of the driver scheduling problem for TL carriers that operate alternative dispatching methods such as RN-only and mixed fleet dispatching. To the best of our knowledge no prescriptive models currently exist in the literature to solve this operational problem for the carriers. Some of the challenges associated with obtaining schedules for truck drivers other than the large-scale nature of these problems relate to driver considerations such as enforcing hours-of-service regulations and limitations on how long drivers can be away from their domiciles as well as timing requirements for truckload service. TL carriers should be able to construct driving duties that bring drivers home frequently as truckloads are dispatched using alternative systems to traditional PtP in order to improve driver retention.

Also, as there are multiple players involved in TL freight transportation (i.e., carriers, drivers and shippers) and all of them have their own metrics of interest, an alternative approach using multi-objective optimization may be adequate to solve TLRND and TLRND-MD considering their different perspectives. Moreover, other techniques such as stochastic programming and robust optimization can be applied to these problems to account for uncertainty in the timing and location of the demand for TL freight transportation, risk associated to natural and human disruptions, and other practical aspects of this industry.

Finally, based on the experience with this research, we would like to continue exploring other integrated decision problems in transportation and logistics. We are particularly interested in studying research problems where logistics collaboration and supply chain planning are relevant. These types of problems usually include sourcing decisions, service network design, inventory routing, resource allocation, profit/savings sharing decisions and other strategic,

tactical and operational level decisions. These problems usually present operational complexities that are difficult to handle with traditional modeling approaches. We are interested in using analytical and optimization approaches such as the ones utilized in this dissertation to improve the design and operations of this type of systems and obtain solutions that are more applicable in practice.

We are also interested in problems at the intersection of facility location analysis and supply chain planning. Relevant tactical and operational decisions in supply chain management such as routing and choice of transportation modes have not been extensively integrated with location decisions. The role of facility location is significant in supply chain network planning and it is important to develop models that simultaneously capture many aspects that are relevant to real-world distribution problems. There is also a need for modeling these types of integrated problems considering a more complex structure of the supply chain network as compared to existing literature where a single product and a single location layer are usually assumed. Aspects related to postponement and returns only increase the complexity of these integrated problems of supply chain planning when making location decisions at a strategic level. Studying these types of problems will definitively provide a solid base for extending our research to other problems in logistics collaboration and transportation planning.

