ADVANCES IN LTL LOAD PLAN DESIGN

A Dissertation
Presented to
The Academic Faculty

by

Yang Zhang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2010

ADVANCES IN LTL LOAD PLAN DESIGN

Approved by:

Alan L. Erera, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Martin W. P. Savelsbergh, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Mark Ferguson
College of Management
Georgia Institute of Technology

Ellis Johnson
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Joel Sokol
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 01 July 2010

To my wife and our parents

ACKNOWLEDGEMENTS

I have been privileged to work with Alan Erera and Martin Savelsbergh. They guided and supported me every step of the way in making the research contained in this dissertation possible, and patiently allowed me to develop and mature as a researcher. I would especially like to thank Martin Savelsbergh for finding me in the early years of my graduate study.

I would like to thank Mark Ferguson, Ellis Johnson, and Joel Sokol, for agreeing to serve on my dissertation committee. Their constructive comments and inspiring suggestions are invaluable to the completion of this dissertation.

I am grateful to many fellow doctoral students at Georgia Tech. Of special note, I would like to thank Mike Hewitt for going above and beyond to help me with many problems.

Finally, I would like to thank my wife Jieyun Zhou and our parents. Their unwavering love and support is the reason I am where I am today. To them I dedicate this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A load plan specifies how freight is routed through a linehaul terminal network operated by a less-than-truckload (LTL) carrier. Determining the design of the load plan is critical to effective operations of such carriers. This dissertation makes contributions in modeling and algorithm design for three problems in LTL load plan design: refined execution cost estimation, dynamic load planning, and stochastic load plan design.

Chapter 2 focuses on accurate estimation of the operational execution costs of a load plan. Load plan design models in use or proposed today approximate transportation costs by using costs per trailer dispatched between terminals. Furthermore, empty transportation costs are determined by solving a trailer re-balancing problem. These approximations ignore two important ideas: (1) trailers are typically moved behind tractors in trains of two or three trailers, and the cost of moving a trailer train is not linear in the number of trailers; and (2) drivers must be scheduled for each dispatch, and driver rules introduce additional empty travel than that minimally required for trailer balance. We develop models that more accurately capture key operations of LTL carriers. A computational study demonstrates that our technology produces accurate operational execution costs estimates, typically within 2% of actual incurred costs.

Chapter 3 describes dynamic load planning (DLP) technology. Traditionally, load plans are revised infrequently by LTL carriers due to the difficulty of solving the associated optimization problem. Since freight volumes served vary each operating day, carriers typically operate by manually adjusting the plan at each terminal to each

day's operating conditions. Technological advances have now enabled carriers to consider more thorough, system-wide daily load plan updates. We develop technologies that efficiently and effectively adjust a nominal load plan for a given day based on the actual freight to be served by the carrier. We present two approaches for adjusting an existing load plan: an integer programming based local search procedure, and a greedy randomized adaptive search heuristic. A computational study using complete network data from a national carrier demonstrates that the proposed technology can produce significant cost savings.

Chapter 4 studies the stochastic load plan design problem. Load plan design models commonly represent origin-destination freight volumes using average demands derived from historical data, the drawback of which is that they do not describe freight volume fluctuations. We investigate load plan design models that explicitly utilize information on freight volume uncertainty during planning, and design load plans that most cost-effectively deal with varying freight volumes and lead to the lowest expected cost. We present Sample Average Approximation (SAA) approaches for solving stochastic integer programming formulations of the load plan design problem with demand uncertainty. In addition to applying the standard SAA approach, we also propose a modified version which, in order to correct the bias in the branch-and-bound search that results from using a sample, frequently computes an exact evaluation of the solution expected cost and a lower bound on this cost, to more accurately guide the search process.

CHAPTER I

INTRODUCTION

The trucking industry provides an essential service to the U.S. economy by transporting goods from business to business and from business to consumer. Less-than-truckload (LTL) transportation is an important segment serving businesses that ship quantities ranging from 150 lbs to 10,000 lbs, *i.e.*, less-than-truckload quantities. A typical shipment occupies only 5-10% of the trailer capacity. Hence, transporting each customer shipment directly from origin to destination is not economically viable. LTL carriers therefore collect and consolidate freight from multiple shippers, and route shipments through a terminal network of cross-dock transfer points to increase trailer utilization. A so-called *load plan* specifies how shipments traveling from each origin to destination are routed through the terminal network, and where along the way they are transferred from one terminal to another. Effective load plans are designed to minimize total linehaul transportation and handling costs, while satisfying origin-to-destination maximum transit time requirements for customers.

1.1 Less-Than-Truckload Freight Transportation

LTL linehaul networks are comprised of two types of terminals: *end-of-line* terminals that serve only as origin or destination terminals, and *breakbulk* terminals that additionally serve as transfer points for shipments. City operations are used at every terminal to organize the pickup and delivery of freight to customers with the small geographic area served by the terminal.

LTL networks enable consolidation of freight from many customers and take advantage of transportation economies of scale. During the day, city operations tours are used to both deliver shipments to customers and to collect freight before returning to the terminal. Shipments arriving at the terminal are then sorted and loaded into outbound trailers. When terminals do not collect enough arriving freight to build a nearly-full trailer direct to a destination terminal, some shipments may be loaded first on a trailer bound to an intermediate breakbulk terminal. Upon arrival at a breakbulk terminal, shipments are mixed with other arrivals (including those from local city operations), and the sorting and loading process continues until freight reaches its ultimate terminal destination.

Terminals in LTL networks are organized as cross-docks, and are set up to enable efficient transfer of freight from one trailer to another. Cross-docking a shipment, however, does require some time and handling cost. When designing a load plan, it is important to consider both the time and cost of transporting shipments, as well as the additional handling time and cost introduced by routing shipments through intermediate breakbulk terminals. A typical LTL shipment may travel from an origin terminal to a destination terminal, and pass through usually one or two intermediate breakbulk terminals en route.

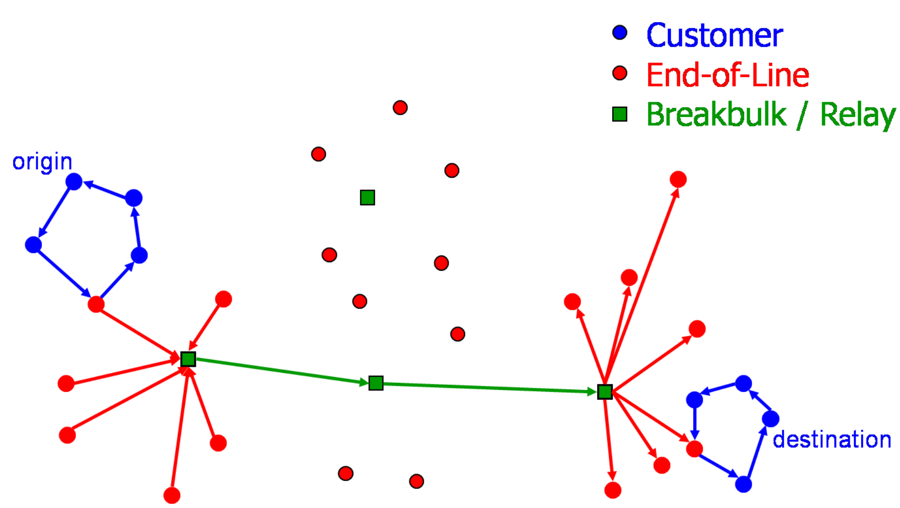

Figure 1: An LTL Network

An originating shipment is typically delivered by the city operation to the origin terminal by the late afternoon, and must be transported to the destination terminal by early morning on the day of delivery specified by the service standard. For an example drawn from a national LTL carrier, a shipment originating in Atlanta, GA on Monday with a destination of Cincinnati, OH and a service standard of one business day may arrive at the Atlanta terminal on Monday by 6 p.m. and must be moved to the Cincinnati terminal by no later than 8 a.m. Tuesday morning.

An important concept in load planning is that of a *direct*. A direct is a trailer movement from one terminal to another, where the trailer is loaded at the origin terminal and unloaded at the destination terminal with no intermediate loading or unloading. Each direct, therefore, specifies where freight handling occurs in the network. During load planning, shipments are planned to be loaded onto a sequence of directs that connect the origin terminal of the shipment to its ultimate destination.

A direct consists of either a single dispatch or a sequence of dispatches along the *legs* of the *trailer path* associated with the direct. In case a trailer path consists of multiple legs, the freight is *relayed* at the intermediate terminals. Relaying is necessary because of the limitations imposed on drivers by the Department of Transportation. For safety reasons, a driver is not allowed to drive for more than 11 hours or be on duty for more than 14 hours before requiring a rest period of at least 10 hours. Therefore, when the travel time between the origin and destination of a direct is more than a single driver can cover without a rest period, one or more relays are introduced. Usually, a relay happens at a breakbulk terminal, although they may happen at special relay facilities. At a relay point, the load is transferred to another driver and continues with minimal delay. For example, a direct from Dallas to San Francisco covers 27 hours of drive time and involves two relays and three drivers. It happens frequently that different directs include common legs in their trailer paths. For example, both the Dallas-San Francisco and the Dallas-El Paso directs include the Dallas-El Paso leg in their respective trailer paths.

LTL carriers pack freight into 28-foot trailers known as *pups* or 53-foot *vans*.

Typically, one tractor pulls either a single van or two pups. Most carriers use pups in their linehaul operations because a pup fills up more quickly than a van, and by combining pups with different final destinations it is possible to build loads that can be dispatched earlier, and thus improve service. For example, a pup on the Dallas-San Francisco direct can be paired up with a pup on the Dallas-El Paso direct to form a dispatch on the Dallas-El Paso leg. Effectively exploiting the advantages of using pups requires proper pup matching at breakbulks and relays, *i.e.*, deciding which pups to pair up into loads. Note that empty and loaded pups can also be combined into loads.

Driver management is a complex task for LTL carriers, since numerous rules govern how drivers can be used and are compensated (*e.g.*, a driver is compensated for a long rest away from his domicile to cover meals and accommodation). Furthermore, carriers are concerned about the quality of life of their drivers and want them to rest at their domiciles with some frequency, *e.g.*, at least every other night. In fact, LTL carriers often execute empty movements in order to return drivers to their domiciles.

1.2 Load Plan Design

Consider a path from a shipment's origin to destination consisting of a sequence of directs. A complete load plan will specify such a path for each shipment, and thus prescribes how all freight should be routed through the linehaul network. A traditional load plan also has additional structure, where the set of all paths terminating at a specific destination terminal d form a directed in-tree on the network of potential directs (see Figure 2 for an illustration). Thus, all shipments that pass through intermediate terminal i on a path to d are loaded onto the outbound direct (i, j). This simplifies terminal operations since a dock worker only has to examine the destination of a shipment to determine the appropriate outbound trailer for loading. For example, the load plan may give the following instruction: "all freight in Jackson, TN destined for Atlanta, GA loads next to Nashville, TN."

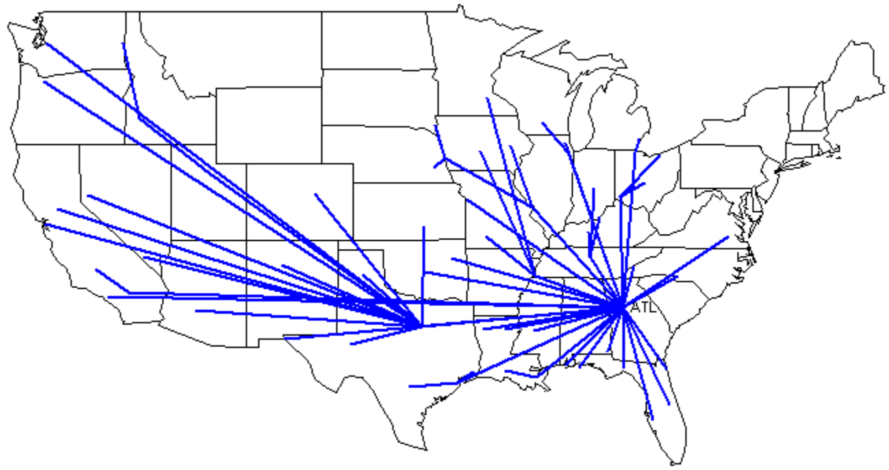

Figure 2: Freight Paths Form an In-Tree into a Destination Terminal

The load plan determines how freight is routed through a carrier's network, and thus where opportunities for consolidation occur. Consequently, load plan design is critical to effective operations of an LTL carrier. Load plans are designed to minimize total linehaul costs which are comprised of:

1. transportation costs associated with moving loaded and empty trailers; and

2. handling costs associated with transferring freight between trailers at a terminal.

Consider the network presented in Figure 3 for an example of load plan design. Above each arc, the number inside the parentheses represents the freight volume measured in fractional trailerloads, and the number outside the parentheses represents the corresponding required number of trailers if all origin-destination freight were to be sent direct. Low load factors are observed on many long directs. On the other hand, Figure 4 shows the freight routing under a load plan and the resulting required number of trailers on each direct. By consolidating and routing freight in a terminal network, we reduce the total trailer miles.

In the U.S., national carriers may spend millions of dollars weekly on transportation and handling costs. Thus, small percentage gains in trailer utilization can lead

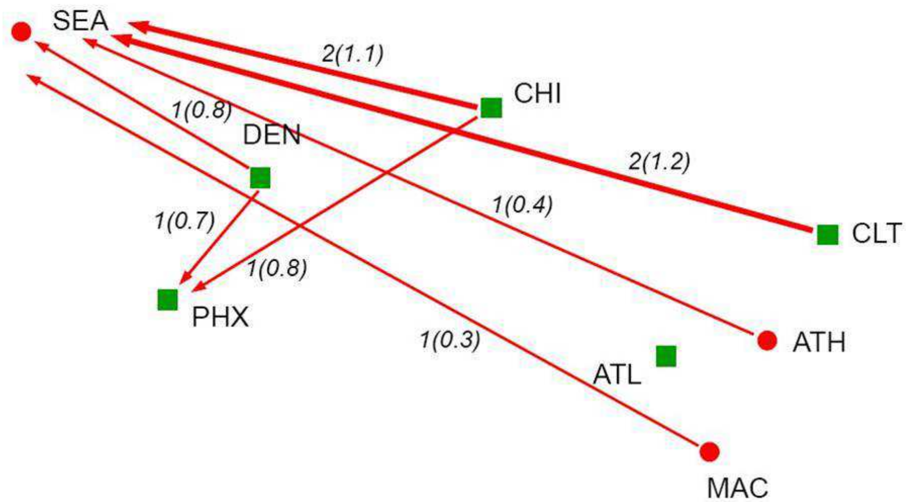

Figure 3: Load Plan Design Example - Freight Volumes

to significant monetary savings. Freight fluctuations, whether seasonal or caused by changing economic conditions, force LTL carriers to regularly review and adjust their load plan.

LTL shipments are quoted a service standard from origin to destination in business days. Historically these standards were long enough (often 5 or more business days) that service only loosely constrained freight routing decisions. Today, service standards of 1, and 2 days are much more common. Figure 5 presents a freight profile by service standard for a national carrier. These tighter standards must be enforced when planning shipment paths. Shorter service standards reduce opportunities for consolidation (since consolidation introduces handling time and circuity time penalties). As a result, carriers need methods for designing load plans that accurately model how short service standards constrain shipment paths and the consolidation opportunities that still exist.

Load plan design should also account for the trailer resource requirements that result from the plan. LTL carriers typically serve an overall freight profile that contains some geographic imbalance, *e.g.*, there is more freight flowing into Florida, than flowing out of Florida. Thus, trailers need to be moved empty from freight "sinks" to

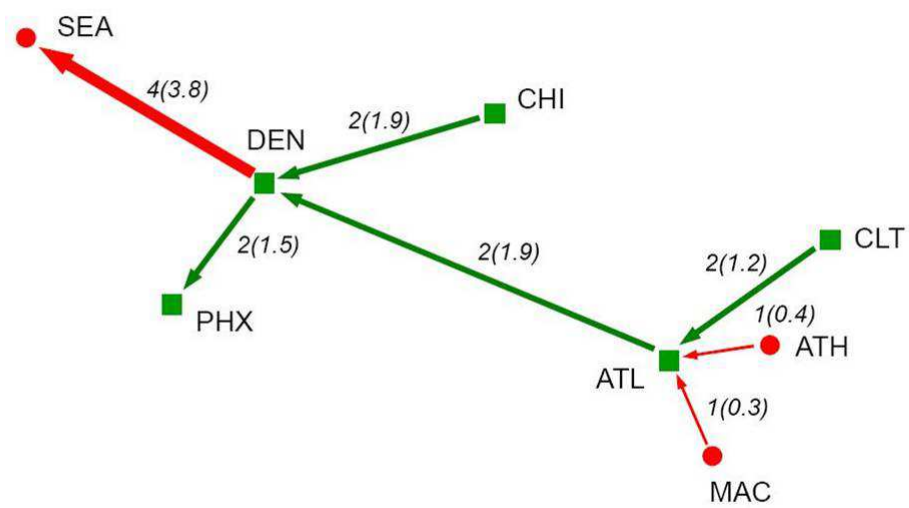

Figure 4: Load Plan Design Example - Consolidated Movements

ensure the availability at freight "sources". Empty trailer repositioning movements result in costs. Good load plans take advantage of trailer capacity that naturally arises in backhaul lanes to reduce total system costs.

1.3 Related Literature

Early research in LTL load plan design focused on models developed using static networks that do not explicitly capture service standard constraints or the timing of consolidation opportunities. A local improvement heuristic for such a model is presented in [16]; related work includes [18], [19], and [17]. Recognizing the limitations of the static network models, a dynamic model that can more accurately model consolidation timing is presented in [20]. The paper presents an alternative heuristic that relies on determining service network arc subgradients by solving large-scale multicommodity network flow problems. This approach, however, allows origin-destination shipments to split onto multiple paths and does not model empty equipment balancing decisions.

More recent research attempts to build and solve models that more accurately capture LTL linehaul costs. A column generation approach to create load plans where

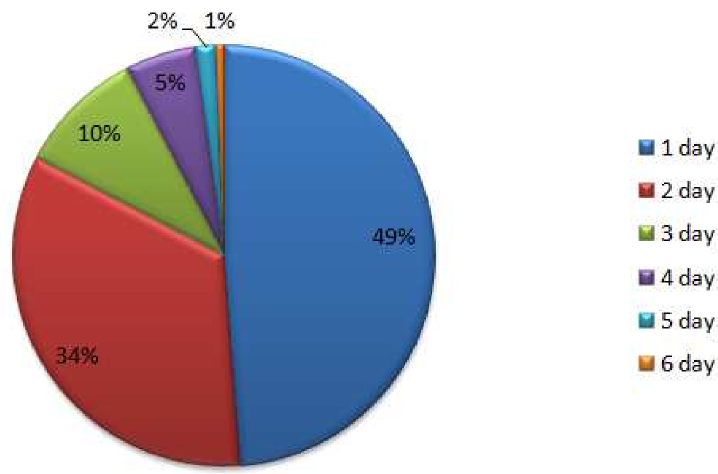

Figure 5: Freight Profile by Service Standard

columns represent freight path trees into a destination is developed [9]. A slope scaling heuristic is used to linearize costs when generating columns. The approach explicitly models service requirements of shipments, and only allows service-feasible paths to be selected. However, freight flows are mapped to a simplified time-space network with only one copy of an arc for each direct movement per day; this approximation may overestimate opportunities for consolidation cost savings. Most recently, [5] develop a model that uses a detailed time-space network representation to accurately model the timing of freight consolidation opportunities, and considers decisions for loaded and empty trailer movements simultaneously. The paper proposes a local search solution heuristic that searches a large neighborhood each iteration using an integer program.

Load plan design can be seen as a special case of service network design; this problem class has also received a great deal of attention (see [3] or [24] for a review of research in this area). The need to consider equipment management decisions in service network design problems is recognized in [15], which presents both a model and a metaheuristic for the problem. However, the instance sizes considered are significantly smaller than those typical for load planning for a large LTL carrier, and it is not clear how effective the proposed solution approach would be if adapted to the load plan design problem.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows:

Chapter 2 focuses on accurate estimation of the operational execution costs of a load plan. Load plan design models in use or proposed today approximate transportation costs by using costs per trailer dispatched between terminals. Furthermore, empty transportation costs are determined by solving a trailer re-balancing problem. These approximations ignore two important ideas: (1) trailers are typically moved behind tractors in trains of two or three trailers, and the cost of moving a trailer train is not linear in the number of trailers; and (2) drivers must be scheduled for each dispatch, and driver rules introduce additional empty travel than that minimally required for trailer balance. We develop models that more accurately capture key operations of LTL carriers. A computational study demonstrates that our technology produces accurate operational execution costs estimates, typically within 2% of actual incurred costs.

Chapter 3 describes dynamic load planning (DLP) technology. Traditionally, load plans are revised infrequently by LTL carriers due to the difficulty of solving the associated optimization problem. Since freight volumes served vary each operating day, carriers typically operate by manually adjusting the plan at each terminal to each day's operating conditions. Technological advances have now enabled carriers to consider more thorough, system-wide daily load plan updates. We develop technologies that efficiently and effectively adjust a nominal load plan for a given day based on the actual freight to be served by the carrier. We present two approaches for adjusting an existing load plan: an integer programming based local search procedure, and a greedy randomized adaptive search heuristic. A computational study using complete network data from a national carrier demonstrates that the proposed technology can produce significant cost savings.

Chapter 4 studies the stochastic load plan design problem. Load plan design models commonly represent origin-destination freight volumes using average demands derived from historical data, the drawback of which is that they do not describe freight volume fluctuations. We investigate load plan design models that explicitly utilize information on freight volume uncertainty during planning, and design load plans that most cost-effectively deal with varying freight volumes and lead to the lowest expected cost. We present Sample Average Approximation (SAA) approaches for solving stochastic integer programming formulations of the load plan design problem with demand uncertainty. In addition to applying the standard SAA approach, we also propose a modified version which, in order to correct the bias in the branch-and-bound search that results from using a sample, frequently computes an exact evaluation of the solution expected cost and a lower bound on this cost, to more accurately guide the search process.

Finally, Chapter 5 provides some concluding remarks and discusses possible improvements for future research.

CHAPTER II

REFINED EXECUTION COST ESTIMATION

2.1 Introduction

Our focus in this chapter is not on load plan design, but instead on accurately estimating the operational execution costs of a given load plan. During load plan design, transportation costs are usually approximated using linear cost factors per trailer dispatched between terminal pairs; often this cost is determined by multiplying a cost per mile by the mileage separating the terminals. However, this can be a crude approximation, since actual transportation costs are affected by the dispatched driver tours, and driver tours are severely restricted by government regulations and company and/or union policies. These policies and regulations can impact the amount of empty travel required, and may lead to more empty travel than predicted by empty trailer balancing models. Furthermore, short trailers (often referred to as *pups*) can be moved by a single driver in trains of two or three trailers; in this research, we assume that a trailer train contains at most two trailers. Since it is difficult to predict in advance what fraction of trailers dispatched on a lane between two terminals will travel alone or in a train, it is not easy to determine an appropriate linear cost per trailer. As a result, load plan design methods may substantially under- or over-estimate transportation costs. Such cost estimation errors may have unintended and costly consequences.

The technology we develop and present in this chapter takes a set of shipments for a certain planning horizon and a load plan to route shipments through the terminal network, and then builds driver dispatches with associated dispatch windows (a dispatch corresponds to a combination of up to two trailers and each trailer contains one or more shipments) and generates cost-effective driver tours to cover these dispatches and balance empty trailers. The cost of executing these driver tours is then our estimate of the transportation costs incurred when executing the load plan.

Having an accurate estimate of the cost of executing a new load plan is an essential part of the load plan design process. An ancillary benefit of our approach is that it builds a set of cost-effective driver tours. First, this set of tours may be useful in practice. Second, the set also provides input useful for determining the number of drivers needed at the different terminals. Identifying a set of suggested driver tours for adjusted load plans is especially important to speed up implementation, and thus the realization of any cost savings resulting from the use of the adjusted load plan.

We have conducted a computational study of the proposed approach using an actual load plan and actual shipment data from a super-regional LTL carrier operating in the continental U.S. We compare the execution cost estimates of the load plan from two different approaches with the actual linehaul costs incurred in practice when executing the load plan. The first estimate comes from SuperSpin, the current industry standard software for load plan design. The second estimate is taken from the technology presented in this chapter. The results show that SuperSpin tends to underestimate actual costs, between 88.8% to 90.6% of actual, while our technology provides accurate cost estimates, between 99.6% and 101.7% of actual.

Summarizing, this research makes contributions primarily in the context of load plan design, evaluation, and execution for LTL carriers. Specifically, we have developed technology that

- improves load plan execution cost estimates; accuracy improvements on the order of 10-15% are shown for a super-regional carrier,
- builds a set of dispatches and generates a set of cost-effective driver duties and tours covering these dispatches; driver duties and tours satisfy government

regulations and union and/or company rules and can thus be used in practice, and

- solves real-life instances efficiently; less than 2 hours for instances representing a week of data with over 140,000 shipments, which equates to over 10,000 loads and approximately 6,000 driver duties.

The remainder of this chapter is organized as follows. In Section 2.2, we review relevant literature. In Section 2.3, we formally state the load plan costing optimization problem and discuss the modeling issues and choices. In Section 2.4, we introduce our solution approach. In Section 2.5 we present the results of an extensive computational study using historical data from a super-regional LTL carrier in the U.S.

2.2 *Additional Related Literature*

Problems also related to the one we consider in this chapter are the focus of [2] and [21], in which solution approaches are developed that integrate empty balancing with a pup matching and routing for small package express carrier operations. The proposed set partitioning model uses composite variables that define complete paths for one or more trailers, and employs templates to limit the set of such composites generated. While we also consider a pup loading and matching problem, our matching problem is somewhat simpler since we assume that the best trailer path is known for each direct. Furthermore, we estimate transportation costs more precisely since we construct feasible driver tours to cover loads. Greedy approaches are developed in [6] to construct driver tours that cover dispatches; in this research, we develop an optimization-based set covering heuristic.

2.3 *Model Formulation*

As mentioned in the introduction, our focus is not load plan design, but accurately estimating the operational execution costs of a given load plan. A number of modeling choices were made when formulating the problem. These choices are discussed below.

1. *The problem is formulated on a time-space network.* Flat network representations, *i.e.*, networks without an explicit time dimension, such as the ones used in [18], [16], [19], and [17], are based on two important assumptions: (1) the total trailer loads needed on a direct during the planning horizon can be determined by assuming that all freight traveling at any time within the planning horizon can be consolidated; and (2) service standard constraints can be modeled by using a proxy, *e.g.*, by ensuring a minimum trailer frequency on a direct per day. In today's LTL market where 1-day and 2-day service have become the norm, these assumptions are no longer valid. It is necessary to use a representation that can explicitly represent time. A detailed time-space network model allows consolidation timing and service standards to be modeled accurately. Given a time discretization of the planning horizon, multiple nodes are created for each terminal, one for each time point, so that each node represents a location and a point in time. For each leg in the linehaul network, we create multiple transportation arcs in the time-space network, each representing the possibility to move freight at a particular time. Each node in the time-space network is connected with an arc to the node representing the same terminal at the next time point, thus modeling the possibility to hold freight at a terminal. See Figure 6 for an illustration.

2. *The planning horizon considered is a week.* The freight volumes within a week often exhibit marked variability by day-of-week, but freight patterns tend to be similar across weeks. As a result, carriers have started to explore day-differentiated load plans, *i.e.*, load plans that allow for different freight routing decisions on different days of the week.

   Carriers often out-source a portion of their transportation needs to third-party

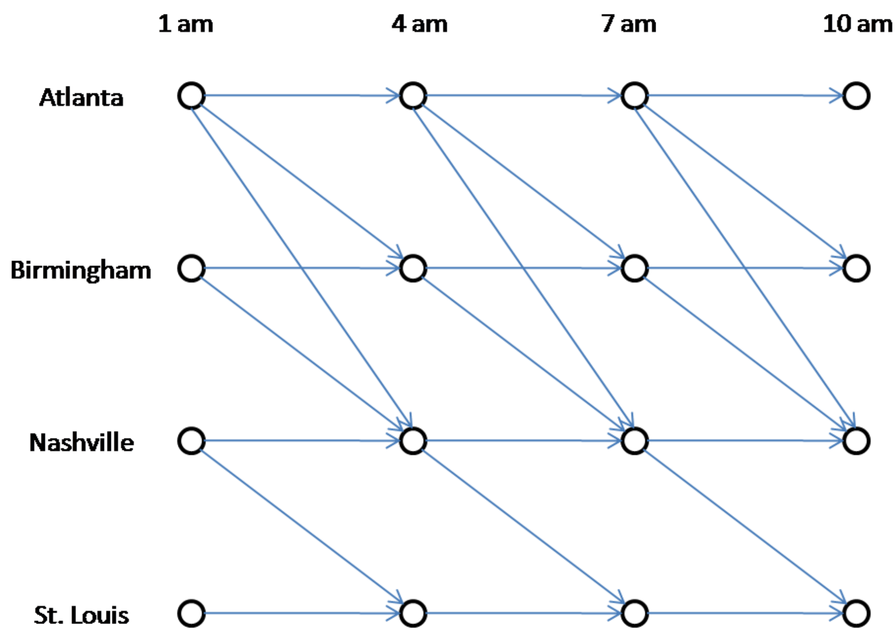

Figure 6: Time-Space Network

carriers, a practice referred to for the remainder of this dissertation as *purchased transportation*. Usually, the third-party carriers are railroads, but occasionally also trucking companies are used. Transporting freight by rail is cheaper, but slower than by truck. Since weekend days do not count against service, carriers often utilize rail transportation over the weekend. In fact, most rail options are only available near the end of the week. Purchased transportation schedules tend to repeat weekly.

The above discussion suggests that a week-long planning horizon is appropriate. To accurately capture daily freight volume fluctuations, we model freight originating at a terminal on a given day and destined for another terminal on another day as a commodity. Arcs representing purchased transportation options are only created at their scheduled time of the week.

3. *Time is discretized in hours.* Time must be modeled at a fine level of granularity for two reasons: (1) to be able to accurately model the driver rules discussed in Section 1.1, and (2) to be able to properly model freight paths between

origin-destination pairs with tight service standards. Consider the freight path encountered at a super-regional carrier for freight available in Lexington, KY at 7 pm and due in Grayling, MI at 8 am the next day shown in the top part of Table 1.

Table 1: Modeling Freight Paths

Travel from Lexington, KY to Cincinnati, OH for 2 hours
Spend 0.5 hour being handled at Cincinnati, OH
Travel from Cincinnati, OH to Toledo, OH for 3.75 hours
Spend 0.5 hour being handled at Toledo, OH
Travel from Toledo, OH to Grayling, MI for 4.46 hours
Leave Lexington, KY at 19:00, arrive at Cincinnati, OH at 21:00
Finish handling at Cincinnati, OH at 21:30
Leave Cincinnati, OH at 22:00, arrive at Toledo, OH at 01:45
Finish handling at Toledo, OH at 02:15
Leave Toledo, OH at 03:00, arrive at Grayling, MI at 07:27

An hourly time discretization, *i.e.*, constructing a node at every hour, allows us to accurately model this freight path by timing the dispatches as shown in the bottom part of Table 1.

4. *Freight enters the linehaul system at 7 p.m. and leaves the linehaul system at 8 a.m..* All freight picked up during the day is assumed to be ready to be send into the linehaul system at 7 p.m. local time. All freight to be delivered during the day must arrive at its destination terminal at 8 a.m. local time. Thus, we model the freight that enters the linehaul network at terminal t_1 on day d_1 and is due at terminal t_2 on day d_2 as originating in the time-space network at node $n_1 = (t_1, 7 \text{ p.m. } d_1)$ and is destined to node $n_2 = (t_2, 8 \text{ a.m. } d_2)$.

5. *Handling 1-day service freight takes 30 minutes and handling all other freight takes 2 hours.* A certain amount of time is required for handling freight at intermediate breakbulks. Special handling procedures are generally used at breakbulks to prioritize the processing of 1-day freight to ensure that it can

meet its service expectation. Therefore, a short handling time for 1-day freight is appropriate. We note that handling of freight can only occur during business hours, which typically start at 12 am on Monday morning, and end Saturday at noon. The terminals are, however, accessible to drivers arriving/departing all weekend long.

6. *Modeling full truckload freight.* To diversity their offerings, and because it is more profitable, many LTL carriers run a small full truckload operation as well. Full truckload shipments do not require any intermediate handling, but the trailers used for full truckload service are relayed along legs according to the load plan. LTL carriers frequently fill truckload trailers with LTL shipments to exploit any remaining trailer capacity. Therefore, full truckload freight should be considered when estimating the execution cost of a load plan.

We can now state the LTL load plan cost estimation problem as the problem of determining a freight path for each commodity in the time-space network, conforming to the load plan, and creating valid driver tours to cover the resulting dispatches with minimal total cost over the week. Since handling costs are fixed given a load plan, minimizing total cost is equivalent to minimizing the transportation cost required to move empty and loaded trailers. As we will describe in the following section, total transportation cost in this problem is assumed to be the sum of the costs of executing the set of driver tours necessary to move all empty and loaded trailers.

2.4 Solution Approach

We have designed and implemented a three-phase solution approach for the LTL load plan cost estimation problem.

2.4.1 Phase I: Loading and Matching Pups

In the first phase, we determine a timed path for each commodity, we build loaded trailers on each direct, and combine loaded trailers on a direct into dispatches, *i.e.*, the trailers are pup-matched. We have developed a GRASP heuristic to determine the timed paths. The GRASP heuristic sequentially chooses paths for commodities using a shortest path algorithm; note that since the load plan is fixed, determining the path for a commodity is simply selecting the dispatch times for each of the direct moves in the load plan path (represented by arcs in the time-space network). The sequential nature of the search enables us to estimate the marginal cost of adding the commodity under consideration to all possible dispatch arcs, and thus to minimize the marginal cost increase that results from selecting a set of feasible dispatch times. More formally, the marginal cost of adding a commodity of size c (measured in fractional trailerloads) to arc a as a leg of the trailer path of direct d_0 is defined as follows. Suppose that $d_0, \ldots, d_m$ are the directs whose trailer paths include arc a as a leg and that the dispatch cost on arc a is p_a. Furthermore, let w_0 be the existing freight (measured in fractional trailerloads) on arc a for direct d_0 and let e_i be the current number of trailers on leg a for direct d_i, $i = 0, \ldots, m$. Finally, let U be the capacity of a trailer. Adding commodity c to arc a changes the required number of trailers on leg a for direct d_0 from e_0 to $\lceil \frac{w_0+c}{U} \rceil$. Hence the marginal cost is

$$\left(\left\lceil \frac{\lceil \frac{w_0+c}{U} \rceil + \sum_{i=1}^{m} e_i}{2} \right\rceil - \left\lceil \frac{\sum_{i=0}^{m} e_i}{2} \right\rceil \right) p_a. \tag{1}$$

The sequence in which the commodities are processed impacts the paths chosen, hence we decided to implement a GRASP heuristic. Let a commodity's slack time be defined as the maximum length of time it can be held at its origin such that it can still be dispatched along a path that satisfies the service deadline. A large slack time is an indication of more flexibility for choosing dispatch times along the path, and hence more opportunities for taking advantage of available capacity on trailers along the way that have been "opened" for transporting other commodities. On the other hand, a small slack time for a commodity implies that there is little or no flexibility for choosing dispatch times. Therefore, for a given direct in its load plan path, unless there is a trailer with sufficient remaining capacity dispatched at the exact time required by this commodity, a new trailer must be opened to accommodate this commodity. Clearly, it is better to open such new trailers earlier in the heuristic to allow other commodities to fill in any remaining unused capacity therein. This suggests that we process commodities with small slack times first. Furthermore, a commodity with smaller size c is more likely to be able to take advantage of remaining capacity in open trailers; thus, we break slack time ties between commodities by choosing those with larger sizes first. The GRASP heuristic is described in Algorithm 1.

Algorithm 1 GRASP for Pup Loading and Matching

```
Sort the commodities in order of increasing slack time. In case of ties, sort the
commodities in order of decreasing weight.
for i = 1 to N do
  Create a copy of the commodity list
  while the list is not empty do
    Select a commodity from the list biased towards the top, i.e., the k-th com-
    modity c_k with probability λ · (1 − λ)^{k−1}, k = 1, 2, ...
    Find the least-marginal-cost path for c_k that conforms to the load plan, using
    the cost in (1)
    Remove c_k from the commodity list
  end while
  if an improved solution is found then
    Update the best solution
  end if
end for
```

Once timed paths for all the commodities have been selected by the GRASP heuristic, we have determined, for each arc a,

- a set of commodities that move on arc a; each commodity is associated with a direct whose trailer path includes a

- the number of required trailers for each such direct, by grouping commodities associated with common directs into trailers
- the number of dispatches required on arc a, by matching up trailers required for all directs whose trailer path includes a

At this point, pup-matched dispatches moving all the freight are constructed.

2.4.2 Phase II: Determining Dispatch Windows

Note that in Phase I, a timed path is selected for each commodity, and thus the trailers that have been implicitly constructed all have specific dispatch times. However, the shipments comprising a dispatch may not be tightly constrained by service, and thus may have flexibility in the selection of actual dispatch times. Such dispatch flexibility for loads will be beneficial when building driver tours. Therefore in Phase II, we use a linear program to determine dispatch windows for each load constructed in Phase I.

Let P be the set of dispatches (or loads) built in Phase I corresponding to purchased transportation. Since purchased transportation takes place on fixed schedules (recall that purchased transportation is typically provided by railroad companies) these dispatches cannot be altered. Let L be the set of dispatches (or loads) built in Phase I corresponding to transportation provided by company drivers. Our goal is to find for each load $i \in L$ an earliest and a latest dispatch time, denoted by α_i and β_i respectively, such that when all loads are dispatched between their earliest and latest dispatch time all freight is moved feasibly, i.e., every shipment reaches its destination at or before its due time and can make feasible connections at transfer and relay points. See Figure 7 for an illustration of a dispatch window.

We introduce some additional notation before discussing the linear programming model. Let o^c and d^c denote the ready time at the origin and the cut time at the destination for commodity c. Furthermore, let $p_1^c, \ldots, p_{n_c}^c$ denote the sequence of

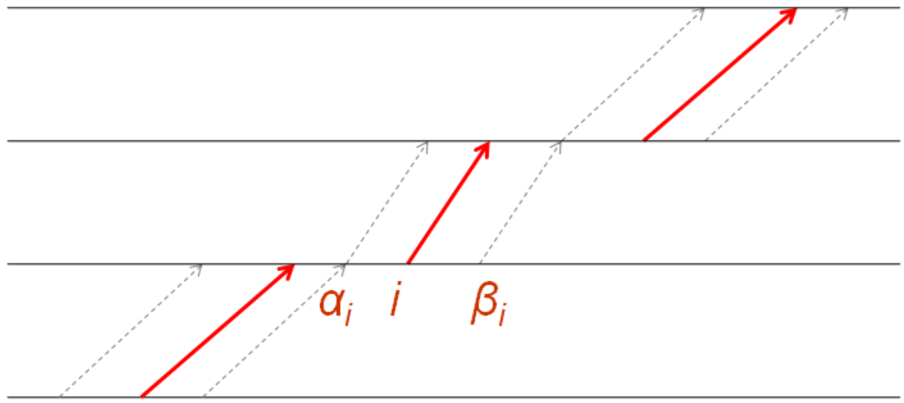

Figure 7: Dispatch Window

dispatches for commodity c. Finally, let $dt_{p_k^c}$, $tt_{p_k^c}$, $ht_{p_k^c}$, and $ft_{p_k^c}$, be the dispatch time, travel time, handling time, and finish time, respectively, of dispatch p_k^c, where the dispatch time and the finish time are based on the timed path selected for commodity c. Our goal is to determine the flexibility of load dispatch times, i.e., the load dispatch windows, and the objective function of the linear program should reflect this. We have chosen to maximize the sum of the widths of individual dispatch windows. An alternative is to maximize the minimum width of any dispatch window. However, since there typically are a few dispatches without any flexibility this objective does not produce any useful information.

The linear program to determine dispatch windows is as follows

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{i \in L} (\beta_i - \alpha_i) \\
\text{subject to} \quad & \alpha_{p_1^c} \geq o^c \quad \forall c,\ p_1^c \in L && (2) \\
& \beta_{p_{n_c}^c} + tt_{p_{n_c}^c} \leq d^c \quad \forall c,\ p_{n_c}^c \in L && (3) \\
& \beta_{p_l^c} + tt_{p_l^c} + ht_{p_l^c} \leq \alpha_{p_{l+1}^c} && (4) \\
& \qquad \forall c,\ 1 \leq l \leq n_c - 1,\ p_l^c \in L,\ p_{l+1}^c \in L \\
& \beta_{p_l^c} + tt_{p_l^c} + ht_{p_l^c} \leq dt_{p_{l+1}^c} && (5) \\
& \qquad \forall c,\ 1 \leq l \leq n_c - 1,\ p_l^c \in L,\ p_{l+1}^c \in P \\
& ft_{p_l^c} \leq \alpha_{p_{l+1}^c} && (6) \\
& \qquad \forall c,\ 1 \leq l \leq n_c - 1,\ p_l^c \in P,\ p_{l+1}^c \in L \\
& \alpha_{p_l^c} \leq dt_{p_l^c} \leq \beta_{p_l^c} \quad \forall c,\ 1 \leq l \leq n_c,\ p_l^c \in L && (7)
\end{aligned}
$$

Constraints (2) ensure that the first dispatch occurs no earlier than the ready time at the origin and constraints (3) ensure that last dispatch is such that the freight arrives at the destination before its cut time. Constraints (4), (5), and (6) ensure feasible connections at transfer and relay points. Constraints (7) forces the dispatch times on the timed path selected in Phase I to be feasible.

The linear program presented above ignores one important problem characteristic: terminals operate only limited hours over the weekend. Therefore, we may have produced dispatch windows that require handling to take place during weekend hours. A post-processing step is added to fix such situations. More specifically, the predecessor's latest dispatch time is pushed back or the successor's earliest dispatch time is pushed forward, whichever applicable, to their dispatch times on the timed path selected in Phase I to be feasible, since all connections are feasible with these dispatch times. The post-processing algorithm is given in Algorithm 2.

Algorithm 2 Post-Processing Dispatch Windows

for all commodity c **do**
  for $l = 1$ to $n_c - 1$ **do**
    if $p_l^c \in L$ **and** c requires a handling after traveling on p_l^c **and** the time period from $\beta_{p_l^c} + tt_{p_l^c}$ to $\beta_{p_l^c} + tt_{p_l^c} + ht_{p_l^c}$ does not fall entirely in the business hours **then**
      $\tau \leftarrow$ start of next business day $+ ht_{p_l^c}$
      if $p_{l+1}^c \in L$ **and** $\tau > \alpha_{p_{l+1}^c}$ **then**
        $\beta_{p_l^c} \leftarrow dt_{p_l^c}$
        $\alpha_{p_{l+1}^c} \leftarrow dt_{p_{l+1}^c}$
      else if $p_{l+1}^c \in P$ **and** $\tau > dt_{p_{l+1}^c}$ **then**
        $\beta_{p_l^c} \leftarrow dt_{p_l^c}$
      end if
    end if
  end for
end for

2.4.3 Phase III: Constructing Driver Tours

In the third phase, we determine driver tours to cover all dispatches that can be performed by a single driver. A driver tour begins and ends at a driver domicile, and thus forms a timed cycle, and consists of one or more duties. A duty is a feasible sequence of timed dispatches that can be performed in a single day and abides by Hours of Service regulations. If a tour contains multiple duties, the duties are separated by a rest period. The Hours of Service regulations impose the following restrictions on drivers: a driver is allowed to drive up to 11 hours in a duty, a duty must not exceed 14 hours, and a driver must rest for at least 10 hours between duties. Note that duties may include empty dispatches. If we assume that drivers are always dispatched with two trailers, empty trailer balance over time is implied.

LTL companies must compensate drivers for long rests spent away from their domiciles, referred to as lay-downs. Lay-down costs typically include hotel room stays and meals. Most companies like to have their drivers resting at their domiciles with some frequency. Single-man drivers typically do not rest away from their domicile two nights in a row. Therefore a tour consists of either one or two duties. If a tour contains two duties (the first ending away from the domicile, and the second returning to the domicile), the long rest that separates the two duties should not exceed 14 hours (and has to be at least 10 hours). A duty typically contains no more than four dispatches.

Since we want to build single-man driver tours, we modify the time-space network by removing all purchased transportation arcs, by removing all arcs representing travel of more than 11 hours, and by adding lay-down arcs from every node to the nodes representing the same terminal 10 to 14 hours into the future.

To determine a low-cost set of driver tours covering all dispatches that can be performed by a single driver, we use a set-covering model. Let $I \subseteq L$ be the subset of loads that can be performed by single driver, *i.e.*, loads that do not require more than 11 hours of driving. For each $i \in I$, let $A(i)$ be the subset of arcs in the time-space network associated with load i that fall within its dispatch window. Since $A(i)$ and $A(i')$ with $i \neq i'$ may contain common arcs, an arc does not uniquely identify a load. For each arc a, let $I(a) = \{i \in I \mid a \in A(i)\}$ be the set of loads that can potentially use a.

In the set covering model the goal is to select a subset of tours covering all the dispatches at minimum cost. Let T be the set of tours, c_t be the cost of executing tour $t \in T$, a_{it} be the number of times tour $t \in T$ covers load i, and z_i be the number of dispatches required for load i. If x_t represents the number of times tour $t \in T$ is executed, then we have the following integer programming formulation:

$$\begin{aligned} \text{Minimize} \quad & \sum_{t \in T} c_t x_t && (8) \\ \text{subject to} \quad & \sum_{t \in T} a_{it} x_t \geq z_i \quad \forall i \in I \\ & x_t \in \mathbb{Z}^+ \quad \forall t \in T \end{aligned}$$

As the set of tours is too large to consider explicitly we rely on column generation to solve the linear programming relaxation. Given a dual solution π to the linear programming relaxation of a restricted master problem, the pricing problem seeks to identify a tour with negative reduced costs. More specifically, the pricing problem seeks a tour minimizing $\sum_{a\in t}(c_a - \max_{i\in I(a)} \pi_i)$. Note that because multiple loads may be covered with the same arc, we have to look at $\max_{i\in I(a)} \pi_i$ when determining the dual value to use on an arc. The pricing problem is thus a resource-constrained shortest path problem with arc cost $c_a - \max_{i\in I(a)} \pi_i$.

We keep track of four resources to ensure the feasibility of the tour found: the duty time, the driving time in a duty, the number of dispatches in a duty, and the number of lay-downs in a duty. Let d_a be the driving time on arc a and H be the lay-down cost. Then the resource extension functions for the various arc types and the resource limits are summarized in Table 2.

Table 2: Resource Extension Functions and Resource Limits

	Initial value	Resource extension functions			Resource limits
		Transportation arc a	Waiting arc	Lay-down arc	at a node
Duty time	0	$+\lceil d_a \rceil$	$+1$	reset to 0	[0,14]
Driving time in duty	0	$+d_a$	unchanged	reset to 0	[0,11]
Num of dispatches in duty	0	$+1$	unchanged	reset to 0	[0,4]
Num of lay-downs in duty	0	unchanged	unchanged	$+1$	[0,1]
Cost	0	$+(c_a - \max\limits_{i\in A(a)} \pi_i)$	unchanged	$+H$	

We round up the driving time when updating the duty time because freight consolidation only takes place at the discretization points, and thus dispatches only occur at these discretization points, *i.e.*, whole hours. Note that the duty time label is only used in the dynamic programming algorithm for solving the resource-constrained shortest path problem. When reporting duty times in our computational study, we calculate and report actual duty times.

We solve the resource-constrained shortest path problem using a typical dynamic programming approach (see [8] and [4] for discussions of dynamic programming approaches for constrained shortest path problems). In the path extension step, a waiting arc and a lay-down arc are disallowed to immediately follow each other in order to prevent undesirable long rests.

The following ideas were incorporated to accelerate the column generation process:

- We do not solve the pricing problem completely, but terminate the search as soon as a feasible tour with a negative reduced cost is found, and then add the newly found column to the restricted master problem, which is then re-solved.

- We restricted the search to tours that start with a loaded dispatch. This does not preclude good solutions, but speeds up the search considerably. Furthermore, we sort the loads in order of increasing cost $c_i - \pi_i$ and select loads in that order to start a tour.

- Because only one column is added in each iteration, many dual prices will not change between successive pricing problem solves. It is thus reasonable to assume that a load that failed to produce a tour with a negative reduced cost will likely continue to do so in the near future. Hence we exclude it from consideration for a number of iterations.

The algorithm to find a low-cost set of driver tours covering all dispatches that can be performed by a single driver is described in Algorithm 3.

2.4.3.1 Meet-and-Turns and Initial Columns

It is well-known that a good set of initial columns can reduce the running time of a column generation algorithm. However, before presenting our approach for creating initial columns, we have to discuss *meet-and-turns*, which are used by LTL carriers on long legs to reduce lay-down costs. A meet-and-turn is considered when two drivers move loads in opposite directions on a leg that is longer than half of the maximum allowed driving time in a duty, *i.e.*, 5.5 hours. Without intervention, the drivers moving these loads will be unable to return to their domiciles at the end of the day because they would violate the driving time limit. A meet-and-turn, illustrated in Figure 8, instead has the two drivers meet at a location along the leg, exchange their loads, and then return to their respective starting location. This ensures that both

Algorithm 3 Load Covering

Generate a set of initial columns
repeat
  Solve the linear programming relaxation of the restricted master problem
  Retrieve the dual prices
  Sort the loads in increasing order of $(c_i - \pi_i)$
  for $i = 1$ to $|I|$ **do**
    if load i is not excluded from consideration **then**
      Invoke a dynamic programming search for a tour that starts with load i
      while a tour with a negative cost is not found **and** there are feasible extensions **do**
        Perform a dominance check and a path extension
      end while
      if a desirable tour is found **then**
        Add a column representing the tour
        break loop
      else
        Exclude load i from consideration for M iterations
      end if
    end if
  end for
until a column with negative reduced cost is not found
Solve a set covering problem over the columns in the restricted master problem

loads arrive at their destination on time, and both drivers get back to their domiciles at the end of the day. A parking lot or a rest area suffices as a meet-and-turn location. Executing meet-and-turns reduces lay-down expenses for the carriers and improves the quality of life for the drivers.

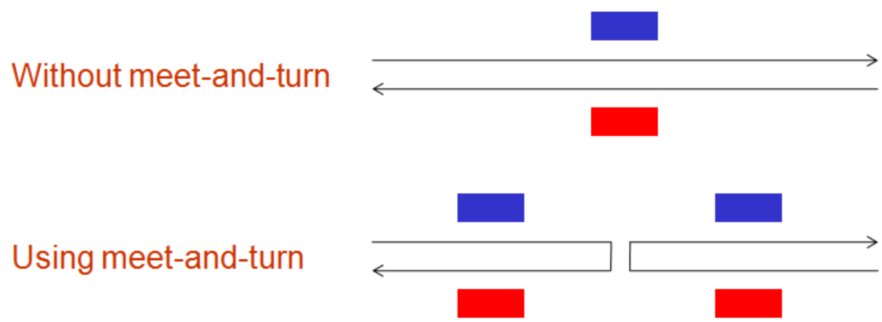

Figure 8: Meet-and-Turn

To generate a set of initial columns, *i.e.*, driver tours, we use templates of desirable driver tours. Some of these templates involve meet-and-turns. The algorithm for creating initial columns is described in Algorithm 4.

2.4.3.2 Short Driving and Duty Times

Hours of Service regulations, which are motivated by safety considerations, only restrict the maximum driving and duty times. Therefore, short driver duties with short driving times are legal, but may not be cost-effective. When faced with short driver duties, non-unionized carriers often resort to "dual-using" drivers by having them perform dock work, and by staffing dock workers accordingly. Short driver duties are thus acceptable, but typically undesirable.

In this section, we propose a penalty-based approach that allows the analysis of the tradeoff between the quality of the tours (in terms of duty time and driving time) and the execution costs. The penalty-based approach penalizes short duties with a term in the objective function that is proportional to the difference between the actual and the maximum allowed driving time in a duty, *i.e.*, 11 hours. The reason that we

Algorithm 4 Creation of Initial Columns

Let SL contain loads that require less than or equal to 5.5 hours of driving and let LL contain loads that require more than 5.5 hours of driving
for all $i, j \in SL,\ i \neq j$ **do**
  if i and j can form a feasible *out-and-back* tour without a lay-down **then**
    Create a column representing the tour
  end if
end for
for all $i, j \in LL,\ i \neq j$ **do**
  if i and j can form a feasible meet-and-turn **then**
    Create a column representing both tours in the meet-and-turn
  end if
end for
for all $i, j \in LL,\ i \neq j$ **do**
  if i and j can form a feasible out-and-back tour with a lay-down **then**
    Create a column representing the tour
  end if
end for
for all $i \in I = SL$ such that i has not been covered by any tour **do**
  Create a column representing an out-and-back tour with an outbound dispatch moving i and empty inbound dispatch
end for
for all $i \in I = LL$ such that i has not been covered by any tour **do**
  Create a column representing a meet-and-turn consisting of i and an empty dispatch in the opposite direction
end for

penalize short driving times rather than short duty times is because waiting between dispatches is counted towards duty time, and waiting should not be encouraged for its own sake.

Let n_t be the number of duties in tour t (either 1 or 2), let α be a parameter indicating the weight we assign to the penalty term, and let LD be the set of lay-down arcs. Then the cost of a tour t becomes

$$\begin{aligned}
& \sum_{a \in t} \Big(c_a - \max_{i \in A(a)} \pi_i \Big) + \alpha \Big(11 \cdot n_t - \sum_{a \in t} d_a \Big) \\
= \;& \alpha \cdot 11 \cdot \Big(1 + \sum_{a \in t \cap LD} 1 \Big) + \sum_{a \in t} \Big(c_a - \max_{i \in A(a)} \pi_i - \alpha \cdot d_a \Big) \\
= \;& \alpha \cdot 11 + \sum_{a \in t} \Big(c_a - \max_{i \in A(a)} \pi_i - \alpha \cdot d_a + \mathbf{1}_{LD}(a) \cdot \alpha \cdot 11 \Big)
\end{aligned}$$

which is still additive on arcs.

Therefore, the same solution methodology can be applied with only minor modifications. All that is required is to adapt a few elements in the last row of Table 2 as shown in Table 3.

Table 3: Cost Extension with Penalty

	Initial value	Transportation arc a	Waiting arc	Lay-down arc	
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Cost	$\alpha \cdot 11$	$+ \, (c_a - \max_{i \in A(a)} \pi_i) - \alpha \cdot d_a$	unchanged	$+ \, H + \alpha \cdot 11$	

2.5 Computational Study

We next present the results of a set of computational experiments conducted to tune and analyze the performance of our proposed LTL load plan cost estimation technology. We use four instances, each representing an actual week of shipment data of a super-regional LTL carrier in the U.S. The carrier's linehaul network consists of 253 terminals (end-of-lines, breakbulks, and relays) and 8,152 linehaul legs, and the carrier transports over 140,000 shipments every week. Each week begins on a Sunday at 12:00 a.m., and concludes on a Saturday at 11:59 p.m. Table 4 gives the start and end dates of the weeks used in our computational experiments.

Table 4: Weeks Used in Our Computational Study

Instance	Start date	End date
$W1$	March 01, 2009	March 07, 2009
$W2$	March 08, 2009	March 14, 2009
$W3$	March 15, 2009	March 21, 2009
$W4$	March 22, 2009	March 28, 2009

All computational experiments were carried out on a system with a 2.66 GHz Intel Xeon processor and 4 GB of RAM, and using CPLEX 11.1 as the optimization engine.

2.5.1 GRASP Heuristic Parameters

The first experiment is designed to determine the parameters λ and N for the GRASP heuristic for loading and matching pups. For $\lambda = 0.1, 0.2, ..., 0.9, 1.0$, we let Algorithm 1 run for 100 iterations and monitor the progress of the value of the best solution found. Note that for $\lambda = 1$ the algorithm reduces to a greedy heuristic, and the behavior of the algorithm is deterministic, so there is no benefit of performing more than one iteration. Each iteration takes approximately 4 minutes to run. Figure 9 and 10 show the progress over time for weeks $W1$ and $W4$ (similar behavior was observed for weeks $W2$ and $W3$).

The results indicate that although a greater level of randomization, i.e., a smaller value of λ, tends to lead to a slightly better solution over time, the benefit is minimal as the difference between the overall best solution value and the one found by the greedy heuristic is less than 0.40%. Hence, for the rest of the computational experiments we will use the greedy heuristic.

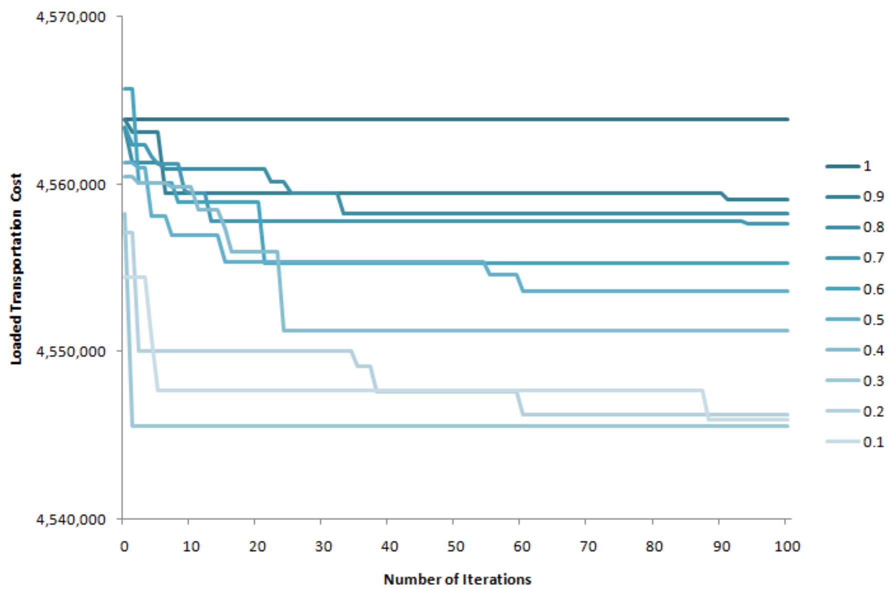

Figure 9: Progress of GRASP Heuristic for Instance $W1$

2.5.2 Dispatches and Dispatch Windows

Next, we present a few statistics related to the dispatches built in Phase I and the dispatch windows determined in Phase II. (Note that the dispatch windows are determined using a linear program and thus are computed in a matter of seconds.) In Figure 11, we show the number of dispatches occurring at particular times during the day as determined by Phase I.

We see that most dispatches occur between 7 p.m. and 6 a.m. This is not unexpected, and in line with what happens in practice, as a significant portion of shipments have 1-day service guarantees, which implies that they have to be moved between 7 p.m. and 8 a.m. In Figure 12, we show the distribution of the widths of dispatch windows as determined by Phase II.

As can be seen, a few dispatches have little or no flexibility and have to be dispatched according to a specific schedule to make service; most likely these represent shipments in relatively long corridors with a 1-day service guarantee. At the other end of the spectrum are a few dispatches that have a lot of flexibility; most likely these

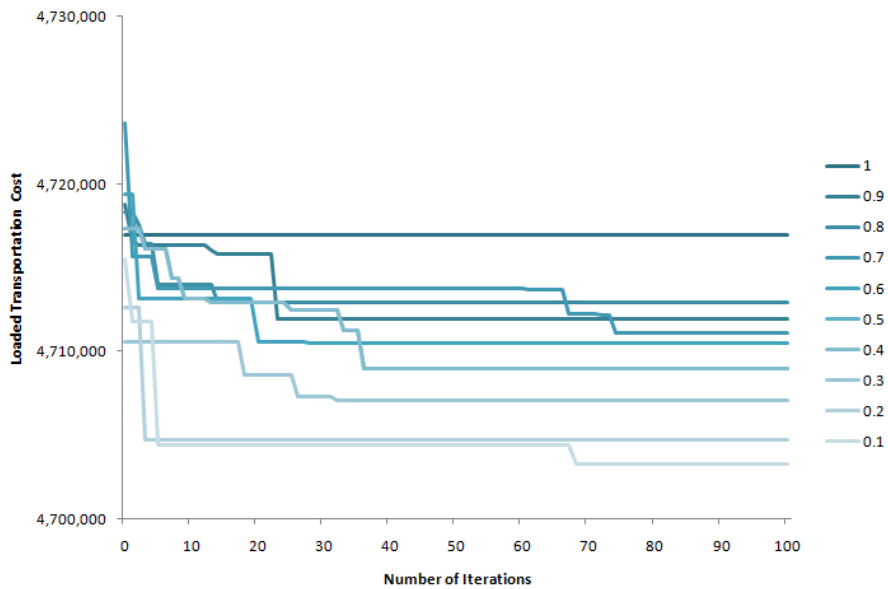

Figure 10: Progress of GRASP Heuristic for Instance $W4$

represent shipments on origin-destination pairs that are relatively close, but have a 5-day service guarantee. From an operational execution perspective, the most relevant information is that most dispatches have some flexibility, which can be exploited to build low-cost driver tours.

It is also insightful examine the dispatch windows on a single linehaul leg in more detail. Figure 13 shows all the Markham-Chicago dispatches and their dispatch windows.

A few interesting observations can be made. First, the dispatches occurring at 7 p.m. and 8 p.m., which likely represent a substantial portion of the shipments picked up during the day have little or no flexibility. Again, these likely represent dispatches involving shipments with 1-day service. Furthermore, we see that the dispatches between 10 a.m. and 6 p.m. have the most flexibility. These likely represent dispatches between breakbulks involving shipments with service levels that can relatively easily be achieved.

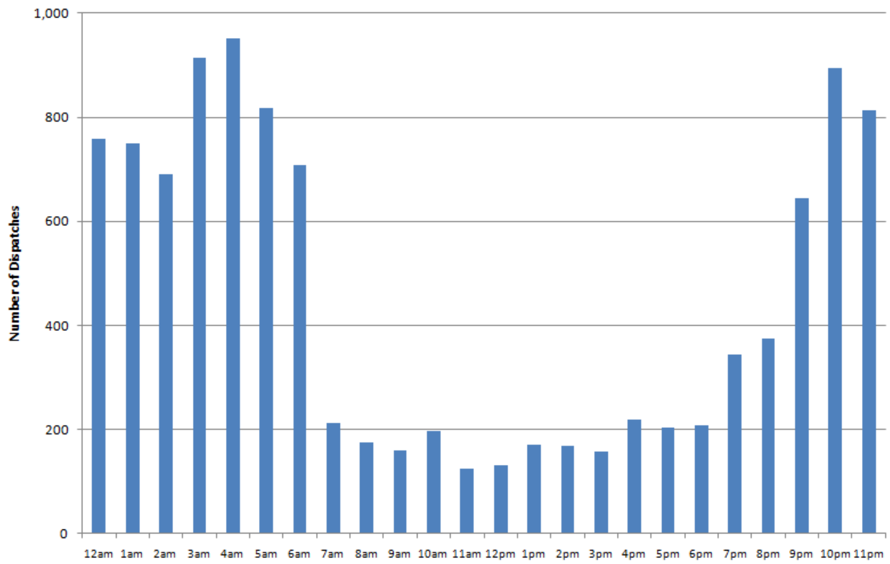

Figure 11: Dispatch Pattern

2.5.3 Column Generation and IP Optimization

Next, we consider parameter tuning for the column generation and IP optimization processes at the heart of Phase III. Recall that during the column generation process, if an attempt to build a tour with a negative reduced cost starting with a particular load fails, we exclude that load from consideration for the next M iterations to hopefully avoid spending computing time on finding negative reduced cost columns that is unlikely to be successful. The tradeoff between the computing time and the value of the final LP solution when we vary M is shown in Figure 14; for $M = 50, 100, 1,000, 10,000, \infty$.

We see that re-visiting loads provides a small benefit, but it comes at a very high price in terms computing time. Hence, for the remaining computational experiments, we have used $M = \infty$, *i.e.*, we will not re-visit a load ever again once our attempt to build a tour with a negative reduced cost starting from that load fails.

Next, we provide more details about the initial columns generated using structured templates; the templates are summarized in Table 5. Figure 15 shows the composition of the columns in the initial LP solution and the final LP solution in terms of their

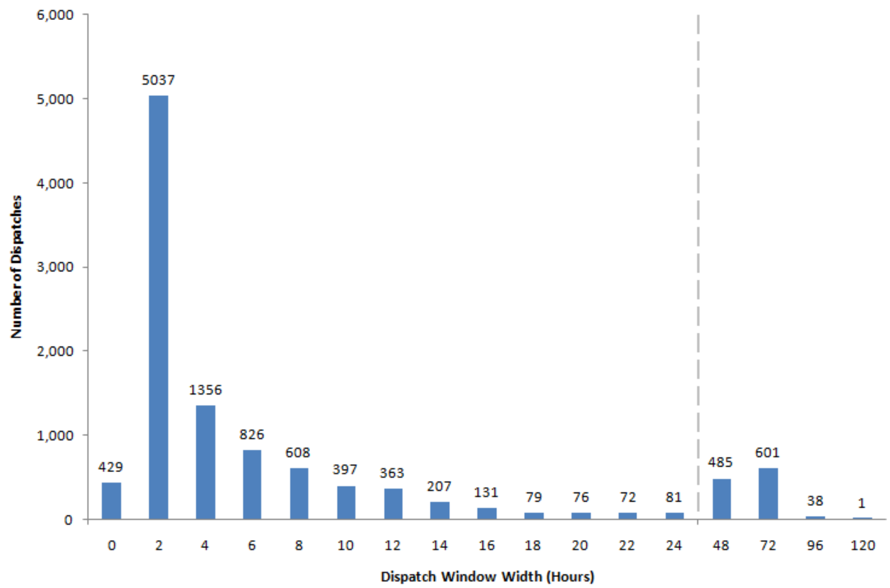

Figure 12: Dispatch Window Widths

Table 5: Template Types

Template code	Dispatch length	Type	Lay-down	Loaded/empty dispatch
SHORT-OB-LD	less than or equal to 5.5 hours	out-and-back	No	both loaded
LONG-MT-LD	more than 5.5 hours	meet-and-turn	No	both loaded
LONG-OB-LD	more than 5.5 hours	out-and-back	Yes	both loaded
SHORT-OB-EMT	less than or equal to 5.5 hours	out-and-back	No	1 loaded and 1 empty
LONG-MT-EMT	more than 5.5 hours	meet-and-turn	No	1 loaded and 1 empty

structure, i.e., the template corresponding to their structure.

Of course in the final LP solution we encounter structures that were not present in the initial LP solution. These structures are lumped together under the "template" COLGEN. For example, columns representing tours with duties involving more than 2 dispatches will end up under this template. This includes, for example, triangular duties, i.e., duties of the form A-B-C-A (dispatches AB, BC, and CA), which can be quite effective. Column generation is used precisely to generate such duties if desirable. The figure demonstrates that using these more complicated structures substantially reduces the use of inefficient structures with empty dispatches.

Finally, and most importantly, in Table 6 we report the value of the final LP solution and the value of the IP solution generated using the columns in the final LP solution (where the stopping criterion for the IP solve was an optimality gap of less

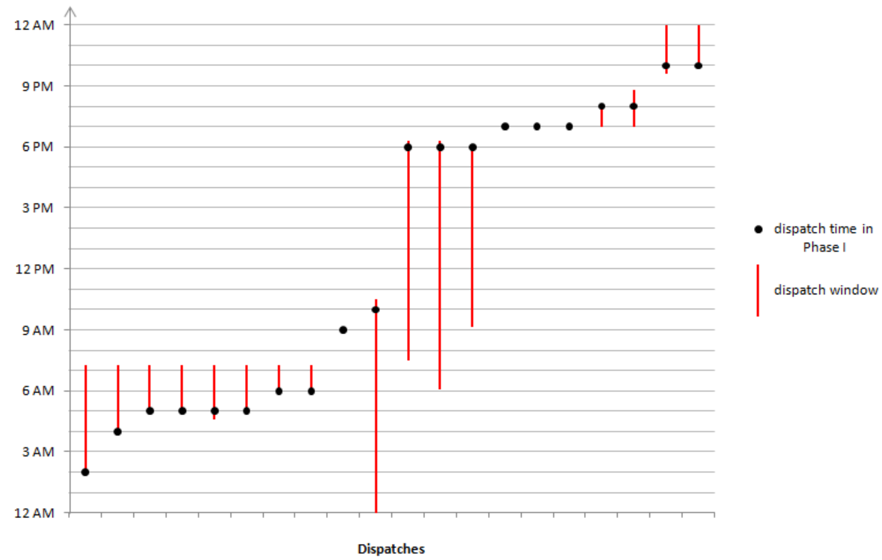

Figure 13: Markham-Chicago Dispatches with their Dispatch Windows

than 0.1%). We see that optimal or near-optimal solutions are produced.

Table 6: Comparison of LP and IP Solutions

	Value LP solution	Value IP solution
$W1$	3,247,583	3,249,398
$W2$	3,267,249	3,269,375
$W3$	3,311,176	3,314,513
$W4$	3,350,560	3,353,409

2.5.4 Tour Structures

In this section, we provide more details about the structure of the driver tours generated. In Table 7, we report the number of tours with 1 duty and 2 duties, the number of duties with 1, 2, 3, and 4 dispatches, and the number of duties involving meet-and-turns, and the number of loaded versus empty dispatches. We see that a relatively small percentage of duties involve more than 2 legs. Since counts only provide a partial picture, Figure 16 and 17 show the distribution of the driving and duty time of the duties.

We see that the majority of duties have a driving time of more than 7 hours and

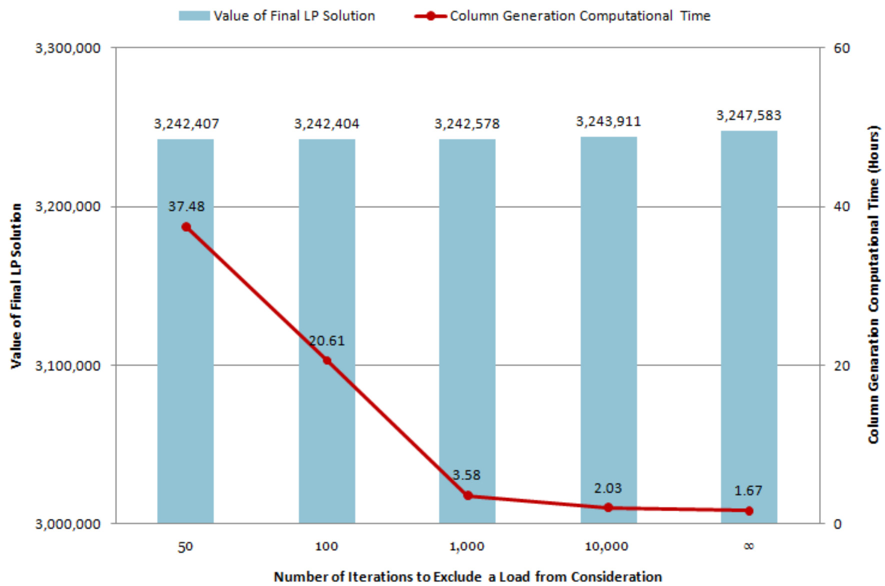

Figure 14: Impact of Varying the Number of Iterations to Exclude a Load from Consideration

a duty time of more than 9 hours, which is desirable. However, a non-trivial fraction corresponds to short duties with short driving times.

2.5.5 Execution Cost Estimates

The main goal of this research was to develop technology to accurately estimate the operational execution cost of a load plan. To demonstrate that we have achieved our goal, we present for the four instances, in Figure 18, the actual execution costs incurred by the carrier, the execution cost estimate produced by SuperSpin, the de facto industry-standard for load plan design, and our execution cost estimate (where the actual execution costs are normalized at 100% and the two estimates are given as a percentage of the actual costs).

The figure shows that our technology produces remarkably accurate execution cost estimates, within 1.7% of the actual execution costs incurred for each of the four weeks. The figure also shows that SuperSpin tends to under-predict execution costs (about 90% of the actual execution costs incurred), primarily due to over-estimation

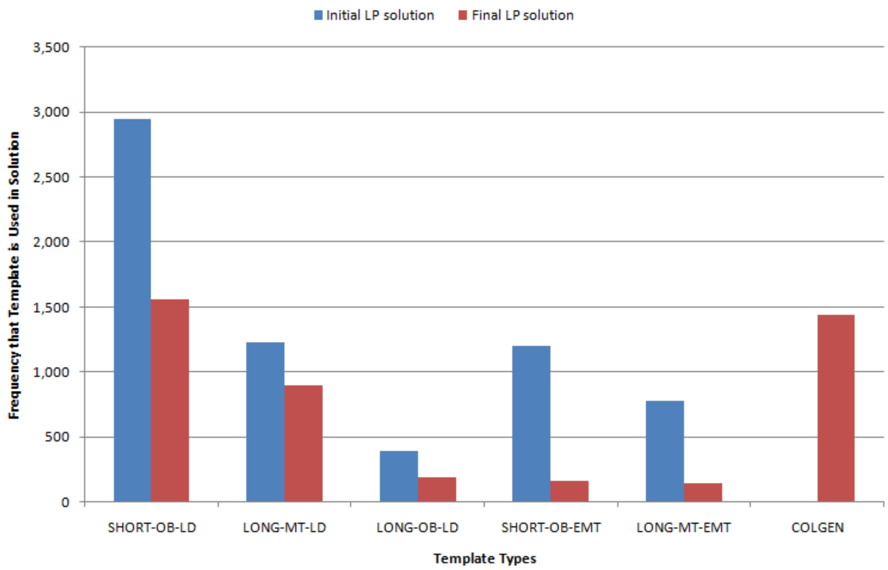

Figure 15: Template Uses in Solutions

of the consolidation opportunities.

Furthermore, we present, in Figure 19, for cost estimate the breakdown into loaded transportation costs, empty repositioning costs, and lay-down costs.

2.5.6 Varying Maximum Allowed Number of Dispatches in a Duty

During the construction of tours we limit the number of dispatches in a duty. There are two reasons for that. Firstly, duties with a small number of legs are preferred by both drivers and the carrier. Secondly, limiting the number of dispatches per duty limits the number of feasible duties and thus simplifies the pricing problem, which will reduce the computing time. In the next experiment, we investigate the impact of varying the maximum number of dispatches allowed in a duty. Figure 20 shows the total linehaul cost and the number of column generation iterations versus the maximum allowed number of dispatches in a duty.

As we allow a duty to contain more dispatches, the technology is able to generate more complicated and efficient driver tours, and thus to reduce the total linehaul cost. However, we see that the benefits of allowing more than 4 dispatches in a duty

Table 7: Tour Structure

		$W1$	$W2$	$W3$	$W4$
Tours	1 duty	4026	3993	4066	4068
	2 duties	1354	1387	1391	1436
	total	5380	5380	5457	5504
Duties	1 leg	1027	1029	1039	1099
	2 legs, non-meet-and-turn	2964	2981	2997	3008
	meet-and-turns	2040	2016	2064	2066
	3 legs	624	662	646	689
	4 legs	79	79	102	78
	total	6734	6767	6848	6940
Dispatches	loaded	11039	11073	11183	11361
	empty	1528	1649	1664	1626
	total	12567	12722	12847	12987

is negligible.

2.5.7 Short Driving and Duty Times

Up to now, short driving and duty times were not discouraged. As we observed in Figures 16 and 17, a majority of the duties have a driving and duty time close to their respective limits, but there are a fair number of duties with small driving and duty times.

In Figure 21, we analyze the tradeoff between the “quality” of the tours, in terms of their driving and duty time, and the operational execution costs.

We see that an increase in the average driving time of 1.5 hours and an increase in the average duty of 1 hour comes at an increase in operational execution costs of 1%.

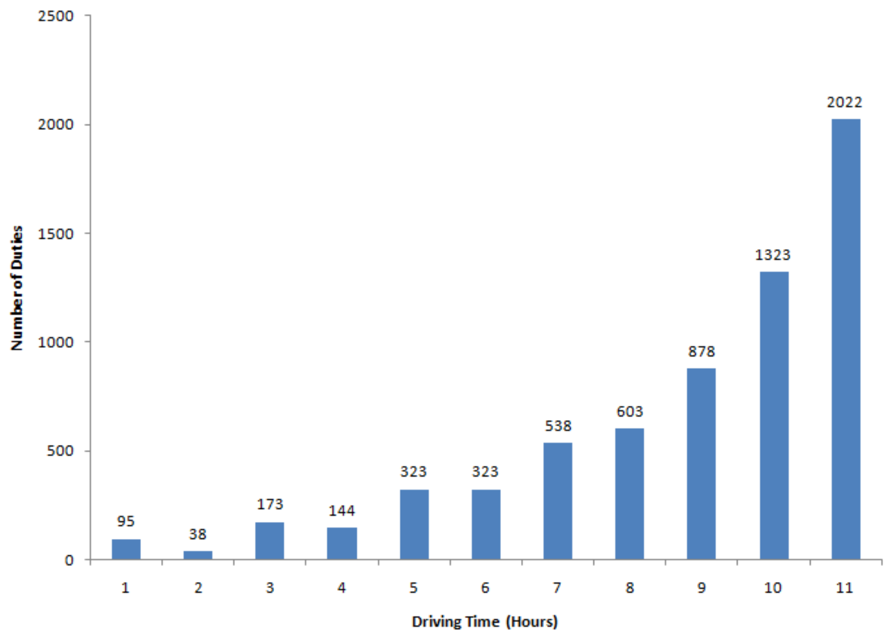

Figure 16: Driving Time Histogram

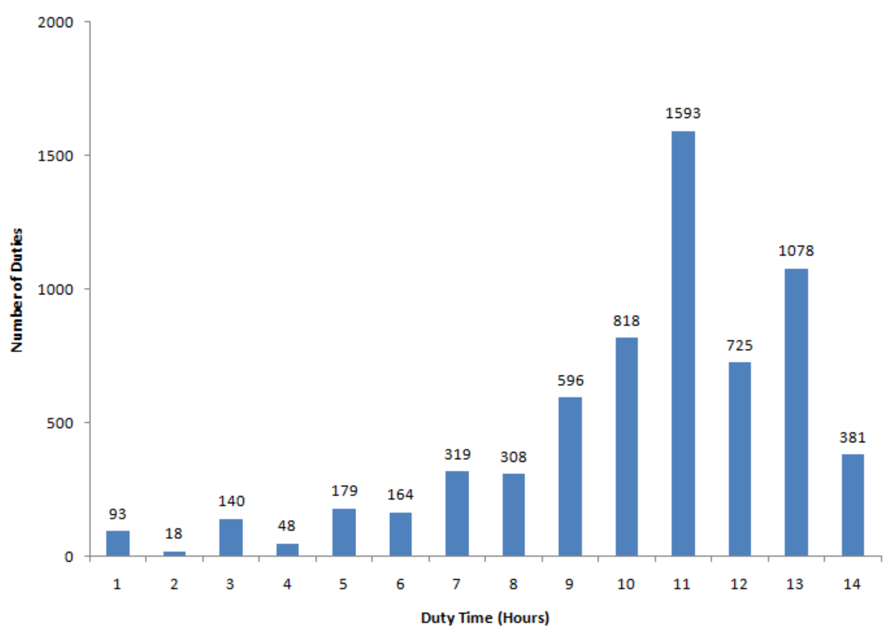

Figure 17: Duty Time Histogram

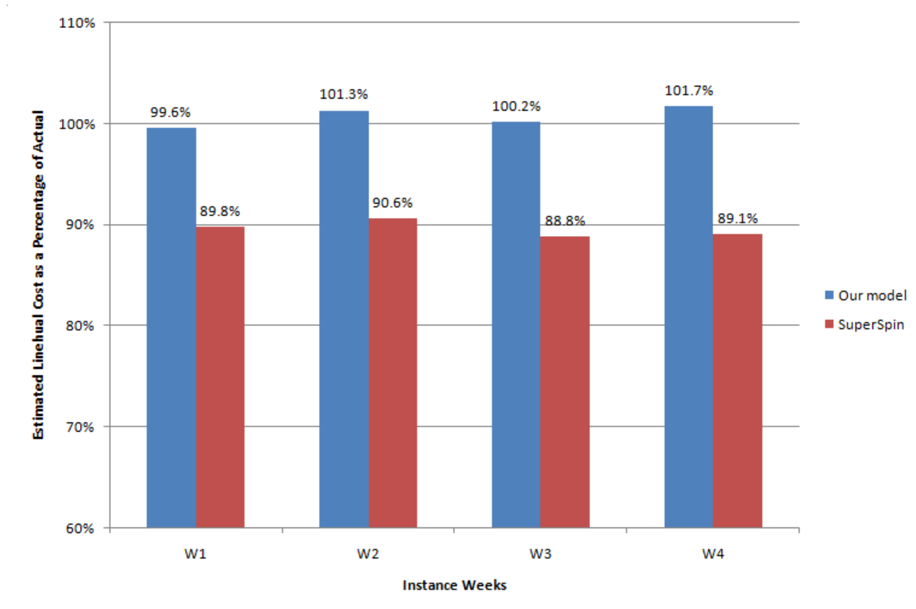

Figure 18: Estimated Linehaul Cost as a Percentage of Actual Execution Cost

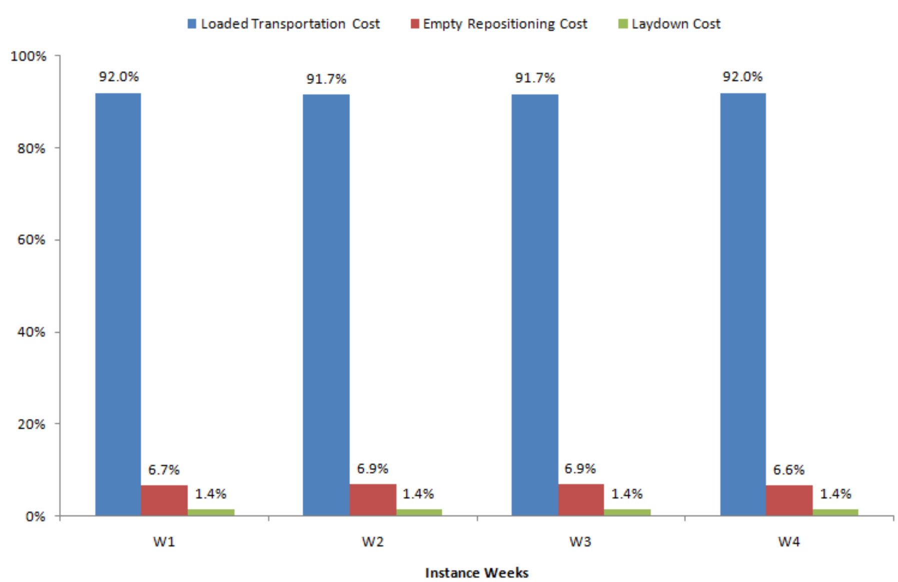

Figure 19: Cost Breakdowns

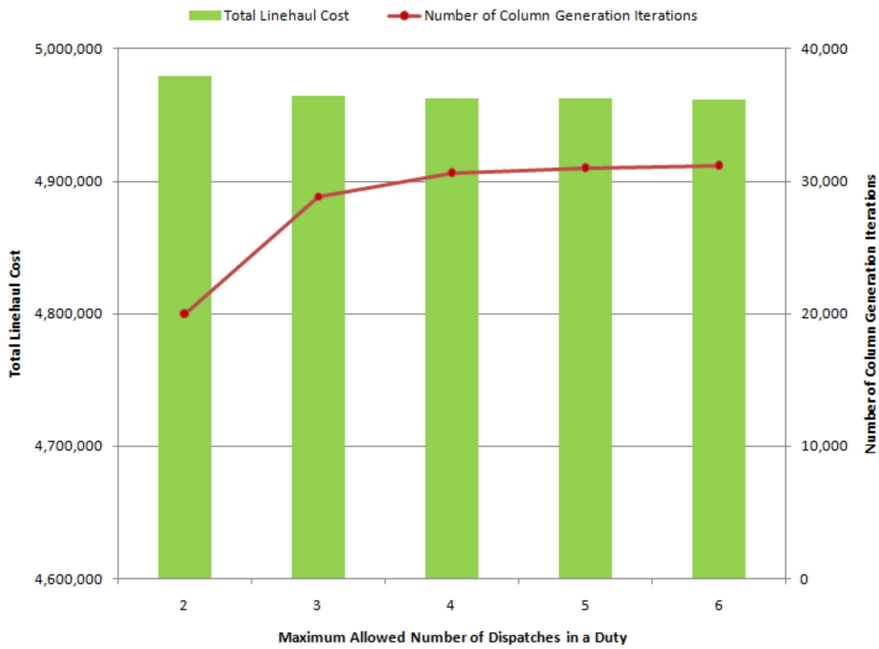

Figure 20: Impact of Varying the Maximum Number of Allowed Dispatches in a Duty

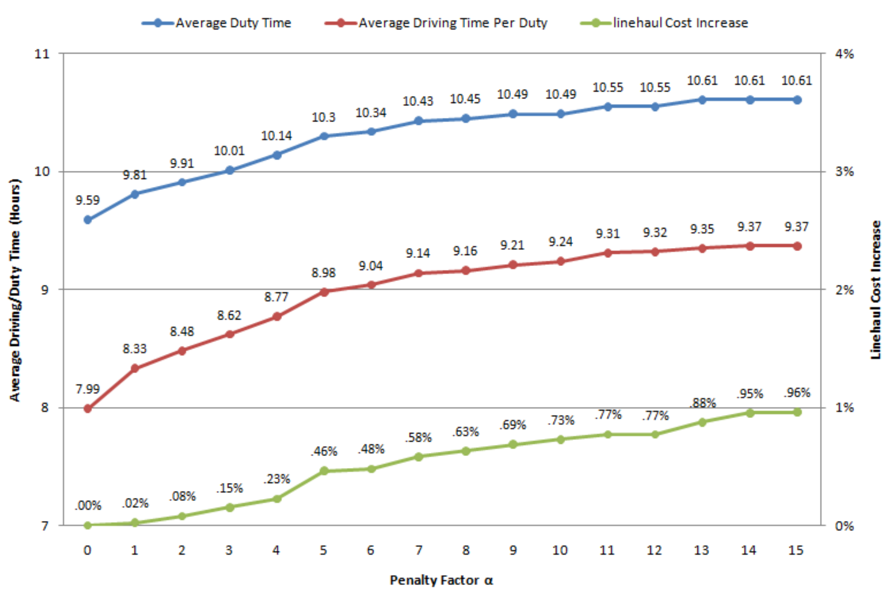

Figure 21: Impacts of Varying Penalty Factor α

CHAPTER III

DYNAMIC LOAD PLANNING

3.1 Introduction

Traditionally, carriers revise load plans somewhat infrequently, perhaps once every few months. Constructing a good load plan is a complex and time consuming task; load plan design is a difficult constrained optimization problem. Furthermore, implementing frequent load plan changes at a terminal could increase the rate of shipment routing errors. Since freight volumes do vary from day to day, carriers currently adapt the plans to take advantage of consolidation opportunities that decrease costs; currently this is performed locally by terminal managers, and may not result in effective systems-level decisions.

A number of technological advances and changes in practices at LTL firms are changing the service network design environment, and it is now possible for carriers to consider implementing frequent load plan updates:

- Hand-held scanners at pickup points of shipments provide immediate accurate information on actual freight picked up from customers during the day;

- Global positioning and mobile communication devices allow better tracking of in-transit freight; and

- Cross-dock automation again through hand-held scanner technology enables dock workers to reliably (re-)direct arriving shipments to the correct loading door, and reduces the necessity for a consistent predetermined plan.

In this chapter, we develop new technology for dynamic load planning (DLP), which efficiently and effectively alters a load plan for a given day based on accurate information on the current state of the LTL system, including an updated forecast of the actual freight to be transported and the current status of trailer resources. DLP technology is intended to be used each day at a time when nearly all new freight entering the system has been picked up (*e.g.*, 6 p.m.). Since DLP technology is to be used in a real-time operational environment, it needs to produce modified load plans in a short amount of time, *e.g.*, less than 5 minutes.

We present two approaches for solving DLP problems: an integer programming based local search procedure, and a greedy randomized adaptive search heuristic. A computational study using data from a national carrier currently implementing the technology demonstrates that the DLP solutions can produce significant cost savings.

The research presented in this chapter makes contributions in the context of load planning and algorithm design. Specifically,

- we are the first to study load planning in a dynamic operational setting, and to addresses the challenges encountered in such an environment;
- we efficiently solve real-life instances, requiring only minutes of computation time for the optimization of the entire network of a national carrier;
- we create load plan adjustments that produce significant cost-savings, in the range of 7-10 percent of total linehaul costs;
- we judiciously choose and exploit a set of templates for freight paths to successfully balance the tradeoff between solution quality and solution time; and
- we demonstrate an effective greedy randomized adaptive search inspired heuristic for a large-scale service network design problem.

As part of this study, we also examine the potential value of routing freight at an individual shipment level, which represents a new frontier in load planning. Our computational study shows that substantial cost savings are possible.

The remainder of this chapter is organized as follows. Section 3.2 details a few key modeling features for dynamic load planning. Section 3.3 outlines how we model freight routing and how we choose a set of templates for freight paths. Section 3.4 presents integer programming based approaches we have developed for DLP, and Section 3.5 describes the greedy randomized adaptive search heuristic.

3.2 Modeling Dynamic Load Planning

A time-space model is presented in [5] for the tactical static load planning problem in which it is assumed that origin-destination freight flow patterns repeat weekly. The model minimizes total weekly transportation and handling costs such that each origin-destination shipment satisfies prescribed service levels. The starting point for our dynamic load planning technology is a similar time-space model. Below we discuss a few key features.

3.2.1 Modeling Time

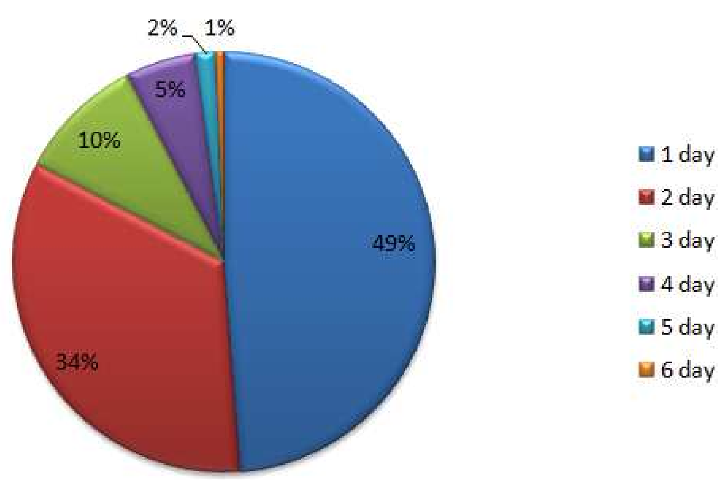

Figure 22: Freight Profile by Service Standard

Figure 22 shows that approximately 83% of the freight volume for the regional carrier that sponsored this research has a service standard of either 1 or 2 business days. Such short service standards reduce consolidation opportunities substantially since freight typically cannot wait very long at intermediate breakbulk transfer points. Thus, it is not only necessary to explicitly model time in any reasonable model, but to do so at a fine level of detail in order to correctly approximate transportation costs. For example, consider the following path for freight originating in Chicago, IL at 7 p.m. destined for Cincinnati, OH at 8 a.m. the next day:

- Travel for 4 hours from Chicago, IL to Indianapolis, IN
- Freight handling for 30 minutes in Indianapolis, IN
- Travel for 3 hours from Indianapolis, IN to Cincinnati, OH

Furthermore, consider the path for freight originating in St. Louis, MO at 7 p.m. also destined overnight for Cincinnati, OH:

- Travel for 5 hours from St. Louis, MO to Indianapolis, IN
- Freight handling for 30 minutes in Indianapolis, IN
- Travel for 3 hours from Indianapolis, IN to Cincinnati, OH

A time-space model that uses a daily time-granularity would conclude that each of these paths is infeasible, and thus a potential real-world consolidation opportunity is lost. Since freight is handled and consolidated primarily during the overnight hours, we divide a day into time windows separated by the following breakpoints: 1 a.m., 3 a.m., 5 a.m., 8 a.m., 10 a.m., 2 p.m., 7 p.m., 9 p.m., and 11 p.m. By specifying nodes at such times, we can time the dispatches such that the freight can be consolidated at Indianapolis, IN into a common trailer (or trailers) outbound to Cincinnati, OH:

- Chicago, IL - Cincinnati, OH freight:
  - Leave Chicago, IL at 7 p.m. CST, arrive at Indianapolis, IN at 12 a.m. EST

  - Finish handling at Indianapolis, IN at 12:30 a.m. EST
  - Leave Indianapolis, IN at 3 a.m. EST, arrive at Cincinnati, OH at 6 a.m. EST

- St. Louis, MO - Cincinnati, OH freight:
  - Leave St. Louis, MO at 7 p.m. CST, arrive at Indianapolis, IN at 1 a.m. EST
  - Finish handling at Indianapolis, IN at 1:30 a.m. EST
  - Leave Indianapolis, IN at 3 a.m. EST, arrive at Cincinnati, OH at 6 a.m. EST

3.2.2 Planning Horizon

In static load plan design, it is typically assumed that origin-destination freight flow patterns repeat, either daily or weekly. Thus, a wrapped version of the time-space network is appropriate, where arcs connect nodes at the end of the planning period to nodes at the beginning of the planning period. In dynamic load planning, a wrapped time-space network is not appropriate since actual origin-destination freight flows differ on a daily basis. Two key decisions must be made when setting up a time-space network for dynamic load plan generation: the length of the planning horizon, and the destination of freight with a due time after the end of the chosen horizon. We address the second issue by simply following the original load plan. That is, we set the destination for freight with a due time after the end of the planning horizon to be the last terminal along the path specified by the original load plan that falls within the horizon. See Figure 23 for an illustration.

This approach, although reasonable, can be restrictive if the chosen planning horizon is too short, in which case it may prevent us from consolidating on longer directs. Therefore, it is advantageous to have a horizon for which a high percentage of

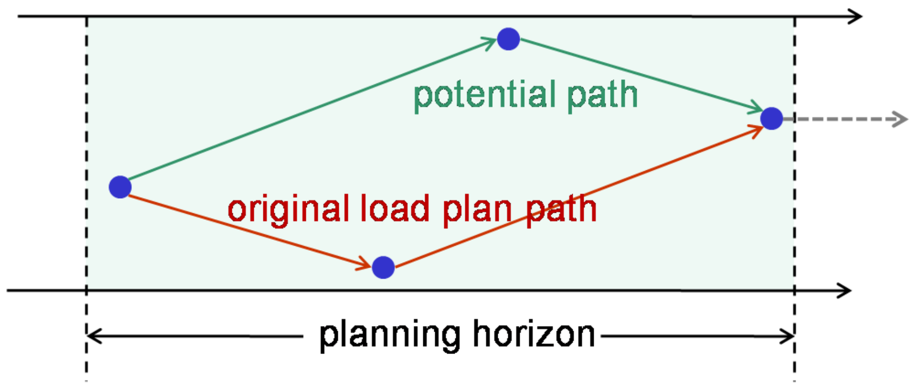

Figure 23: DLP Freight Target Destination

freight reaches its final destination within the horizon. Referring again to the freight volume profile in Figure 22, we see that 56% of the freight would reach its destination if we use a 24-hour planning horizon (starting and ending at 6 p.m.) and 84% if we extend the planning horizon to 38 hours (starting at 6 p.m. and ending at 8 a.m.) as it would include 2-day freight. Thus, we have chosen to work with a 38-hour horizon. See Figure 24 for an illustration.

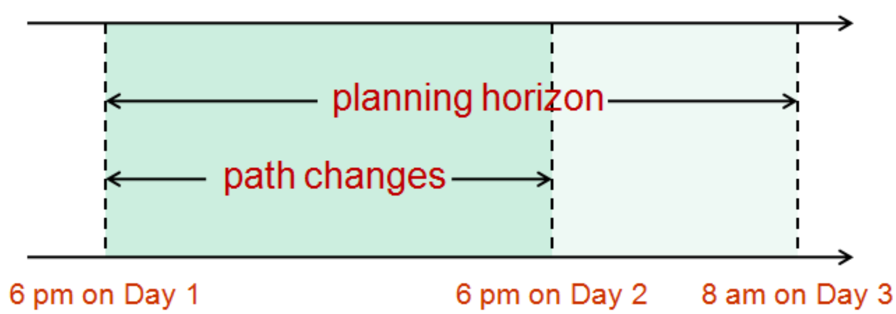

Figure 24: DLP Planning Horizon

For this study, we assume that DLP will be executed once each evening at 6 p.m., when most newly arriving freight has been picked up by collection tours. Note that since DLP will be executed each evening, any load plan adjustments (path changes) will only remain in effect for the next 24 hours despite the 38-hour planning horizon. This is therefore a fairly typical rolling horizon optimization approach.

There are four time zones in the continental U.S., creating additional challenges for dynamic load planning. For example, if DLP is run at 6 p.m. eastern standard time, which corresponds to 3 p.m. pacific standard time, originating freight information at terminals on the west coast may not be availabel yet. Multiple time zones can be accommodated by running DLP multiple times during the evening hours of each day. Each time, the system uses up-to-the-minute information on each shipment that is in the carrier's system and has not yet been delivered to their destination terminal, and projections of future freight volumes in later time zones, to make decisions. Earlier decisions may be modified as freight volume projections become more accurate.

3.2.3 Initial Trailer Resources

When transferring freight arrives at a terminal during the day, dock workers begin to build outbound trailers according to the then active load plan. Therefore, when DLP is run at 6 p.m., some shipments have already been loaded into outbound trailers with designated destinations. The outbound directs for these shipments are considered fixed and will not be modified regardless of whether the trailer has been closed and moved from the loading door, or not. However, if a trailer is open at a loading door and has remaining capacity, DLP may add freight to that trailer. When such an outbound trailer arrives at its destination, the shipments will be unloaded from the trailer and can then be re-routed and re-consolidated.

3.2.4 Accounting for Empty Trailer Movements

Freight flows tend to be imbalanced, *e.g.*, there is more freight flowing into Florida, then flowing out of Florida. Thus, trailers need to be repositioned periodically from freight "sinks" to ensure the availability at freight "sources". The repositioning of empty trailers creates opportunities for routing actual freight: transporting freight in a trailer that is being repositioned is essentially free. For example, suppose that we have the following freight flows on the network depicted in Figure 25:

- One trailerload originating at C and destined for A,

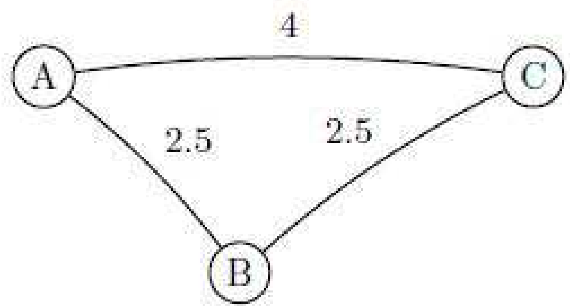

Figure 25: Example Network Showing Per Trailer Cost

- One trailerload originating at C and destined for B,
- A half trailerload originating at A and destined for C, and
- A half trailerload originating at B and destined for C.

Given that full trailerloads are traveling from C to A and from C to B, there is no need to consolidate those freight flows. If we ignore the need to repositioning trailers and only focus on consolidation, then we would route the half trailerload from A to C via B to consolidate it with the half trailerload from B to C (see Figure 26, where the numbers above arcs represent the number of trailers dispatched, and the numbers beside each node represent the balance of trailers at that node.) To restore trailer balance a trailer must be sent from B to C.

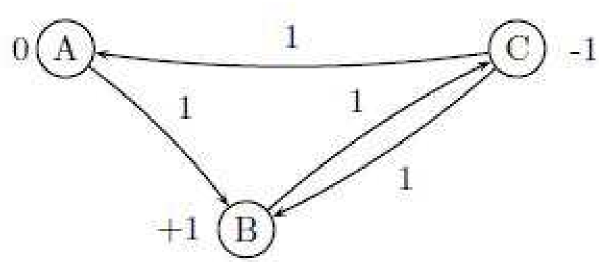

Figure 26: Trailer movements when repositioning is not considered

If on the other hand, we recognize that more freight is flowing out of C then flowing into C, and thus that trailers must be repositioned back to C, it now is cost-effective to send half a trailerload from A to C and half a trailerload from B to C (see Figure 27).

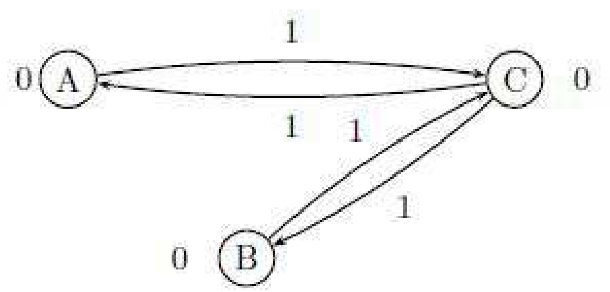

Figure 27: Trailer movements when repositioning is considered

The latter solution avoids repositioning and is likely to be cheaper than the former solution. This is also what would happen in practice; a carrier would recognize that $A \rightarrow C$ and $B \rightarrow C$ are backhaul lanes, and hence would not want to move freight away from those lanes.

Properly accounting for trailer repositioning in the context of dynamic load planning is complicated since it is not necessary for each terminal to maintain strict trailer balance within the planning horizon; the actual timing of repositioning moves need not be explicitly resolved. We have chosen a pragmatic approach in this research, and we do not allow freight to shift away from known backhaul lanes even if that may seem to lead to better consolidation. This approach is reasonable, since most empty trailer repositioning is caused by freight flow imbalances and does not vary much with load plan changes.

Backhaul lanes can be derived by examining the original load plan. To compute the set of backhaul lanes, we use a time-space network with a week-long planning horizon, and techniques similar to those developed in Chapter 2 to build trailer loads based on freight volume projections for each day of the week, according to the original load plan. Then a minimum cost flow problem is solved on a static (space) network to resolve trailer imbalances and produce the set of backhaul lanes.

3.2.5 Load Plan Structure

Traditional load plans require that freight (whether originating or transferring) at a terminal destined for a common destination will be loaded next to a common outbound terminal. This simplifies terminal operations since a dock worker only has to examine the destination of a shipment to determine the appropriate outbound trailer for loading. This restriction enforces an "in-tree" structure into each destination.

Currently, however, dock automation technology, such as handheld scanners that allow a dock worker to read the outbound trailer off a display after scanning an inbound shipment, make it possible to relax this constraint and route freight at the shipment level. We will consider and analyze both options and will refer to them as **DLP-INTREE**, where the load plan is adjusted, and **DLP-SPLIT**, where shipments are re-routed on an individual basis.

3.3 Freight Path Templates

The goal of dynamically adjusting the load plan is to find the best consolidation opportunities given the actual freight in the system. That means determining the path that freight in the system will follow the next 24 hours, *i.e.*, selecting the directs used to route freight and establishing when and where to hold freight to maximize consolidation. As discussed in Section 3.2, to make these decisions we model terminals and potential directs on a time-space network. To be more precise, let (U, L) denote the carrier's linehaul network, where U is the set of terminals in the carrier's network and L is the set of potential directs connecting terminals. For a given time discretization of a planning horizon T, we define the time-space linehaul network (N, A), where N denotes the set of nodes and A denotes the set of arcs. Each node

$n = (u,t), u \in U, t \in T$ represents a terminal at a particular point in time. Let $N^* = \{(u,t) \in N | t \leq 6$ p.m. the next business day$\}$. Each arc $a = ((u_1, t_1), (u_2, t_2))$ with u_1 and $u_2 \in U$ and $u_1 \neq u_2$ represents a potential dispatch from u_1 at time t_1 on direct (u_1, u_2) arriving in u_2 at time t_2. We create such arcs for each direct $l = (u_1, u_2) \in L$ and each timed copy (u_1, t_1) of the origin node u_1. The destination node (u_2, t_2) is then chosen to be the earliest timed copy of the node u_2 such that $t_2 - t_1$ is no less than the transit time of the underlying direct l. We also create arcs $a = ((u_1, t_1), (u_1, t_2))$ to connect subsequent timed copies of each node u_1. These allow us to model holding a trailer or shipment at terminal u_1.

Given networks (U, L) and (N, A), let $\delta^+(u) \subseteq L$ denote the set of potential outbound directs from terminal $u \in U$; for each arc $a \in A$, let $l(a)$ denote the direct $l \in L$ corresponding to a, c_a denote the per-trailer travel cost along arc a, M_a^w denote the maximum weight per trailer in pounds, and M_a^b denote the maximum cube per trailer in cubic feet.

We model freight that enters the linehaul network as commodities in the time-space network. These commodities include

1. Actual picked up shipments. We model them as entering the time-space network at the node representing the first location and time where we can make changes. The meaning of this depends on the status of the shipment at the time DLP is run:

   - *In a trailer in transit, or in a closed outbound trailer in a yard.* The commodity enters the time-space network at the next handling terminal at the estimated arrival time at the next handling terminal.
   - *In an inbound trailer at a terminal, still to be unloaded.* The commodity enters the time-space network at its current terminal at a time equal to the current time plus the estimated time required to unload it.

- *On the dock, unloaded but not yet loaded.* The commodity enters the time-space network at its current terminal at the current time.
- *In an open outbound trailer at door.* The commodity enters the time-space network at its current terminal at the current time. The destination of the first direct is fixed and all shipments in this open outbound trailer must depart the on the same timed copy of the direct.

2. Projected freight volumes still to be picked up during the current day d_1. We model projected freight entering the linehaul network at terminal u_1 as entering the time-space network at node $n_1 = (u_1, t_1)$ where $t_1 = \max(\text{current time}, d_1$ @ 7 p.m.).

3. Projected freight volumes to be picked up during the next business day $d_1 + 1$. This freight is included because the planning period covers freight originating on the next business day. We model freight entering the linehaul network at terminal u_1 as entering the time-space network at node $n_1 = (u_1, t_1)$ where $t_1 = d_1 + 1$ @ 7 p.m.

As discussed earlier, freight destined for terminal u with a due day d within the planning period, *i.e.*, $d \leq d_1 + 2$ is given destination node $n = (u, d$ @ 8 a.m.), and freight with a due date after the end of the planning period is given as destination node the last node within the planning period on the path specified by the original load plan.

Let K denote the set of commodities. For each commodity $k \in K$, let $o(k)$ denote the origin terminal, $d(k)$ denote the destination terminal, w_k denote the weight in pounds, b_k denote the cube in cubic feet. Let $K(d) \subseteq K, d \in U$ denote the set of commodities with destination terminal d.

Given that there is little time available for adjusting load plans, *i.e.*, at most 5 minutes, it is crucial to carefully select the adjustments to consider. As load planning is all about identifying freight paths, we have chosen to work with a set of freight path templates that correspond to load plan changes that are most likely to provide additional opportunities for consolidation. The freight path templates are discussed below:

- *Skip direct.* Skip directs are motivated by situations where freight volumes are large enough to justify building longer directs to save on handling costs at intermediate terminals. Consider the example in Figure 28. The original load plan path from A to D is $A - B - C - D$, where B and C are handling points. If we have a nearly full trailer worth of freight at B bound for C and then D, it makes sense to build a direct trailer from B to D, skipping C and thus saving the handling cost that would have been incurred at C. In general, we consider skipping each handling point on the original load plan path, *e.g.*, create paths $A - C - D$ and $A - B - D$ in this case.

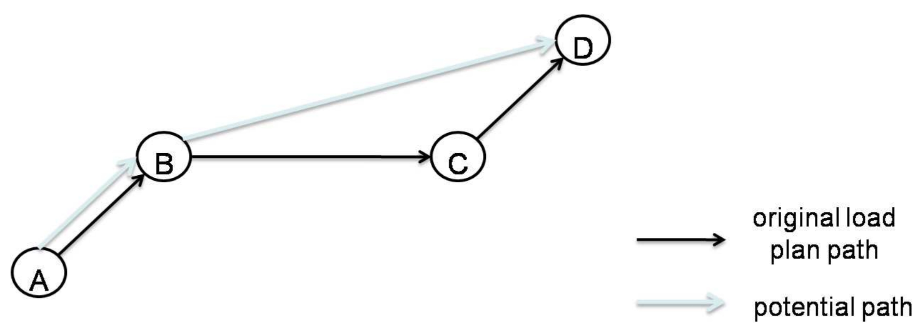

Figure 28: Skip Direct

- *Add direct.* In opposite situations where we have low load factors on a direct, we may save on transportation costs by "breaking up" the original direct and introducing extra handles to increase consolidation. Consider the example in Figure 29. The original load plan path from A to D is $A - C - D$; breakbulk terminal B is on the route from A to C. Suppose that the load factor on direct $A - C$ is low, and we already have a half-full trailer flowing on $B - C$, then by breaking direct $A - C$ and handling the freight at B, we save transportation costs of moving one trailer on $B-C$. If this saving dominates the handling cost at B, this change reduces the total system cost. We examine direct $C-D$ for similar opportunities.

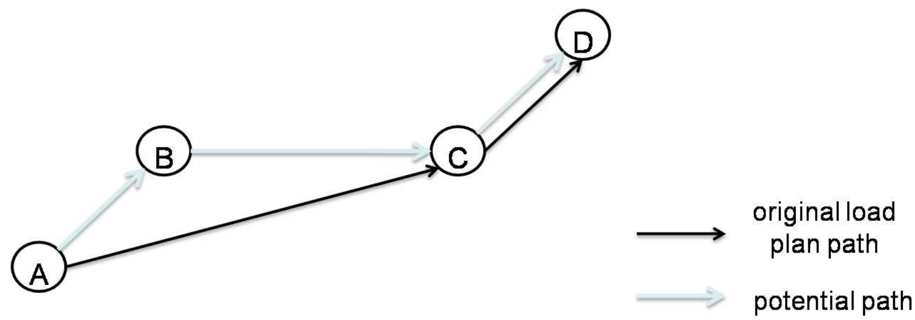

Figure 29: Add Direct

- *Alternate outbound at freight origin.* Loading freight to alternate outbound directions may improve consolidation. This is most effective near the freight origin, where the freight has not been heavily consolidated. Consider the example in Figure 30. The original load plan path from A to F is $A-B-C-F$. At freight origin A, instead of loading the shipment to B, we consider loading it to another direction D, and from D continue on the original load plan path to F, resulting in $A-D-E-F$. If, for example, the remaining capacities of trailers already flowing on $A-D$, $D-E$ and $E-F$ all can accommodate the $A-F$ freight, while routing it on the original load plan path $A-B-C-F$ would require opening new trailers, then the new route is preferred (ignoring handling cost differences).

  When we repeat this for each alternate outbound direction at A, we could potentially create a large number of paths. To keep the size of potential paths manageable, we only consider up to the L_O shortest ones among them.

- *Alternate outbound at origin breakbulk.* When freight originates at an end-of-line, in addition to alternate outbound loading at freight origin, we also search

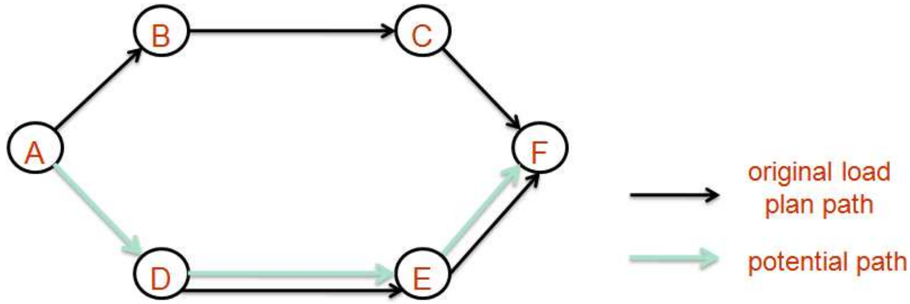

Figure 30: Alternate Outbound at Freight Origin

for such opportunities at the origin breakbulk, *i.e.*, the first breakbulk (second overall) terminal on the original load plan path. Consider the example in Figure 31. The original load plan path from A to E is $A - B - C - E$. This time, we keep the loading decision at the freight origin A (to B), but examine alternate outbound directions at the first breakbulk B, *e.g.*, to D. Then we continue on the original load plan path from D to E, resulting in a new path $A - B - D - E$. We repeat this procedure for each alternate outbound direction at B and only the shortest L_B paths are admitted.

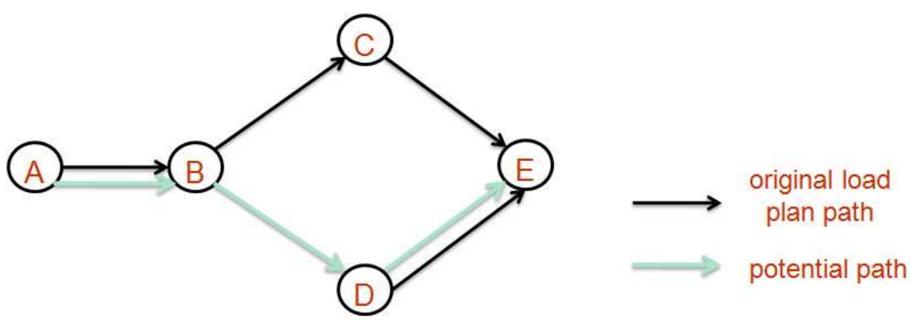

Figure 31: Alternate Outbound at Origin Breakbulk

Besides load plan changes that improve consolidation, additional savings may be achieved by executing so called “milk runs” involving freight origins and destinations. Specifically, when multiple end-of-lines dispatch trailers to a common breakbulk, or vice versa when a breakbulk dispatch trailers to multiple end-of-lines, it may be possible combine the trips from (or to) multiple end-of-line into a single one by stopping at an intermediate end-of-line to pick up (or kick off) some freight, and then continuing on to the final destination of the trip.

- *Inbound Milk Run.* Consider the example in Figure 32. End-of-lines A and B both load trailers to breakbulk O. If we load $A-O$ freight as the headload of the $A-O$ trailer, and route it through B, we could potentially pick up $B-O$ freight at B, continue on to O, and thus save a separate dispatch from B to O.

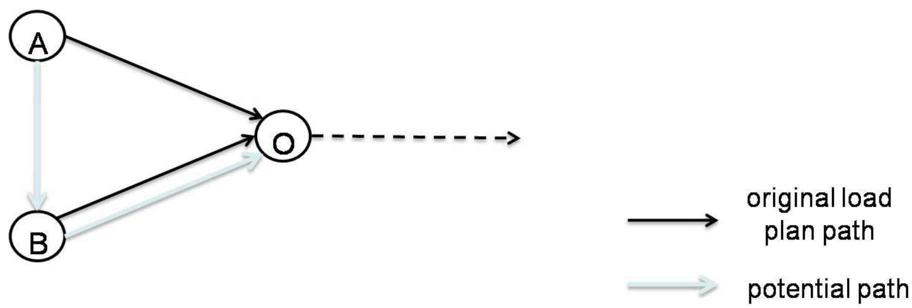

Figure 32: Inbound Milk Run

- *Outbound Milk Run.* Consider the example in Figure 33. Breakbulk O loads trailers to end-of-lines A and B. If we load $O-B$ freight as the headload of the $O-B$ dispatch, $O-A$ freight on the same trailer towards the back of it, and route the it though A, we could kick off $O-A$ freight at A, continue on to B, and thus save a separate dispatch from O to A.

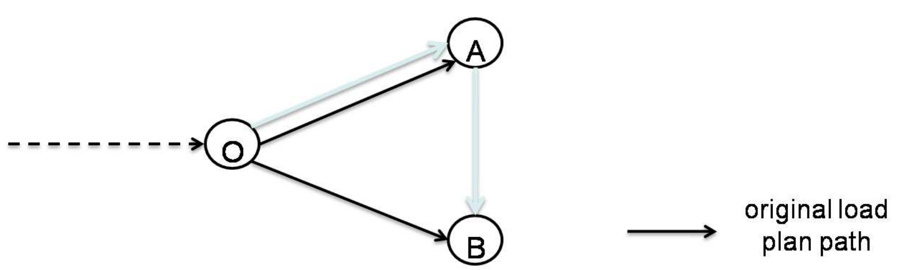

Figure 33: Outbound Milk Run

3.4 Integer Programming Approaches

Due to the success of integer programming based search for the static load planning problem, as reported in [5], it seemed natural to explore the same approach in the context of dynamic load planning. The approach uses a path-based optimization model on the time-space network. Let $P(k)$ be a set of possible freight paths for commodity $k \in K$, where a freight path p is a sequence of arcs, *i.e.*, $p = (a_1, \ldots, a_{n_p})$. Each path $p = (a_1, \ldots, a_{n_p}) \in P(k)$ connects the origin and the destination node of k. How commodity k is routed then simply becomes a question of choosing a path $p \in P(k)$. Associated with a path $p = (a_1, \ldots, a_{n_p})$ is an underlying path p of directs $\mathsf{p} = (l(a_1), \ldots, l(a_{n_p}))$. Note that given a path p, we can calculate its total handling cost h_p per pound by summing the costs for the intermediate terminals visited.

To construct a set of paths $P(k)$ for commodity k, we first check the service feasibility of the paths generated using the templates discussed in the previous section. That is, we compare the remaining time until the due time at the destination and the minimum amount of time required to reach the destination, which is the sum of

- the transit times of the directs,
- 30 minutes per intermediate handling for 1-day freight (special handling procedures have been put in place at breakbulks to streamline the processing of 1-day freight so as to ensure that it can make service), or two hours per intermediate handling for multi-day freight, and
- 30 minutes for loading or unloading at the intermediate stop of a milk run.

We then map the service-feasible paths to the time-space network. For each such path of directs, we include not only the minimum duration path p into $P(k)$, but potentially also other versions that add holding arcs if they are also feasible. Adding such timed copies models the ability to hold freight at intermediate terminals to improve the plan. We construct a limited set of such paths by only holding freight until specific events occur. First, we allow freight to be held at a terminal until the time that new freight originates at that terminal; thus, freight arriving at a breakbulk during the day can be consolidated with that evening's originating outbound freight. Second, we allow freight to be held at a terminal until its cut time, *i.e.*, the latest time at which the freight can be dispatched and still arrive on time to its destination. In this way, freight destined for common destinations may be consolidated.

When selecting paths for commodities, we must ensure consistency between the paths chosen for commodities in a common open outbound trailer. Therefore, let T denote the set of open outbound trailers. For each $t \in T$, let $C(t) \subseteq K$ be the set of commodities in trailer t and $D(t) \subseteq A$ be the set of possible dispatch arcs for t, *i.e.*, a set of timed copies of the same direct.

3.4.1 DLP-SPLIT Integer Program

We first present an integer programming formulation, referred to as DLP-SPLIT-IP, for DLP-SPLIT. It has three sets of decision variables. First, x variables indicate whether commodity k uses path p, *i.e.*, $x_p^k \in \{0,1\}\ \ \forall k \in K,\ \ \forall p \in P(k)$. Second, z variables enforce consistency between paths for commodities in an open outbound trailer by indicating whether arc a is chosen for open trailer t, *i.e.*, $z_a^t \in \{0,1\}\ \ \forall t \in T,\ \ \forall a \in D(T)$. Finally, τ variables count the required number of trailers that move on arc a, *i.e.*, $\tau_a \in \mathbb{Z}_+\ \ \forall a \in A$.

The formulation is to then minimize

$$\sum_{a \in A} c_a \tau_a + \sum_{k \in K} \sum_{p \in P(k)} h_p w_k x_p^k$$

subject to

$$\sum_{p \in P(k)} x_p^k = 1\ \ \forall k \in K \tag{9}$$

$$\sum_{a \in D(t)} z_a^t = 1\ \ \forall t \in T \tag{10}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq z_a^t \quad \forall t \in T, \;\; \forall k \in C(t), \;\; \forall a \in D(t) \tag{11}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} w_k x_p^k \leq \tau_a M_a^w \quad \forall a \in A \tag{12}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} b_k x_p^k \leq \tau_a M_a^b \quad \forall a \in A \tag{13}$$

The objective function represents the total transportation and handling costs. Constraints (9) ensure that a path is chosen for each commodity. Constraints (10) ensure that a single departure arc is selected for each open outbound trailer. Constraints (11) ensure that a path for a commodity in an open outbound trailer can only be chosen when the first arc on the path is the same as the departure arc for the open trailer. Constraints (12) and (13) ensure that the number of trailer trailers on an arc is sufficient to carry the freight that moves along the arc, *i.e.*, the freight "assigned" to the arc through the chosen paths.

3.4.2 DLP-INTREE Integer Program

As mentioned earlier, a traditional load plan specifies the outbound direct a shipment should take at its current location given its final destination. Choosing the outbound direct for freight at terminal u and destined for terminal d (regardless of its origin or service standard) corresponds to choosing a single arc from $\delta^+(u)$ for freight destined to node $d \in U$.

In our path-based approach, we choose for each commodity $k \in K$ a path of arcs, where each arc a is a timed copy of a direct. Therefore, when adjusting a load plan, we must ensure consistency among the paths chosen for commodities with a common final destination, *i.e.*, for commodities $k \in K(d)$ for all $d \in U$.

Additional y variables are introduced to enforce consistency between paths for commodities with common destinations indicating whether direct $l \in \delta^+(u)$ is chosen for commodities destined for terminal d routed through terminal u until 6 p.m. the next business day, *i.e.*, $y_l^d \in \{0, 1\} \;\; \forall d \in U, \;\; \forall l \in \delta^+(u), u \in U$. Assuming the load plan reverts back to the original one after that, we let $\widehat{y}_l^d$ denote the fixed outbound direct decisions given by the original load plan. For each $a \in A$, let $o(a)$ be the origin node of a.

The formulation, referred to as DLP-INTREE-IP, is to minimize

$$\sum_{a \in A} c_a \tau_a + \sum_{k \in K} \sum_{p \in P(k)} h_p w_k x_p^k$$

subject to

$$\sum_{p \in P(k)} x_p^k = 1 \;\; \forall k \in K \tag{14}$$

$$\sum_{a \in D(t)} z_a^t = 1 \;\; \forall t \in T \tag{15}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq z_a^t \;\; \forall t \in T, \;\; \forall k \in C(t), \;\; \forall a \in D(t) \tag{16}$$

$$\sum_{l \in \delta^+(u)} y_l^d \leq 1 \;\; \forall u \in U, \;\; \forall d \in U \tag{17}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq y_{l(a)}^{d(k)} \;\; \forall k \in K, \forall a \in A, o(a) \in N^* \tag{18}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq \widehat{y}_{l(a)}^{d(k)} \;\; \forall k \in K, \forall a \in A, o(a) \in N \setminus N^* \tag{19}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} w_k x_p^k \leq \tau_a M_a^w \;\; \forall a \in A \tag{20}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} b_k x_p^k \leq \tau_a M_a^b \;\; \forall a \in A \tag{21}$$

The additional constraints (17) ensure that a single outbound direct is selected for freight at terminal u and destined for terminal d. Constraints (18) and (19) ensure that a path for commodity k can only be chosen when all of its component directs are chosen.

3.4.3 Inbound-IP Based Search

Realistically-sized instances of DLP-SPLIT-IP and DLP-INTREE-IP cannot be solved directly by commercial integer programming solvers, let alone within 5 minutes.

A decomposition approach that uses exact optimization within heuristic search is proposed in [5]; see Algorithm 5 for a general outline of the procedure. We apply the same technique to the dynamic load planning problem.

Algorithm 5 Integer Programming Based Neighborhood Search

Require: a feasible solution to the integer program
  while the search time has not exceeded a prespecified limit T **do**
    Choose a subset of variables V
    Solve the integer program with all variables not in V fixed at their current value
    if an improved solution is found **then**
      Update the best known feasible solution
    end if
  end while

The choice of a subset of variables V is motivated by the "in-tree" structure of traditional load plans, *i.e.*, directs into a destination d must form an in-tree (see Figure 34). We refer to the associated integer program as an Inbound IP into d, or IIP_d, with DLP-SPLIT-IP and DLP-INTREE-IP variants. The purpose of IIP_d is to improve the current solution by optimally choosing the directs used for d-bound freight, and by optimally choosing when and where d-bound freight is held. The IIP_d problem is to determine a set of paths for all commodities in $K(d)$; note that this problem is then to determine a new directed in-tree into d. More formally, given a current feasible solution $(\bar{z}, \bar{y}, \bar{x}, \bar{\tau})$, IIP_d is defined by holding fixed the variables

- $y_l^u = \bar{y}_l^u \quad \forall u \in U$ such that $u \neq d$,
- $x_p^k = \bar{x}_p^k \;\; \forall k \in K \setminus K(d)$, and
- $z_a^t = \bar{z}_a^t \;\; \forall a \in D(t), \;\; \forall t \in T$ such that $C(t) \not\subseteq K(d)$.

A specialized version of Algorithm 5 is presented in Algorithm 6. Since our approach improves the load plan by re-routing freight destined for a specific terminal, we do not want to spend time solving Inbound IPs for terminals for which little freight is destined. Thus, we only consider the top $p\%$ of terminals for which freight is destined.

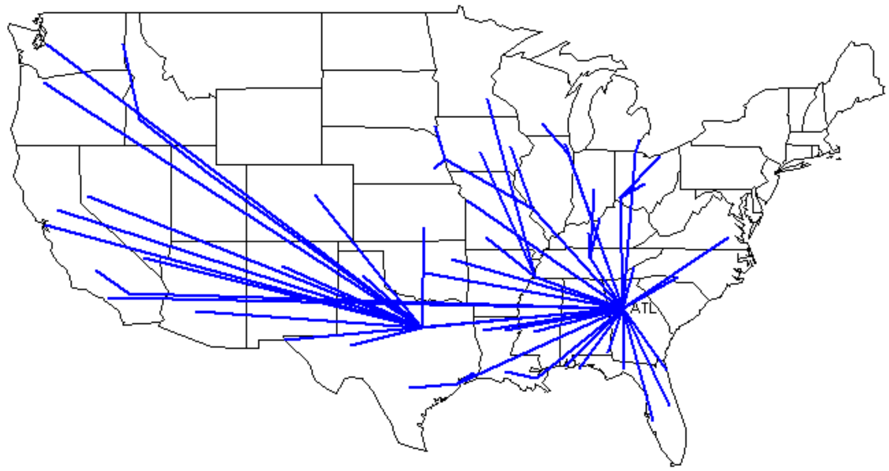

Figure 34: Freight Paths into a Destination Terminal

The algorithm we present iterates through this subset of terminals in a round-robin manner.

Algorithm 6 IIP Neighborhood Search

Require: an initial load plan $(\bar{z}, \bar{y}, \bar{x}, \bar{\tau})$
  for each terminal d **do**
    Set $F_d = \sum_{k \in K(d)} w_k$, the total amount of freight destined for d
  end for
  Set $\mathcal{T}$ = array of top $p\%$ of terminals with respect to F_d
  Sort $\mathcal{T}$ in descending order of F_d
  Set $i = 0$
  while the search time has not exceeded the prespecified limit **do**
    Choose destination terminal $d = \mathcal{T}[i \ \ mod \ \ |\mathcal{T}|]$
    Solve Inbound-IP IIP_d
    if Solution to IIP_d gives lower total load plan cost **then**
      Update $(\bar{z}, \bar{y}, \bar{x}, \bar{\tau})$
    end if
    Set $i = i + 1$
  end while

3.4.4 Computational Results

The algorithms were developed in C++ with CPLEX 11 as the Mixed Integer Program solver, interfaced via ILOG Concert Technology. All computational experiments were carried out on a system with a 2.66 GHz Intel Xeon processor and 4 GB of RAM.

Our test set for the integer programming based approaches consists of the five data instances described in Table 8, each representing a snapshot of the system of a national LTL carrier in the U.S. The carrier's linehaul network consists of 58 breakbulks, 103 end-of-lines, and approximately 24,000 potential directs. A typical DLP run involves approximately 20,000 commodities and 1,000 open outbound trailers in the starting condition.

Table 8: Data Instances for Inbound IP Approach

Instance	Description
$I1$	18:00 EST, Monday, March 23, 2009
$I2$	18:00 EST, Tuesday, March 24, 2009
$I3$	18:00 EST, Wednesday, March 25, 2009
$I4$	18:00 EST, Thursday, March 26, 2009
$I5$	18:00 EST, Friday, March 27, 2009

The computational experiments for the integer programming based approaches were conducted using a slightly simplified version that bases trailer computations only on weight but not cube. The Inbound-IP approach successfully finds cost savings for the less-constrained DLP-SPLIT-IP problem. In Table 9, we report the cost savings measured in percentages relative to the initial load plan provided by the carrier, and each type of paths used. This approach, however, has its limitations in dynamic load planning mainly due to the much more aggressive target run time of 5 minutes. Figure 35 shows the progress of the Inbound-IP approach over the course of its execution. We see that only a small fraction of the savings are achieved in the first 5 minutes because only a few terminals can be re-optimized in that time. For the more constrained DLP-INTREE-IP, the Inbound-IP approach was unable to find improving solutions in 15 minutes.

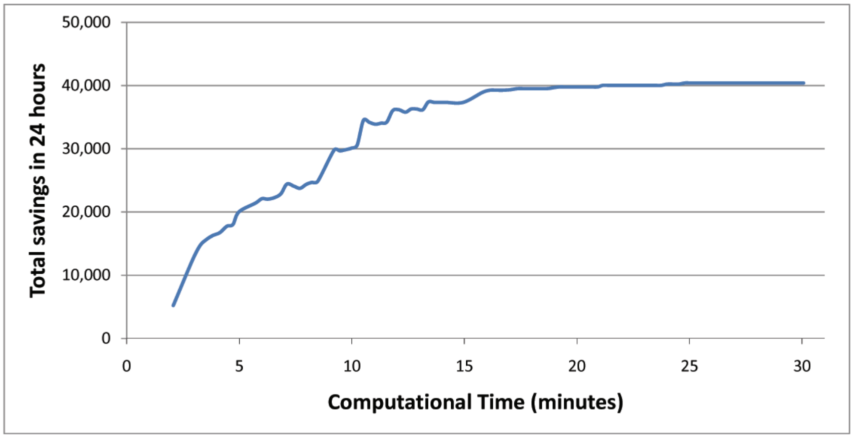

Figure 35: Progress of Inbound-IP Approach

Table 9: Computational Results for Inbound IP Approach for DLP-SPLIT-IP

Instance	$I1$	$I2$	$I3$	$I4$	$I5$
cost savings	4.64%	6.08%	5.82%	8.68%	5.59%
run time (minutes)	15	15	15	15	15
skip direct	0.45%	0.55%	0.49%	0.51%	0.34%
add direct	0.21%	0.11%	0.20%	0.16%	0.20%
alternate outbound at origin	1.40%	1.39%	1.27%	0.91%	0.75%
alternate outbound at origin breakbulk	1.21%	1.13%	1.36%	0.93%	0.96%
inbound milk run	0.62%	0.73%	0.62%	0.87%	0.37%
outbound milk run	0.12%	0.10%	0.06%	0.14%	0.00%

3.5 GRASP-Inspired Heuristic Approaches

The computational experiments discussed above show that the integer programming based approaches are too time-consuming to be of value for dynamic load planning. We next describe a heuristic approach, inspired by the concepts of Greedy Randomized Adaptive Search Procedures (GRASPs), that is capable of producing high-quality load plan adjustments in a short amount of time.

To evaluate a tentative load plan, a timed-copy of the path specified by the load plan needs to be determined for each commodity so as to maximize consolidation (or equivalently to minimize the total cost). Ideally, these timed-copies should be determined simultaneously for all commodities. However, that is computationally prohibitive and an effective and efficient randomized greedy procedure has been developed instead. The efficiency of the randomized greedy procedure relies on the fact that finding an optimal timed-copy of the path specified by the load plan for a single, given commodity can be done efficiently using a shortest path algorithm. The marginal cost of adding commodity k with weight w_k and cube b_k to arc a with current freight flow weight e_a^w and freight flow cube e_a^u, and maximum freight flow weight M_a^w and maximum freight flow cube M_a^b per trailer equals

$$\left(\lceil \max(\frac{e_a^w + w_k}{M_a^w}, \frac{e_a^u + b_k}{M_a^b})\rceil - \lceil \max(\frac{e_a^w}{M_a^w}, \frac{e_a^u}{M_a^b})\rceil\right) c_a. \tag{22}$$

A shortest path algorithm using these arc costs finds the least marginal cost path for commodity k. The greedy aspect of the approach is due to the order in which the commodities are processed. Let a commodity's slack time be defined as the maximum amount of time it can be held at its origin such that it can still meet service. A large slack time is an indication of more flexibility for choosing dispatch times along the path, and hence more opportunities for taking advantage of "free" capacity on trailers along the way that have been "opened" for transporting other commodities. On the other hand, a small slack time for a commodity implies that there is little or no flexibility for choosing dispatch times so as to consolidate freight. Therefore, unless there is a trailer with sufficient remaining capacity dispatched at the exact time required by this commodity, a new trailer has to be opened to accommodate this commodity, which may lead to a low load factor and may thus be costly. This suggests that we select paths for commodities with a small slack time first, and overlay those with paths for commodities with larger slack times. Furthermore, a commodity of lighter weight is more likely to be able to take advantage of remaining capacity on open trailers, which suggest we select paths for commodities with larger weights first. We introduce randomness into the algorithm by processing the commodities not entirely in the order described above. Rather, a Restricted Candidate List (RCM) consisting of the top m commodities (according to the above sorting criterion) is maintained. Then in each iteration, we select a commodity within the RCM with equal probability; or with probability proportional to λ^i for the ith commodity, $\lambda \in (0,1)$.

With an effective and efficient way to evaluate a tentative load plan, it is possible to design local search procedures to dynamically identify high-quality adjustments to a load plan.

3.5.1 Local Search for DLP-INTREE

When dynamically adjusting a load plan, the paths selected for commodities must satisfy the consistency requirements for a load plan, *i.e.*, the paths into a particular destination must form an intree. Therefore, when evaluating a potential adjustment to the load plan, whether it is selecting an outbound direction for an origin-destination pair, or determining whether freight on a direct between an end-of-line and a breakbulk should become part of a milk run, we must collectively consider all the commodities that are affected by such a change.

Our local search heuristic consists of two phases. In the first phase, we search for improving load plan changes for origin-destination pairs. In the second phase, we search for milk run opportunities.

In the first phase, we process all origin-destination pairs in some random order. Given an origin-destination pair (o, d), we determine the set $L(o, d)$ of commodities that are affected by a change in the outbound direction at o for freight destined to d, *i.e.*, the set of commodities destined for d that visit o in the next 24 hours. Next, we evaluate the tentative load plans that result when we change the outbound direction at o for freight destined to d. The freight path templates are used to restrict the outbound directions considered when creating tentative load plans. The evaluation of a tentative load plan is done using the randomized greedy procedure outlined above.

Because the dynamic load planning technology will be executed again in 24 hours, we do not know the load plan that will be in place after 24 hours. However, our planning period covers 38 hours, so we do need to have a load plan in place for last 14 hours of the planning period. We have chosen to use the static load plan for that as that was designed to perform well throughout the week.

In the detailed description given in Algorithm 7, the function $f(n,d)$ represents a tentative (partial) load plan, *i.e.*, it specifies for freight at node n with final destination d, the next destination. We use $u(n)$ to denote the terminal associated with node n and $LMC(k,f)$ to denote the least marginal cost path for commodity k given tentative load plan f.

Algorithm 7 Load Plan Change Local Search for DLP-INTREE

Require: a set s of paths for each commodity, a load plan l

$I \leftarrow$ a random ordering of all origin-destination pairs
for all $(o,d) \in I$ **do**
  $L(o,d) \leftarrow$ set of commodities destined for d and visiting o in the next 24 hours
  $s^* \leftarrow \emptyset$
  $v^* \leftarrow \emptyset$
  for all $v \in$ potential outbound directions from o to d **do**
    // Create tentative load plan
    $f \leftarrow f(n,d) = \begin{cases} v & \text{if } u(n) = o \text{ and } n \in N^* \\ l\big(u(n),d\big) & \text{otherwise} \end{cases}$
    // Evaluate tentative load plan
    $s \leftarrow s \setminus L(o,d)$
    for all $k \in L(o,d)$ **do**
      $s \leftarrow s \bigcup LMC(k,f)$
    end for
    if s improves s^* **then**
      $s^* \leftarrow s$
      $v^* \leftarrow v$
    end if
  end for
  $s \leftarrow s^*$
  $l \leftarrow (l \setminus (o,d)) \bigcup v^*$
end for

In the second phase, we process all directs between an end-of-line and a breakbulk, *i.e.*, the candidates for a milk run, in some random order. Given a direct, we determine the set $L(dir)$ of commodities that are dispatched on the direct in the next 24 hours.

Next, we evaluate the tentative load plans that result when we visit an intermediate milk run stop on the direct. The freight path templates are used to restrict the milk runs considered when creating tentative load plans. As before, the evaluation of a tentative load plan is done using the randomized greedy procedure outlined above. A detailed description is given in Algorithm 8.

Algorithm 8 Milk Run Local Search for DLP-INTREE

Require: a set s of paths for each commodity, a load plan l
  $m \leftarrow \emptyset$ the set of milk runs
  $I \leftarrow$ a random ordering of all directs between an end-of-line and a breakbulk
  for all $dir \in I$ **do**
    $L(dir) \leftarrow$ set of commodities dispatched on dir in the next 24 hours
    $s^* \leftarrow \emptyset$
    $v^* \leftarrow \emptyset$
    for all $v \in$ potential milk run stops for dir **do**
      // Create tentative load plan

$$f \leftarrow f(n,d) = \begin{cases} \text{destination of milk run} & \text{if } u(n) = \text{intermediate stop of a milk run} \\ v & \text{if } \Big(u(n), l\big(u(n), d\big)\Big) = dir \text{ and } n \in N^* \\ m\Big(u(n), l\big(u(n), d\big)\Big) & \text{if } \Big(u(n), l\big(u(n), d\big)\Big) \in m \text{ and } n \in N^* \\ l\big(u(n), d\big) & \text{otherwise} \end{cases}$$

      // Evaluate tentative load plan
      $s \leftarrow s \setminus L(dir)$
      for all $k \in L(dir)$ **do**
        $s \leftarrow s \bigcup LMC(k, f)$
      end for
      if s improves s^* **then**
        $s^* \leftarrow s$
        $v^* \leftarrow v$
      end if
    end for
    $s \leftarrow s^*$
    $m \leftarrow m \bigcup (dir, v^*)$
  end for
  return s

3.5.2 Local Search for DLP-SPLIT

As mentioned earlier, in the near future it may be possible to route freight at an individual shipment level in which case it will no longer be necessary to enforce consistency among shipments with the same final destination. In that case, the additional flexibility can be exploited by adding a third phase in which we check sequentially for each commodity if we can improve the current load plan by replacing its current path with another one. The set of alternate paths is specified by the freight path templates. In the detailed description in Algorithm 9 an alternate path $\bar{p}$ is given as a sequence of terminals, *e.g.*, $(\bar{p}[1], \bar{p}[2], ..., \bar{p}[n])$.

Algorithm 9 Phase 3 for DLP-SPLIT

Require: a set s of paths for each commodity
  Sort commodities in increasing order of slack time; in case of ties, in decreasing order of weight
  while stopping criterion not met **do**
    Create a copy L of the commodity list
    while $L \neq \emptyset$ **do**
      Select a commodity k according to a randomized greedy strategy
      $L \leftarrow L \setminus k$
      $s^* \leftarrow \emptyset$
      for all $\bar{p} \in$ path options from $o(k)$ to $d(k)$ **do**
        // Create tentative load plan
        $f \leftarrow f(n, d(k)) = \begin{cases} \bar{p}[i+1] & \text{if } u(n) = \bar{p}[i] \text{ and } n \in N^* \\ l(u(n), d(k)) & \text{otherwise} \end{cases}$
        // Evaluate tentative load plan
        $s \leftarrow (s \setminus k) \bigcup LMC(k, f)$
        if s improves s^* **then**
          $s^* \leftarrow s$
        end if
      end for
      $s \leftarrow s^*$
    end while
  end while
  return s

3.5.3 Computational Results

Our test set for the GRASP-inspired heuristic approaches consists of the four data instances listed in Table 10, each representing a snapshot of the system of a national U.S. LTL carrier. The carrier's linehaul network consists of 58 breakbulk terminal, 103 end-of-line terminals, and approximately 24,000 potential directs. A typical DLP run involves approximately 20,000 commodities and has about 1,000 open outbound trailers.

Table 10: Data Instances for GRASP-Inspired Heuristic Approaches

Instance	Description
$G1$	18:35 EST, Monday, January 26, 2010
$G2$	17:15 EST, Friday, March 5, 2010
$G3$	19:09 EST, Friday, March 5, 2010
$G4$	20:45 EST, Tuesday, March 30, 2010

The dynamic load planning technology improves a load plan that was constructed using freight projections by exploiting consolidation and cost-savings opportunities created by the actual freight in the system. In Table 11, we report the cost savings and computation times obtained for the four instances when we run both DLP-INTREE and DLP-SPLIT variants of the heuristics. Again, the cost savings are measured in percentages relative to the cost of the initial load plan provided by the carrier for that day. Since a 1% savings represents about $10,000 for the carrier, these suggested changes can have a substantial impact on the carrier's bottom line. Furthermore, the results also show that relaxing the in-tree requirement of a traditional load plan and routing freight at an individual shipment level will increase the savings even more. Finally, we observe that all computation times are less than 3 minutes.

The dynamic load planning technology is built around a set of specific load plan adjustments. In Table 12 we present a breakdown of the cost savings by load plan adjustment.

We see that, as expected, the majority of the cost savings are found in the load

Table 11: Computational Results for GRASP-Inspired Heuristic Approaches

Instance	DLP-INTREE		DLP-SPLIT	
	% cost savings	run time (seconds)	% cost savings	run time (seconds)
$G1$	8.46%	120	9.26%	124
$G2$	10.32%	112	11.53%	122
$G3$	8.60%	133	10.08%	146
$G4$	7.21%	144	8.14%	148

Table 12: Cost Savings Breakdown by Path Types

	$G1$	$G2$	$G3$	$G4$
skip direct	6.33%	3.04%	3.03%	3.88%
add direct	30.78%	24.75%	23.08%	22.05%
alternate outbound	60.13%	67.47%	66.95%	70.67%
inbound milk run	2.63%	4.60%	6.10%	2.72%
outbound milk run	0.13%	0.13%	0.84%	0.68%

plan change phase, but that the cost savings provided by milk-runs is non-trivial. The bulk of the cost savings come from adding directs and sending freight on alternate outbound directs. Identifying such adjustments requires knowledge of the freight flows at several terminals and therefore are not likely to be found by terminal managers. The use of optimization techniques with a system-wide view is crucial.

In the load plan change phase, we consider every origin-destination pair once. Given that the technology requires less than 3 minutes, we have investigated if there is benefit of considering every origin-destination pair twice. The results are presented in Figure 36, which show how the cost-savings accumulate over time. We see that the vast majority of the savings are found in the first pass and there is no real need to expand the extra time and effort in a second pass.

3.5.3.1 Illustrative Examples

In this section, we present a few examples chosen from the computational results to demonstrate the changes that were made by DLP. They were all selected from the results of DLP runs on instance $G4$. In each example we show side-by-side the "before" and "after" pictures of the freight flows, in trailerloads, under the original

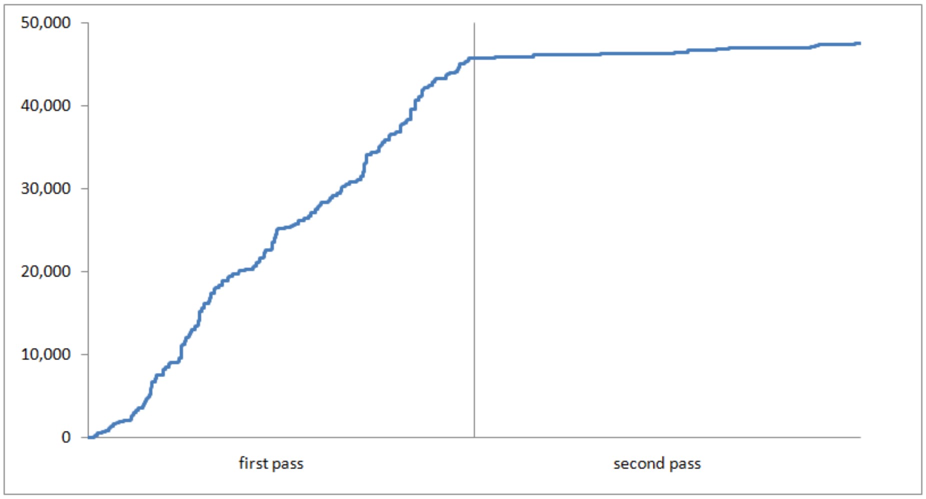

Figure 36: Progress of Load Plan Change Local Search

load plan on the left, and under the DLP-optimized load plan on the right. The required number of trailers on a direct is simply the freight flow rounded up to the nearest integer.

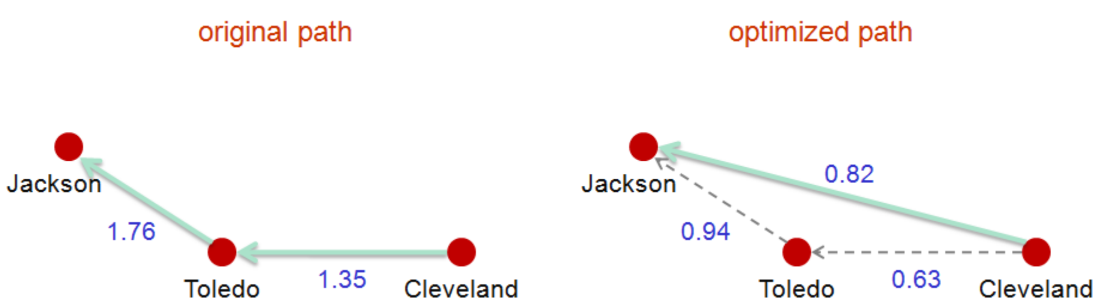

Figure 37: Skip Direct from Computational Results

Figure 37 illustrates a skip direct. Under the original load plan, Cleveland freight destined for Jackson goes to Toledo, and then to Jackson. Two trailers are required on each of Cleveland-Toledo and Toledo-Jackson directs. Out of these freight volumes, 0.82 trailerloads are Cleveland-Jackson freight. DLP determines that we should build a direct trailer from Cleveland to Jackson. The same number of trailers are required as under the original load plan, but we save the handling cost for Cleveland-Jackson freight that would have been incurred at Toledo. In practice, this direct trailer from Cleveland to Jackson will likely still travel on Cleveland-Toledo-Jackson route and be matched with the remaining trailers, but will not be opened at Toledo.

Note that the numbers may not exactly add up for two reasons: there may be other changes involved; and the freight volume is computed based on both weight and cube, so the numbers may not be additive.

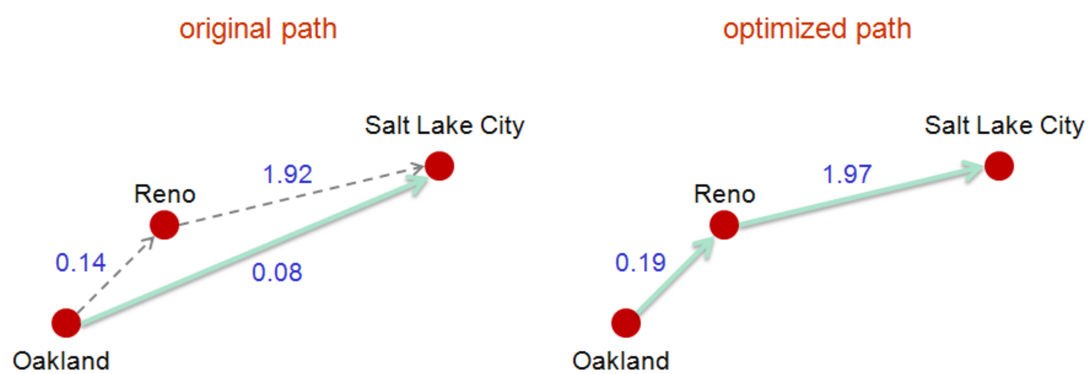

Figure 38: Add Direct from Computational Results

Figure 38 illustrates an add direct. Under the original load plan, Oakland-Salt Lake City freight travels direct even though we have only 0.08 trailerloads of such freight, while on Oakland-Reno and Reno-Salt Lake City directs there are trailers with enough remaining capacities to accommodate such freight. DLP thus determines that we should "break" the Oakland-Salt Lake City direct and re-route the freight through Reno. By doing so we save the transportation costs of dispatching one trailer from Oakland to Salt Lake City.

Figure 39 shows an alternate outbound change. The freight origin-destination pair under consideration is Lubbock-Victoria. Under the original load plan, such freight travels from Lubbock to Houston, and then to Victoria. DLP finds that at the freight origin Lubbock, instead of loading the freight to Houston, we can shift it to the Dallas lane without having to open up new trailers on either Lubbock-Dallas or Dallas-Victoria directs, and we do not require a trailer on Lubbock-Houston direct anymore.

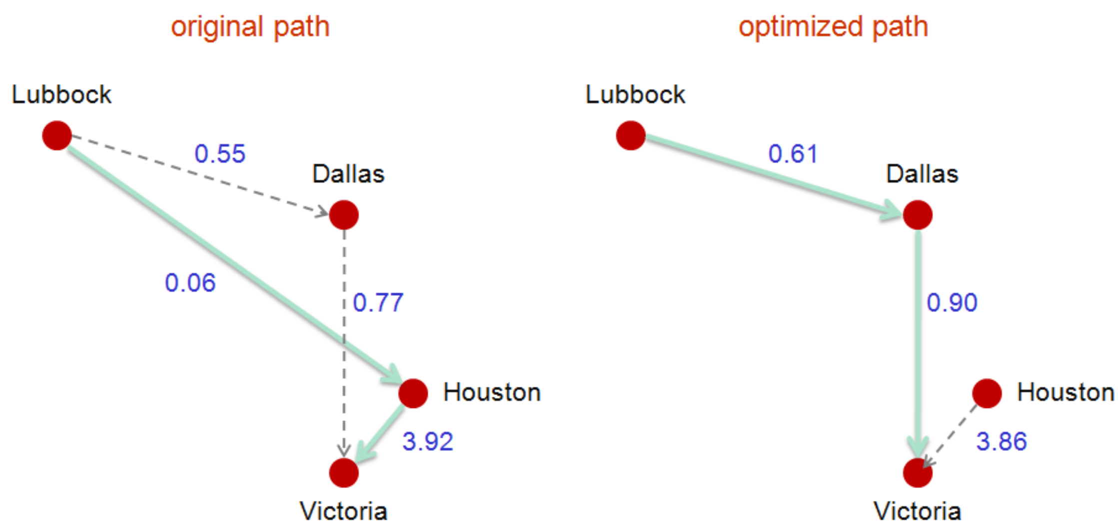

Figure 39: Alternate Outbound from Computational Results

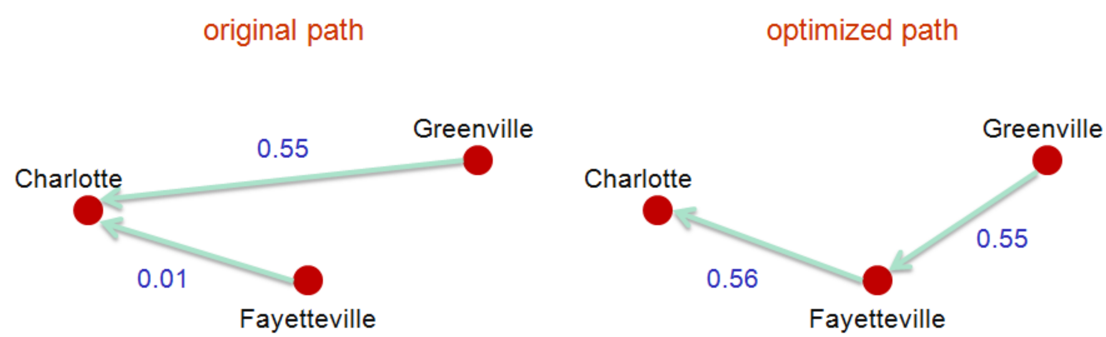

Figure 40: Inbound Milk Run from Computational Results

Figure 40 demonstrates an inbound milk run. End-of-lines Greenville and Fayetteville both dispatch trailers to breakbulk Charlotte, with a total freight volume of less than one trailerload. DLP determines that we should route the Greenville-Charlotte dispatch through Fayetteville to pick up Fayetteville-originating freight. A separate dispatch from Fayetteville to Charlotte is no longer needed.

Figure 41 demonstrates an outbound milk run. Breakbulk Memphis dispatches trailers to end-of-lines Evansville and Jackson, both with low load factors. DLP determines that we should load Jackson-bound freight also to the Memphis-Evansville trailer, route such trailer through Jackson where Jackson-bound freight is kicked off, and continue the dispatch on to the final destination Evansville. Combining the two

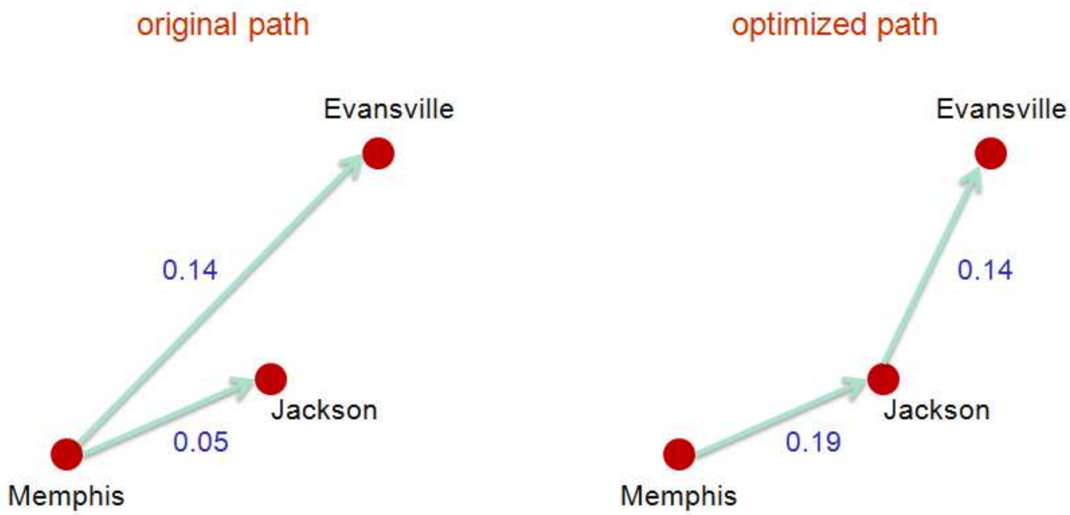

Figure 41: Outbound Milk Run from Computational Results

destinations into a single dispatch removes the need for a separate dispatch from Memphis to Jackson.

CHAPTER IV

STOCHASTIC LOAD PLAN DESIGN

4.1 Introduction

Load plan design models commonly represent origin-destination freight volumes using average demands derived from historical data; for example, an average weekly freight volume or an average daily freight volume might be used. The drawback of using an average is that it does not describe freight volume fluctuations. In this chapter, we investigate load plan design models that attempt to explicitly utilize information on freight volume uncertainty during planning. Our goal is to develop load plans that most cost-effectively deal with varying freight volumes and lead to the lowest expected cost.

Consider the small network presented in Figure 42 for an example of how a stochastic model can lead to a different optimal load plan than that from a deterministic model. The numbers above each arc represent the dispatch cost of a trailer, where L is a large, positive number.

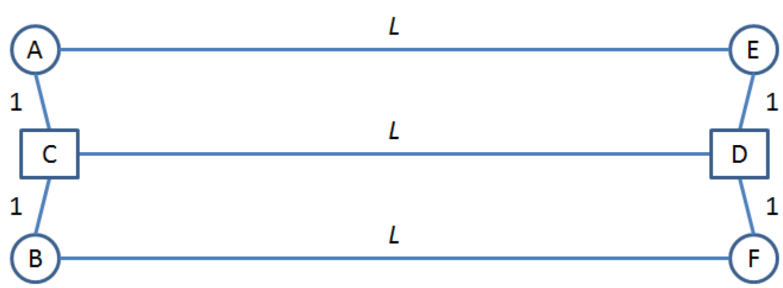

Figure 42: Example Network

Suppose that the network faces two origin-destination pair freight volumes, $A \rightarrow E$ and $B \rightarrow F$, each of which can be represented by a continuous random variable with a uniform distribution on $[1-\epsilon, 1+\epsilon]$, where volume is measured in fractional trailerloads, and $\epsilon < 0.5$ is a small, positive number. Suppose furthermore that the freight volumes on each pair are independent.

A deterministic model that uses average freight volumes sees both $A \to E$ and $B \to F$ freight as exactly one full trailerload. Therefore, there is no need for consolidation and the optimal load plan is to serve each demand using a direct trailer, as shown in Figure 43. The probability distributions above arcs represent the freight flow. The expected number of trailers dispatched on $A \to E$ and on $B \to F$ is

$$\mathbb{E}[\,\lceil X_1 \rceil\,] = 1.5, \qquad X_1 \sim \mathrm{U}(1-\epsilon, 1+\epsilon), \tag{23}$$

and the expected system cost of executing this load plan is

$$c_{\text{det}} = 1.5L + 1.5L = 3L$$

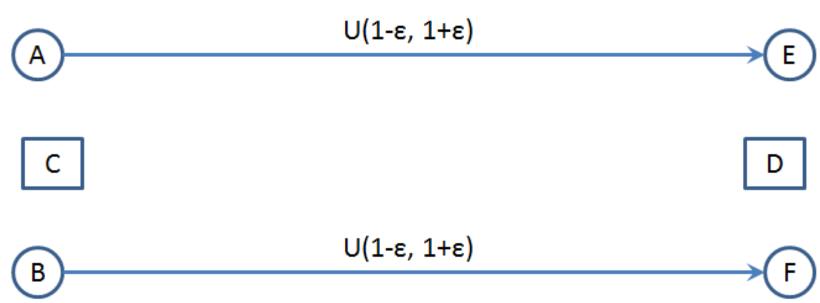

Figure 43: Freight Routing Decisions by Deterministic Load Plan Optimization

On the other hand, a stochastic model that considers freight volume uncertainty in load plan design would produce the solution shown in Figure 44. Both $A \to E$ and $B \to F$ freight is consolidated at breakbulk terminals C and D. This allows the possibility of a high value in one of the demands being offset by a low value in the other. Freight flow on $C \to D$ is the sum of two independent, uniformly distributed random variables, and is itself triangularly distributed with lower limit $2 - 2\epsilon$, upper limit $2 + 2\epsilon$, and mode 2.

Similar to (23), the expected number of trailers dispatched on $A \to C$, $B \to C$,

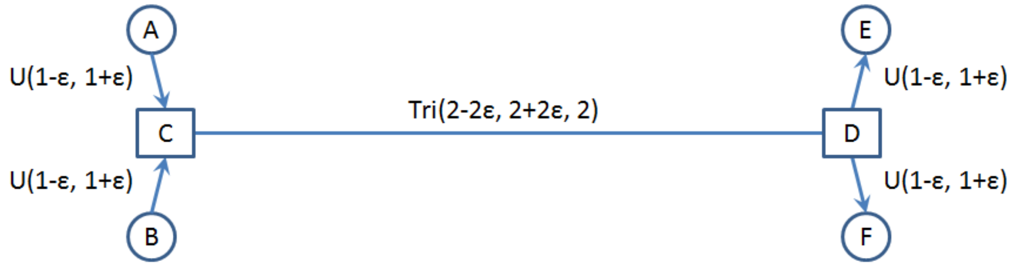

Figure 44: Freight Routing Decisions by Stochastic Load Plan Design

$D \to E$, and $D \to F$ is 1.5. The expected number of trailers dispatched on $C \to D$ is

$$\mathbb{E}[\lceil X_2 \rceil] = 2.5, \qquad X_2 \sim \mathrm{Tri}(2-2\epsilon, 2+2\epsilon, 2)$$

If we ignore cross-dock handling costs, which are typically dominated by transportation costs associated with moving trailers, the expected system cost of executing the second load plan is

$$c_{\text{stoch}} = 1.5 \times 4 + 2.5L = 2.5L + 6$$

For $L > 12$, $c_{\text{stoch}} = 2.5L + 6 < 3L = c_{\text{det}}$, and the stochastic load plan compares favorably.

The example demonstrates that consideration of demand stochasticity may allow load plans to be developed that reduce costs using the well-known concept of *risk pooling* (see [22] [12] for discussions, for example). In supply chain management, risk pooling effects incentivize serving customers from consolidated distribution facilities where one aggregates demand across locations in order to increase the likeliness of high demand from one customer being offset by low demand from another, hence reduces demand variability, decreases safety stock, and reduces average inventory. For load plan design, a similar effect provides further incentives for consolidation beyond those that are induced by scale economies in transportation cost alone.

In this chapter, we develop and present Sample Average Approximation (SAA) approaches for solving stochastic integer programming formulations of the load plan design problem with origin-destination pair demand uncertainty. In addition to applying the standard SAA approach, we also propose a modified version which, in order to correct the bias in the branch-and-bound search that results from using a sample, frequently computes an exact evaluation of the solution expected cost and a lower bound on this cost, to more accurately guide the search process.

The contributions of this research are two-fold. It is the first to study a stochastic service network design problem for LTL carriers, illustrating the importance of explicitly utilizing information on freight volume uncertainty during planning. Second, it demonstrates a scheme of using exact evaluations within solving SAA problems to improve the guidance of the search when it is not expensive to compute such information.

The remainder of this chapter is organized as follows. In Section 4.2, we review relevant literature. In Section 4.3, we discuss modeling issues and choices for the stochastic load plan design problem. Section 4.4 presents integer programming formulations of the load plan design problem and our solution approaches. Section 4.5 reports the results of computational studies conducted using data from a national LTL carrier.

4.2 Additional Related Literature

Stochastic service network design problems were first discussed in [12] and [7]. The papers demonstrate that plans created with explicit consideration of stochastic elements are more robust than those of traditional deterministic models.

The sample average approximation method was introduced in [10]. Theoretical and algorithmic issues related to stochastic integer programs are also discussed in [1]. Computational experiments of the SAA method are reported in [11] and [13], and a successful application to stochastic routing problems in [23].

4.3 Modeling Stochastic Load Plan Design

Solving realistically-sized instances of load plan design models developed using detailed time-space network representations is beyond the capability of the state-of-the-art integer programming solvers available today. A decomposition approach that uses exact optimization within heuristic search is proposed in [5]. Incorporating uncertainty into the model further increases the complexity of the resulting optimization problem. In this research, we focus on modeling and algorithm design for the stochastic aspect of the problem, and thus have chosen to make a number of simplifying assumptions to limit the problem size. Extending our approach to solve detailed, full-sized models using decomposition and heuristic search as proposed in [5] is left for future research.

Specifically, then, we make the following simplifying assumptions in this chapter:

1. Instead of working with the full linehaul network of a national LTL carrier, we use only a portion of the network, *e.g.*, the northwestern United States.

2. We use a planning horizon of a day, and assume that the freight pattern repeats daily.

3. We model freight routing decisions on a static network and ignore dispatch timing. Although we only consider service-feasible paths, we recognize that this may lead to an overestimation of consolidation opportunities. Note that this is similar to [9] which models time coarsely with a single node per day per terminal.

4. We model only transportation costs associated with moving trailers. Cross-dock handling costs are commonly dominated by transportation costs and are ignored.

5. We calculate trailer requirements based on weight only, not cube.

6. We do not model repositioning of trailers for balancing. This allows the recourse problem to be especially simple (a counting problem). Although we use this assumption, it is not conceptually difficult to extend our framework to model trailer balance by solving minimum cost flow repositioning problems during recourse, which is left for future research.

To make freight routing decisions, we model terminals and potential directs on a directed graph $D = (V, A)$, where V is the set of terminals and A is the set of potential directs connecting terminals. Associated with each direct $a \in A$ is a transit time and a cost c_a that respectively reflect how much time and money it takes the carrier to route a trailer from the origin terminal to the destination terminal of direct a.

Each origin-destination freight pair is modeled as a commodity. We assume a discrete uniform distribution for joint realizations of freight volumes of all commodities, *i.e.*, N_0 scenarios, each representing a complete realization of all origin-destination freight volumes, are equally likely to be observed. The set of scenarios can be constructed, for example, using historical freight volume demands on N_0 different days. Let K denote the set of commodities. For each commodity $k \in K$, let $\overline{w}_k$ denote the average weight of the total daily shipments from origin to destination, and w_k^n denote the weight in scenario n, $n \in \{1, \ldots, N_0\}$. Let C denote the capacity of a trailer. Our approach will use a path-based optimization model on D. For each commodity $k \in K$, we construct a set of possible freight paths, denoted by $P(k)$, where a freight path p is a sequence of directs, *i.e.*, $p = (a_1, \ldots, a_{n_p})$. Each path $p \in P(k)$ connects the origin terminal of k to the destination terminal of k. How commodity k is routed then simply becomes a question of choosing a path $p \in P(k)$. By using a path-based model, many practical constraints can easily be enforced, *e.g.*, the restriction that freight is handled at most two times is easily modeled by restricting the path generation step to find only such paths.

To generate a set of paths $P(k)$ for commodity k, we begin by enumerating all paths in D that connects from its origin to destination and involve at most two intermediate handlings. Up to h minimum cost paths with respect to total travel cost are then taken for further inspection (for some given value of h). Since we ignore time in this computation, we next determine which of these paths are service feasible. To do so, we first compute the minimum execution duration of each path by summing the travel times of the directs on the path and the required handling time at intermediate terminals. Consistent with [5] and [9], we assume 30 minutes of handling time for 1-day (overnight) services, and two hours for all other service standards. We then compare this minimum execution duration with the available transit time defined by the service standard, origin time, and due time. We model freight entering the linehaul network at 6 p.m. on the day of pickup, and it must reach its destination terminal by 8 a.m. on the day of delivery.

Recall that a traditional load plan specifies the unique direct that a shipment should take given its current terminal location and its ultimate destination. Hence, the structure of a load plan requires that the directs chosen for freight destined for terminal d must form a directed in-tree rooted at d, as depicted in Figure 45. Therefore, in our path-based approach when choosing paths for commodities, we must ensure that the set of paths chosen for all commodities are such that there is appropriate consistency of the paths selected for commodities with a common destination. We ensure this requirement using so-called path-continuation constraints. For example, suppose that freight originating in Athens, GA and destined for Columbus, OH uses path (Athens $\rightarrow$ Atlanta $\rightarrow$ Cincinnati $\rightarrow$ Columbus). Then freight originating in Atlanta, GA and destined for Columbus, OH must use path (Atlanta $\rightarrow$ Cincinnati $\rightarrow$ Columbus). In general, for any path $p = (a_1, \ldots, a_{n_p}) \in P(k)$ that consists of more than one direct, let $\mathsf{cont}(p) = (a_2, \ldots, a_{n_p})$ be a path for the commodity that originates at the destination terminal of direct a_1 and has the same destination as commodity k. Such path $\mathsf{cont}(p)$ is always generated when it is itself service-feasible. The in-tree structural property can then be ensured by allowing path p to be selected only if $\mathsf{cont}(p)$ is selected.

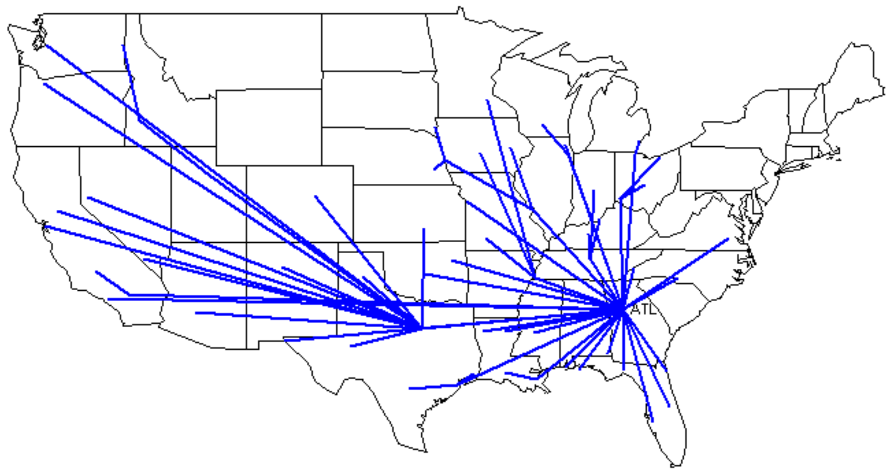

Figure 45: Freight Paths Form an In-Tree into a Destination Terminal

4.4 Load Plan Design Integer Programs

In this section, we present integer programming formulations of the load plan design problem and our solution approaches.

4.4.1 Deterministic Load Plan Optimization

The first formulation we present is the deterministic load plan optimization model, referred to as DetLPO. It has two sets of decision variables : x variables indicate whether commodity k uses path p, *i.e.*, $x_p \in \{0,1\}$, $\forall p \in P(k)$, $\forall k \in K$; and τ variables count the number of trailers that move on direct a, *i.e.*, $\tau_a \in \mathbb{Z}_+$, $\forall a \in A$.

minimize

$$\sum_{a \in A} c_a \tau_a \tag{24}$$

subject to

$$\sum_{p \in P(k)} x_p = 1 \quad \forall k \in K \tag{25}$$

$$x_p \leq x_{\mathsf{cont}(p)} \quad \forall p \in P(k),\ |p| \geq 2,\ \forall k \in K \tag{26}$$

$$\sum_{k \in K} \sum_{p \in P(k)\,:\,a \in p} \overline{w}_k x_p \leq \tau_a C \quad \forall a \in A \tag{27}$$

$$x_p \in \{0, 1\} \quad \forall p \in P(k),\ \forall k \in K$$

$$\tau_a \in \mathbb{Z}_+ \quad \forall a \in A$$

The objective (24) is to minimize total transportation costs, which are assumed to be linear in the number of trailers dispatched on each direct. Constraints (25) ensure that a path is chosen for each commodity. Path-continuation constraints (26) ensures the in-tree structural property of the load plan. Constraints (27) ensure that there are enough trailers dispatched on a direct arc to carry the freight assigned via the paths chosen.

4.4.2 Stochastic Load Plan Design

When freight volume is a random vector w, the stochastic load plan design (SLP) model seeks to minimize the expected total cost. We formulate SLP as a two-stage stochastic programming model. First-stage x variables define the load plan, while second-stage τ variables count the number of trailers required on each direct arc, for each possible value of w.

minimize

$$\mathbb{E}\left[\mathcal{C}(x, w)\right] \tag{28}$$

subject to

$$\sum_{p \in P(k)} x_p = 1 \quad \forall k \in K$$

$$x_p \leq x_{\mathsf{cont}(p)} \quad \forall p \in P(k),\ |p| \geq 2,\ \forall k \in K$$

$$x_p \in \{0,1\} \quad \forall p \in P(k),\ \forall k \in K$$

where $\mathcal{C}(x,w) =$

minimize

$$\sum_{a \in A} c_a \tau_a$$

subject to

$$\sum_{k \in K} \sum_{p \in P(k)\,:\,a \in p} w_k x_p \leq \tau_a C \quad \forall a \in A$$

$$\tau_a \in \mathbb{Z}_+ \quad \forall a \in A$$

Note that, for a fixed first-stage decision x and a particular realization of w, the second stage problem $\mathcal{C}(x,w)$ has a trivial solution which can be explicitly specified as follows:

$$\tau_a = \left\lceil \left(\sum_{k \in K} \sum_{p \in p(k): a \in p} w_k x_p \right) / C \right\rceil \quad \forall a \in A \tag{29}$$

With a finite number of scenarios, the expectation $\mathbb{E}\left[\mathcal{C}(x,w)\right]$ can be evaluated as the finite sum $\frac{1}{N_0}\sum_{n=1}^{N_0} \mathcal{C}(x,w^n)$, and SLP can be converted into a deterministic equivalent form, referred to as SLPDE, by introducing a different set of τ variables for each scenario to represent the trailer flows required on each arc.

minimize

$$\frac{1}{N_0} \sum_{n=1}^{N_0} \left(\sum_{a \in A} c_a \tau_a^n \right)$$

subject to

$$\sum_{p \in P(k)} x_p = 1 \quad \forall k \in K$$

$$x_p \leq x_{\mathsf{cont}(p)} \quad \forall p \in P(k),\ |p| \geq 2,\ \forall k \in K$$

$$\sum_{k \in K} \sum_{p \in P(k)\,:\,a \in p} w_k^n x_p \leq \tau_a^n C \quad \forall a \in A,\ \forall n \in \{1, \ldots, N_0\}$$

$$x_p \in \{0,1\} \quad \forall p \in P(k),\ \forall k \in K$$

$$\tau_a^n \in \mathbb{Z}_+ \quad \forall a \in A,\ \forall n \in \{1, \ldots, N_0\}$$

4.4.3 Sample Average Approximation

With a large number of scenarios, SLPDE is a large-scale integer program. Solving it directly is generally beyond the capability of the state-of-the-art integer programming solvers. To overcome this difficulty, Sample Average Approximation (SAA) approaches use Monte Carlo simulation to reduce the scenario set to a manageable size. A sample $\widehat{w}^1, \ldots, \widehat{w}^N$ of N ($< N_0$) realizations of w is generated, and the expected value function $\mathbb{E}\left[\mathcal{C}(x, w)\right]$ is approximated by the sample average function $\frac{1}{N}\sum_{n=1}^{N} \mathcal{C}(x, \widehat{w}^n)$. The obtained sample average approximation problem is then to

minimize

$$\frac{1}{N}\sum_{n=1}^{N}\left(\sum_{a \in A} c_a \tau_a^n\right) \tag{30}$$

subject to

$$\sum_{p \in P(k)} x_p = 1 \quad \forall k \in K$$

$$x_p \leq x_{\mathsf{cont}(p)} \quad \forall p \in P(k),\ |p| \geq 2,\ \forall k \in K$$

$$\sum_{k \in K}\ \sum_{p \in P(k)\,:\, a \in p} \widehat{w}_k^n x_p \leq \tau_a^n C \quad \forall a \in A,\ \forall n \in \{1, \ldots, N\}$$

$$x_p \in \{0, 1\} \quad \forall p \in P(k),\ \forall k \in K$$

$$\tau_a^n \in \mathbb{Z}_+ \quad \forall a \in A,\ \forall n \in \{1, \ldots, N\}$$

The SAA problem is then solved using a standard branch-and-bound approach, and its solution serves as a candidate solution to the true SLP problem. The sampling-optimization process is repeated M times with different samples to obtain candidate solutions along with statistical estimates of their optimality gaps. Specifically, suppose that by generating M independent samples and solving the associated SAA problems, we obtain optimal objective values $z^1, \ldots, z^M$ and candidate solutions $x^1, \ldots, x^M$. Let z^* denote the optimal objective value of the true SLP problem. We next describe techniques for estimating upper and lower bounds for z^*. The difference between them is the estimated optimality gap.

4.4.3.1 Upper Bound

For each candidate solution x, clearly the objective value $\mathbb{E}\left[\mathcal{C}(x,w)\right]$ is an upper bound for z^*. Using the result of (29), this expectation objective can be exactly evaluated as:

$$\frac{1}{N_0}\sum_{n=1}^{N_0}\sum_{a\in A}\left(c_a\left\lceil\left(\sum_{k\in K}\sum_{p\in p(k):a\in p} w_k^n x_p\right)/C\right\rceil\right) \tag{31}$$

We then choose the best solution among all the candidate solutions $x^1,\ldots,x^M$.

4.4.3.2 Lower Bound

The following lower bounding technique is developed in [13] and [14]. Suppose z^{SAA} is the optimal value obtained from solving an SAA problem. It is well-known that

$$\mathbb{E}[z^{\mathrm{SAA}}] \le z^*$$

Therefore, we can obtain a lower bound to z^* by estimating $\mathbb{E}[z^{\mathrm{SAA}}]$. Recall that $z^1,\ldots,z^M$ denote the optimal objective values of the M independent SAA problems. Then

$$\overline{z} = \frac{1}{M}\sum_{m=1}^{M} z^m$$

is an unbiased estimator of $\mathbb{E}[z^{\mathrm{SAA}}]$ and thus is a statistical lower bound to z^*. Furthermore, the variance of the above estimator (and therefore the variance of the gap estimator) can be estimated by

$$\sigma_{\overline{z}}^2 = \frac{1}{M(M-1)}\sum_{m=1}^{M}(z^m - \overline{z})^2$$

4.4.4 Sample Average Approximation with Exact Evaluations

The idea of incorporating exact evaluations in Sample Average Approximation is a natural one. Since each SAA problem is biased from using a sample, when it is not expensive to compute exact evaluations such as (31), one can frequently generate such information during solving an SAA problem and use it to improve the guidance of the branch-and-bound search.

Specifically, at each node of the branch-and-bound search tree, we have a potentially fractional x. Depending on whether x is integral or fractional, we can compute either the exact expected cost of a solution, or a lower bound, using all N_0 scenarios.

- When x is integral, the exact expected cost associated with it is

$$\frac{1}{N_0}\sum_{n=1}^{N_0}\sum_{a\in A}\left(c_a\left\lceil\left(\sum_{k\in K}\sum_{p\in p(k):a\in p} w_k^n x_p\right)/C\right\rceil\right) \tag{32}$$

- When x is fractional, an approximate lower bound at the node is obtained by relaxing the integrality constraint on trailer count variables

$$\frac{1}{N_0}\sum_{n=1}^{N_0}\sum_{a\in A}\left(c_a\cdot\left(\sum_{k\in K}\sum_{p\in p(k):a\in p} w_k^n x_p\right)/C\right) \tag{33}$$

  This is an approximate lower bound because the x values are obtained by solving the linear relaxation of a formulation that is biased by a sample.

A node is pruned if the lower bound (33) is larger than the cost (32) associated with the incumbent solution. We referred to this modified branch-and-bound procedure as the Sample Average Approximation with Exact Evaluations (SAAEE). Similar to SAA, we also repeat the sampling-optimization procedure M times and choose the best solution among all the M candidate solutions. An interesting question for future research is whether using exact evaluations during SAA solves reduces the required number of solves to achieve the same level of optimality as a standard SAA approach.

4.5 Computational Results

The algorithm was developed in C++ with CPLEX 11 as the Mixed Integer Program (MIP) solver, interfaced via ILOG Concert Technology. When solving DetLPO and SAA, we use an optimality tolerance of 0.01%. For SAAEE, we solve each sample for 2 hours. We perform $M = 10$ replications for SAA and SAAEE. All computational experiments were carried out on a system with a 2.66 GHz Intel Xeon processor and 4 GB of RAM.

Our test instances are generated from a portion of the linehaul network of a national LTL carrier. It consists of the northwestern United States including the states of Washington, Oregon, Idaho, and parts of California, Nevada, and Utah, with a total of 13 terminals (7 breakbulks and 6 end-of-lines), and 156 potential directs and 141 freight origin-destination pairs. See Figure 46 for an illustration.

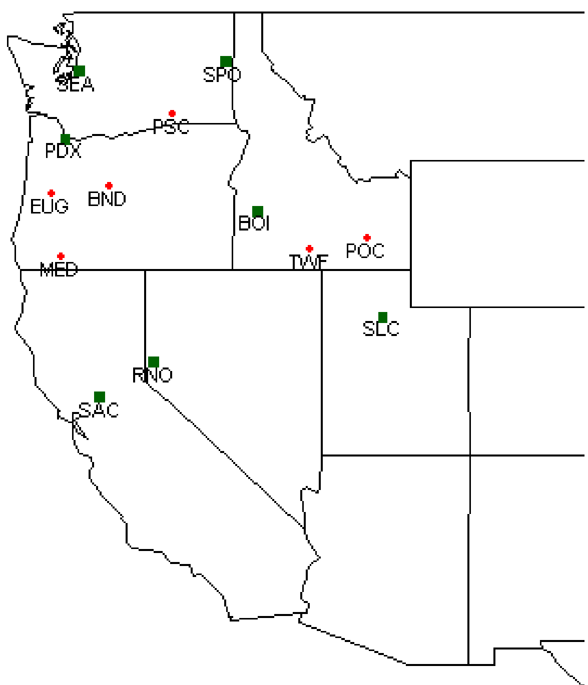

Figure 46: Portion of Linehaul Network for Computational Experiments

We create two instances with different levels of dispersion.

1. I_{low}: scenarios are drawn from $U(0.5\ \overline{w}, 1.5\ \overline{w})$; all commodities thus have the same standard-deviation-to-mean ratio of 0.288675.

2. I_{high}: scenarios are drawn from $U(0, 2\overline{w})$; all commodities thus have the same standard-deviation-to-mean ratio of 0.577350.

In Table 13, we compare the true expectation objective (31) associated with the load plans obtained by solving DetLPO, SAA, and SAAEE. We also report the optimality gaps and their standard deviations. All values are measured in percentages relative to the objective value of DetLPO. In Table 14, we report computation times. The results show that stochastic optimization approaches produce more robust load plans than the deterministic model. Furthermore, the benefit is larger for the more dispersed setting. Finally, we see that, as a result of using exact evaluations to more accurately guide the branch-and-bound search process, SAAEE compares slightly favorably against the standard SAA approach.

Table 13: Comparison of Load Plan Costs

Instance	DetLPO	SAA			SAAEE		
		obj value	opt gap	σ_{gap}	obj value	opt gap	σ_{gap}
I_{low}	100	95.25	0.86	0.004	95.17	0.78	0.004
I_{high}	100	94.29	1.11	0.007	94.16	0.98	0.007

Table 14: Computation Times (Seconds)

Instance	DetLPO	SAA, average per sample	SAAEE, average per sample
I_{low}	3	3747	7200 (limited)
I_{high}	3	4579	7200 (limited)

We next analyze the differences among the load plans obtained from each approach. In Table 15, we report the average number of handlings per commodity. In Table 16, we report the average number of commodities moved on a direct. In Table 17, we report the average length of haul per dispatched trailer. We see an increase in the first two statistics as we move from the deterministic model to the stochastic models, and a decrease in the average length of haul. This is expected because stochastic models are incentivized to consolidate freight beyond those that are induced by scale economies in transportation cost alone.

Table 15: Average Number of Handlings per Commodity

Instance	DetLPO	SAA	SAAEE
I_{low}	0.766	0.851	0.865
I_{high}	0.766	0.839	0.850

Table 16: Average Number of Commodities on a Direct

Instance	DetLPO	SAA	SAAEE
I_{low}	5.533	6.525	6.575
I_{high}	5.533	6.310	6.524

Lastly, we demonstrate the value of using exact evaluations to improve the guidance of the branch-and-bound search. Table 18 is generated based on all feasible solutions that we encounter during SAAEE solves and that improve the then-current incumbent solution based on either SAA objective (30) or exact evaluation (32). These solutions are classified into three categories:

1. *False Improving*: It improves the incumbent solution based on SAA objective (30), but turns out not to be an improving solution based on exact evaluation (32)

2. *Missed Improving*: It does not improve the incumbent solution based on SAA objective (30), but turns out to be an improving solution based on exact evaluation (32); in other words, this solution would have been lost if we do not compute exact evaluations within the branch-and-bound search

3. *Consistent Improving*: Both criteria indicate that it improves the incumbent solution

We see that for a significant portion of the solutions we encountered, using SAA

Table 17: Average Length of Haul per Dispatched Trailer

Instance	DetLPO	SAA	SAAEE
I_{low}	311.2	298.1	297.3
I_{high}	311.7	300.2	297.1

Table 18: Using SAA Objective v.s. Using Exact Evaluation

Instance	Consistent Improving	Missed Improving	False Improving
I_{low}	32.7%	22.1%	45.2%
I_{high}	33.6%	23.8%	42.6%

objective (30) renders a different conclusion than using exact evaluation (32). Incorporating such exact evaluations in the branch-and-bound process thus helps improve the guidance of the search. We recognize, however, that the computational results show only modest benefit of this approach, partly because standard SAA approaches already overcome the drawback that results from using a sample by drawing large enough samples and by repeating the sampling-optimization procedure many times. We believe more research effort will be necessary in the future to improve our approach and to reach a conclusion about the benefit of this type of approach.

CHAPTER V

CONCLUSIONS AND FUTURE RESEARCH

Load plan design technologies, such as dynamic load planning described in Chapter 3 and stochastic load plan design studied in Chapter 4, have to make simplifying assumptions to become computationally tractable. In Chapter 2, we designed technologies that more accurately capture key operation of LTL carriers and estimate the operational execution costs of a load plan. The next challenge is to integrate load plan design and execution cost estimation technologies. For example, load plan design models could be extended to incorporate the building of driver tours to cover planned dispatches.

In stochastic load plan design, we have chosen to make rather strong simplifying assumptions and to focus on the stochastic aspects of modeling and algorithm design. We believe that there is potential to extend our approach to solve detailed, full-sized models by using decomposition and heuristic search as proposed in [5].

Finally, there is potential to improve the stochastic load plan design models by using dynamic load planning adjustments as a *recourse* strategy, allowing actual operations to be responsive to freight volumes. For example, we can modify SLP by letting x variables now indicate selection of *nominal* paths, and introducing additional second-stage v variables to indicate selection of *actual* paths that adjust the nominal paths based on demand realizations. For each $k \in K$ and $p \in P(k)$, let $Q(k,p)$ denote a set of potential DLP-adjusted paths based on p. For each $k \in K$, let $Q(k) = \bigcup_{p \in P(k)} Q(k,p)$.

The stochastic load plan design model with DLP recourses is then to minimize

$$\mathbb{E}\left[\widetilde{\mathcal{C}}(x,w)\right]$$

subject to

$$\sum_{p \in P(k)} x_p = 1 \quad \forall k \in K$$

$$x_p \leq x_{\mathsf{cont}(p)} \quad \forall p \in P(k),\ |p| \geq 2,\ \forall k \in K$$

$$x_p \in \{0,1\} \quad \forall p \in P(k),\ \forall k \in K$$

where $\widetilde{\mathcal{C}}(x,w) =$

minimize

$$\sum_{a \in A} c_a \tau_a$$

subject to

$$\sum_{q \in Q(k)} v_q = 1 \quad \forall k \in K \tag{34}$$

$$v_q \leq \sum_{p \in P(k)\,:\,q \in Q(k,p)} x_p \quad \forall q \in Q(k),\ \forall k \in K \tag{35}$$

$$v_q \leq v_{\mathsf{cont}(q)} \quad \forall q \in Q(k),\ |q| \geq 2,\ \forall k \in K \tag{36}$$

$$\sum_{k \in K} \sum_{q \in Q(k)\,:\,a \in q} w_k v_q \leq \tau_a C \quad \forall a \in A \tag{37}$$

$$v_q \in \{0,1\} \quad \forall q \in Q(k),\ \forall k \in K$$

$$\tau_a \in \mathbb{Z}_+ \quad \forall a \in A$$

Constraints (34) ensure that an actual, demand-responsive path is chosen for each commodity. Constraints (35) ensure that an actual path can only be chosen when the corresponding nominal path is chosen. Path-continuation constraints (36) ensures that the demand-responsive load plan still maintains the in-tree structural property. Constraints (37) ensure that there are enough trailers on an arc to carry the freight assigned to the arc via the actual paths chosen based on freight volume realizations.

REFERENCES

[1] Ahmed, S. and Shapiro, A., "The sample average approximation method for stochastic programs with integer recourse," *Optimization-online*, 2002.

[2] Cohn, A., Root, S., Wang, A., and Mohr, D., "Integration of the load matching and routing problem with equipment balancing for small package carriers," *Transportation Science*, vol. 41, pp. 238–252, 2007.

[3] Crainic, T., "Service network design in freight transportation," *European Journal of Operations Research*, vol. 122, no. 2, pp. 272–288, 2000.

[4] Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F., "Time constrained routing and scheduling," in *Network Routing* (Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., eds.), vol. 8 of *Handbooks in Operations Research and Management Science*, pp. 35–139, Amsterdam: Elsevier, 1995.

[5] Erera, A., Hewitt, M., Savelsbergh, M., and Zhang, Y., "Improved load plan design through integer programming based local search," 2009. submitted for publication.

[6] Erera, A., Karacík, B., and Savelsbergh, M., "A dynamic driver management scheme for less-than-truckload carriers," *Comput. Oper. Res.*, vol. 35, no. 11, pp. 3397–3411, 2008.

[7] Hoff, A., Lium, A.-G., Løkketangen, A., and Crainic, T., "A metaheuristic for stochastic service network design," *Journal of Heuristics*, 2009.

[8] Irnich, S. and Desaulnier, G., "Shortest path problems with resource constraints," in *Column Generation* (Desaulniers, G., Desrosiers, J., and Solomon, M. M., eds.), pp. 33–65, Springer, 2005.

[9] Jarrah, A. I., Johnson, E., and Neubert, L. C., "Large-scale, less-than-truckload service network design," *Operations Research*, vol. 57, no. 3, pp. 609–625, 2009.

[10] Kleywegt, A. J., Shapiro, A., and de Mello, T. H., "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[11] Linderoth, J., Shapiro, A., and Wright, S., "The empirical behavior of sampling methods for stochastic programming," *Annals of Operations Research*, vol. 142, pp. 215–241, 2006.

[12] Lium, A.-G., Crainic, T. G., and Wallace, S. W., "A study of demand stochasticity in service network design," *Transportation Science*, vol. 43, no. 2, pp. 144–157, 2009.

[13] Mak, W.-K., Morton, D. P., and Wood, R. K., "Monte carlo bounding techniques for determining solution quality in stochastic programs," *Operations Research Letters*, vol. 24, no. 1-2, pp. 47 – 56, 1999.

[14] Norkin, V. I., Pflug, G. C., and Ruszczyński, A., "A branch and bound method for stochastic global optimization," *Math. Program.*, vol. 83, no. 3, pp. 425–450, 1998.

[15] Pedersen, M. B., Crainic, T. G., and Madsen, O. B. G., "Models and tabu search metaheuristics for service network design with asset-balance requirements," *Transportation Science*, vol. 43, pp. 158–177, May 2009.

[16] Powell, W. B., "A local improvement heuristic for the design of less-than-truckload motor carrier networks," *Transportation Science*, vol. 20, no. 4, pp. 246–257, 1986.

[17] Powell, W. B. and Koskosidis, I. A., "Shipment routing algorithms with tree constraints," *Transportation Science*, vol. 26, no. 3, pp. 230–245, 1992.

[18] Powell, W. B. and Sheffi, Y., "The load planning problem of motor carriers: Problem description and proposed solution approach," *Transportation Research–A*, vol. 17A, no. 6, pp. 471–480, 1983.

[19] Powell, W. B. and Sheffi, Y., "Design and implementation of an interactive optimization system for network design in the motor carrier industry," *Operations Research*, vol. 37, pp. 12–29, 1989.

[20] Powell, W. and Farvolden, J., "Subgradient methods for the service network design problem," *Transportation Science*, vol. 28, pp. 256–272, 1994.

[21] Root, S. and Cohn, A., "A novel modeling approach for express package carrier planning," *Naval Research Logistics*, vol. 55, pp. 670–683, 2008.

[22] Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E., *Designing and Managing the Supply Chain*. McGraw-Hill/Irwin, second ed., 2002.

[23] Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., and Shapiro, A., "The sample average approximation method applied to stochastic routing problems: A computational study," *Comput. Optim. Appl.*, vol. 24, no. 2-3, pp. 289–333, 2003.

[24] Wieberneit, N., "Service network design for freight transportation: a review," *OR Spectrum*, vol. 30, pp. 77–112, Jan 2008.