DYNAMIC DECISION SUPPORT
FOR REGIONAL LTL CARRIERS

A Thesis
Presented to
The Academic Faculty

by

Prashant Warier

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Industrial and Systems Engineering

School of Industrial and Systems Engineering
Georgia Institute of Technology
August 2007

DYNAMIC DECISION SUPPORT FOR REGIONAL LTL CARRIERS

Approved by:

Martin W. P. Savelsbergh, Co-advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Alan L. Erera, Co-advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Corne Aantjes
Managing Partner
ORTEC

Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Chelsea C. White, III
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 6 April 2007

To my parents and my brother,

Thanks for all your support and love.

ACKNOWLEDGEMENTS

This has been a long journey! I learned a lot professionally and personally during the last few years as a PhD student at Georgia Tech. It feels great to be ending this phase of life but it also feels nostalgic. I take this opportunity to express my sincere gratitude to everyone who supported me during this journey.

First, I would like to thank my advisors, Prof. Martin Savelsbergh and Prof. Alan Erera without whose guidance and support, this thesis would not have been possible. They have motivated me and guided me through the entire PhD process. They were very understanding and patient in every stage of my graduate life, more so during the bad times. I am greatly honored to have advisors like them.

My gratitude also goes to ORTEC, who have financially supported this research. I thank Corne Aantjes at ORTEC, who provided lot of industry insights and channeled my efforts in the right direction. Thanks to Averitt Express for providing me with data for computational studies.

I want to express my gratitude to members of my thesis committee, Prof. Chelsea C. White III, and Prof. Joel Sokol, for their valuable feedback and for their time.

I also want to thank some of my friends at Georgia Tech whose presence has made Atlanta feel like a home away from home. Nisha Rajagopalan, Vandana Mohan, Dhiraj Ratnani, Chandrasekhar Natarajan, Parmjeet Singh, Chinmay Bhide, Manav Sheoran, Rohan Shah and Sowmya Shriraghavan : thanks for your friendship and your support. I want to thank my friends in ISyE, Vivek Gupta, Rajeev Namboothiri and Kapil Gupta for their company during the long days and nights at school.

I wish to thank my entire extended family for providing a loving environment for me. My cousins Shruti and Neethi Rajagopalan were particularly supportive.

Lastly, and most importantly, I wish to thank my parents, Pratap Warier and Anandavalli Warier, and my brother Aswin Warier. They raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis focuses on decision support for regional LTL carriers. The basic operating characteristics of regional LTL carriers are similar to those of national LTL carriers, i.e., they operate linehaul networks with satellites, breakbulks, and relays to consolidate freight so as to be able to cost-effectively serve their customers. However, there are also key differences. Most importantly, because the area covered by a regional carrier is smaller, a regional carrier handles less freight (sometimes significantly less) and therefore typically has fewer consolidation opportunities, which results in higher handling and transportation costs per unit of freight. Consequently, competing with national carriers on price is difficult. Therefore, to gain or maintain market share, regional carriers have to provide better service. To be able to provide better service, regional carriers have to be more dynamic, e.g., they have to be able to deviate from their load plan when appropriate, which creates challenges for decision makers. Regional carriers have a load plan, but do not adhere to it as strictly as national carriers. Regional carriers tend to experience more freight volume variability than the national carriers experience, which is another reason that operations have to be more dynamic. Fewer opportunities for consolidation and higher freight volume variability lead to lower utilization of trailers and hence an increase in the number of drivers required. On the other hand, regional carriers have the advantage, due to shorter distances, that it is easier to get their drivers back to their domicile at the end of a duty, thus reducing *lay over* costs.

Regional carriers deliver about 60% of their shipments within a day and almost all of their shipments within two days. Furthermore, most drivers get back to their domicile at the end of each day. Therefore, the focus of the thesis is the *development of effective and efficient decision models supporting daily operations of a regional LTL carriers.* These decision models should provide excellent service at low cost by dynamically constructing freight-flow plans and dynamically assigning drivers to loads.

This thesis presents an effective solution approach based on two optimization models: a *dynamic load planning* model and a *driver assignment* model. The dynamic load planning model consists of two parts: an integer program to generate the best paths for daily origin-destination freight volumes and an integer program to pack freight into trailers and trailers into loads, and to determine dispatch times for these loads. Techniques to efficiently solve these integer program solution are discussed in detail. The driver assignment model is solved in multiple stages, each stage requiring the solution of a set packing models in which columns represent driver duties. Each stages determines admissible driver duties. The quality and efficiency of the solution approach are demonstrated through a computational study with real-life data from one of the largest regional LTL carriers in the country.

An important "technique" for reducing driver requirements is the use of *meet-and-turn* operations. A basic meet-and-turn operation involves two drivers meeting at a location in between terminals and exchange trucks. A parking lot or a rest area suffices as a meet-and-turn location. This ensures that drivers return to the terminal where they started. More sophisticated meet-and-turn operations also exist, often called *drop and hook* operations. In this case, drivers do not exchange trucks, but one of their trailers. The motivation in this case is not to get drivers back to their domicile, but to reduce load-miles. The thesis presents analytical results quantifying the maximum benefits of using meet and turn operations and optimization techniques for identifying profitable meet-and-turn opportunities.

CHAPTER I

INTRODUCTION

The transportation industry is one of the largest industries in the U.S.; more than 10% of jobs are in transportation and the industry accounts for 6% of the GDP. The trucking industry represents the largest portion, accounting for about 3.5% of the GDP, and is vital for the U.S. economy. The trucking industry has two sectors: Truckload (TL) and Less-Than-Truckload (LTL). Truckload trucking accounts for the major share of revenues, with about 70% of the market being truckload transportation. Truckload carriers move freight for customers with enough freight to fill up an entire truck. Less-Than-Truckload carriers serve businesses that ship quantities ranging from 150 lbs to 10,000 lbs, i.e., less–than–truckload quantities.

To be economically viable, LTL carriers have to consolidate shipments into truckloads. Therefore, LTL carriers pick up shipments from various shippers in a relatively small geographical area, say a city, and bring them to a terminal serving the area, referred to as either a *satellite* terminal or *end-of-line* terminal. These satellite terminals serve as sorting centers and loading facilities for outbound and inbound freight. As there usually is not enough freight at a satellite terminal to build full truckloads to satellite terminals serving other areas, a second level of consolidation is introduced in the system. Outbound freight from a satellite is sent to a *breakbulk* terminal that consolidates freight from different satellite terminals. Breakbulk terminals do handle enough freight to build and dispatch cost efficient loads, i.e., loads that completely or almost completely fill up two trailers (or pups). The loads dispatched at a breakbulk are either destined for another breakbulk terminal or for a satellite terminal. Loads dispatched from a breakbulk to a satellite terminal are called *direct loads*, because they do are not handled at another breakbulk. Even though direct loads offer advantages, the majority of shipments travel from an origin satellite to an origin breakbulk, then to a destination breakbulk and finally to a destination satellite. This hub–and–spoke network is referred to as the (main) linehaul network. When the distance between the origin and destination breakbulk of a load is too long for a single driver to cover in the driving hours allowed by Department of Transportation regulations, intermediate stops are introduced at so–called *relay* terminals. Usually, relay terminals are breakbulk terminals on the path from the origin breakbulk to the destination breakbulk. At a relay terminal, the load is taken over by another driver to ensure continuity and high service levels. Since shipments are not unloaded and loaded at the relay terminals, the path of a shipment is identified by the sequence of terminals where the shipment is handled. The routing of shipments over the linehaul network is prescribed by the *load plan*. The operations of an LTL network is shown graphically in Fig 1 through Fig 7.

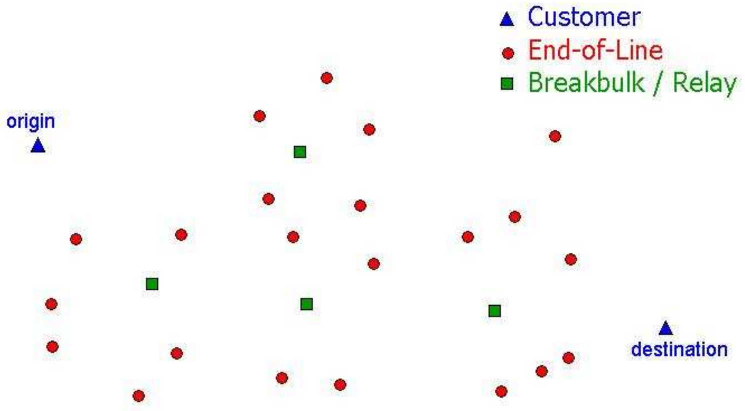

Figure 1: LTL operations

LTL carriers prefer direct loads, because a direct load reduces handling costs as well as handling time, which in turn reduces the total time taken for a shipment to

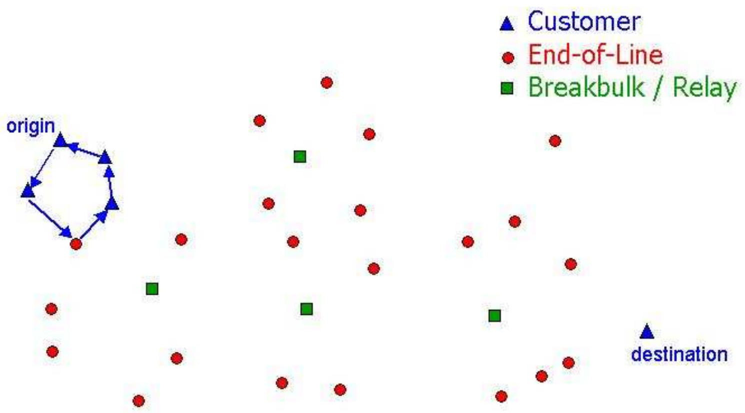

Figure 2: LTL operations

reach it's destination thus improving service. Physically, a direct load follows the same path as a regular load, so as to allow driver changes at relay terminals, but it is not handled until it reaches the destination satellite. Hence, direct loads do not decrease transportation costs. Direct loads from an origin satellite to a destination breakbulk are possible, but happen less frequently as it is less likely that enough freight accumulates quickly to build a direct load. Furthermore, if trailers at a satellite are used for the local pickup and delivery operation, then these trailers may be sent on to the outbound breakbulk without unloading and loading at the satellite terminal, which reduces handling cost and may improve service.

A load typically consists of two trailers, although vans are also used. The reason for using trailers as opposed to vans is that they fill up more quickly and may thus be dispatched earlier which will likely improve service. As trailers move in pairs, trailers need to be matched to form a load. This process is referred to as *pup-matching*. Consider three breakbulk terminals A,B,C and paths A-B and path A-C both using D as a relay (see Figure 8). When there is enough freight to fill a trailer for dispatch from A to B and for dispatch from A to C, the terminal manager at A may decide to

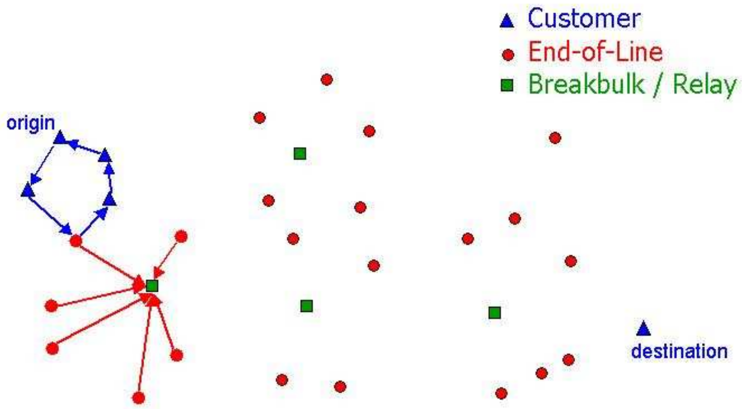

Figure 3: LTL operations

build a load for dispatch to D with one trailer destined for B and one trailer destined for C. Both these trailers will then wait at D until they can be paired up with trailers going to B and C, respectively.

So far our discussion has focused primarily on freight and how freight moves through the linehaul network. Of course freight cannot move without drivers and driver management is a crucial aspect of an efficient and effective linehaul system. An important concept that relates to drivers is that of a *meet and turn.* The basic meet-and-turn operation is shown in Fig 9. Meet and turn locations are points in between terminals (in this case A & B) where drivers meet and exchange trailers. A parking lot or a rest area suffices as a meet-and-turn location. Drivers would start from A and B heading for the meet-and-turn, exchange their loads at the meet-and-turn and return to their *domiciles.* This ensures that the loads make it on time at A and B, but also ensures that drivers get back to their domiciles.

Meet-and-turns are also used to perform what is called *drop and hook* operations. In this situation, drivers will start from A and B, both containing a trailer each for C and D. They meet, exchange one of their trailers and one of them goes to C and

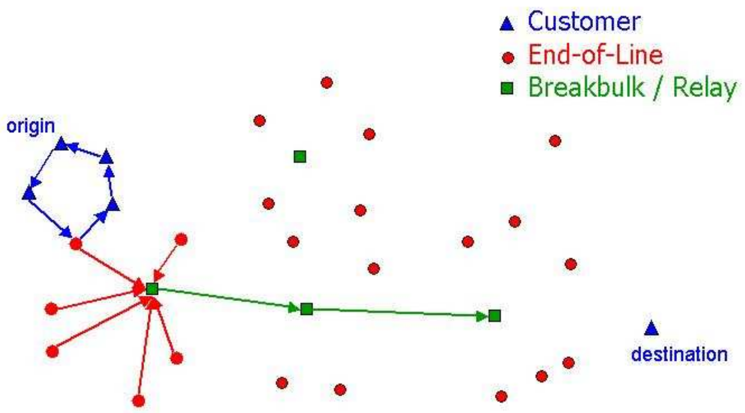

Figure 4: LTL operations

the other goes to D. Obviously, the motivation in this case is not to get drivers back to their domicile, but to reduce load-miles. A drop and hook operation is similar to the pup matching operations performed at breakbulks.

This thesis focuses on decision support for regional LTL carriers. The basic operating characteristics of a regional LTL carrier are similar to those of a national LTL carrier, but there are, sometimes subtle, differences. Regional LTL carriers also operate linehaul networks with satellites and breakbulks. However, since the operating region of a regional LTL carrier is smaller, there are far fewer relays. Clearly, regional carriers have to compete with national carriers, which has resulted in some important differences between regional and national carriers. National carriers, being larger, handle more freight (sometimes significantly more) and therefore typically have more consolidation opportunities, resulting in lower handling and transportation costs per unit of freight. Hence, it will be difficult for regional carriers to compete with national carriers on price. Therefore, regional carriers tend to compete on service. To be able to provide better service, regional carriers have to be more dynamic, e.g., they have to be able to deviate from the load plan when appropriate, which creates

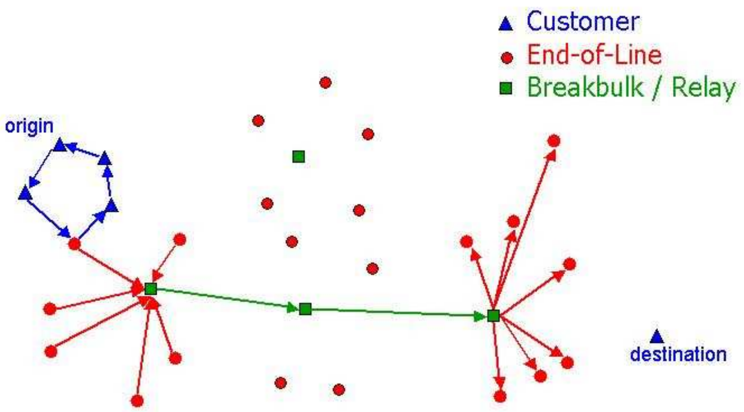

Figure 5: LTL operations

challenges for decision makers. Regional carriers do have a load plan in place, but they do not adhere to it as strictly as the national carriers. Regional carriers tend to experience more freight volume variability than the national carriers experience, which is another reason that operations have to be more dynamic. Regional carriers do have some advantages. Because of shorter driving distances, it is usually easier for regional carriers to get the drivers back to their domicile at the end of a duty. If a driver has to *lay over* for a night at another location, the LTL company incurs a cost as it has to pay the driver for the stay there. Regional carriers therefore tend to use meet-and-turns more frequently than national carriers. Most of the regional LTL carriers are non-union companies. Being non-union has advantages. Far fewer limitations exist on the use of drivers. As a result, the carrier can pretty much meet any need at any time of the day or weekend without incurring a substantial cost penalty.

The regional LTL industry has changed substantially over the years. Carriers once limited to hauling general freight within a few hundred miles are lengthening their lanes and offering expedited delivery, warehousing and logistics services. In

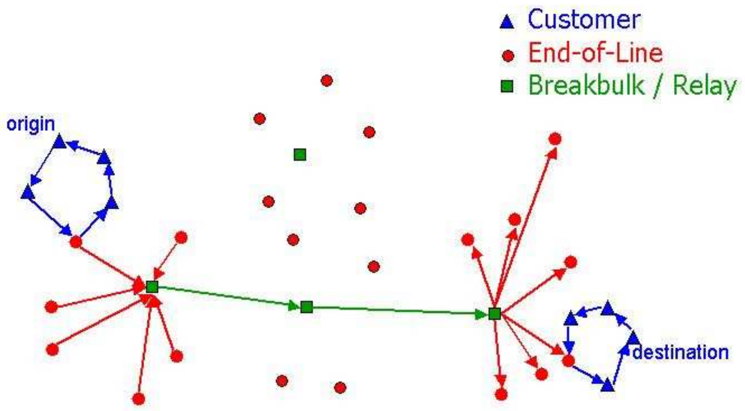

Figure 6: LTL operations

the meantime, their big national competitors increasingly are pushing their way into what once were purely regional markets. Customers want time-defined transit for specific products and weights. Shippers do not care which company offers it or by which mode it goes. Trucking companies of all shapes and sizes are responding by attempting to reconstruct and recast themselves. Driving the change is a combination of technology, a shift in production to Asia and economies of scale that allow big third-party logistics providers and national carriers to move into traditional regional LTL markets. Industry lines have blurred to the point where traditional long-haul carriers like Yellow Roadway now delivers 40 percent of its freight in two days or less. Given the infrastructure of the regional LTLs, with fleet and human resources already in place, new services are generally not capital intensive. To compete with these national carriers, which enjoy economies of scale, regional carriers have to provide better service at lower costs. The only way they can beat the national carriers is on service. This results typically in lower utilization of trailers and hence more drivers used.

Regional carriers deliver about 60% of their shipments within a day and most of

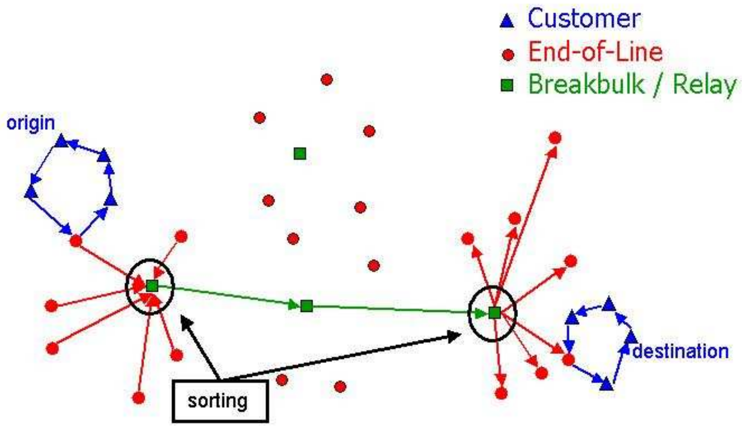

Figure 7: LTL operations

their shipments within two days. Furthermore, most drivers get back to their domicile at the end of the day. Because of these freight and operational characteristics, it seems reasonable and appropriate to consider dynamic daily planning. This is, therefore, the focus of the thesis: *develop effective and efficient decision models that support daily operations of a regional LTL carrier.* These decision models aim to provide better service while reducing costs by dynamically constructing freight-flow plans and dynamically assigning drivers to loads. As such these models focus on plans with less costs, fewer drivers, fewer miles, and higher load factors.

The daily decisions that need to be made by a regional LTL carrier are:

- How to route shipments through the network?
- How combine shipments into trailers and loads?
- When to dispatch loads?
- How to build driver duties?

These daily decisions suggest a natural hierarchical approach.

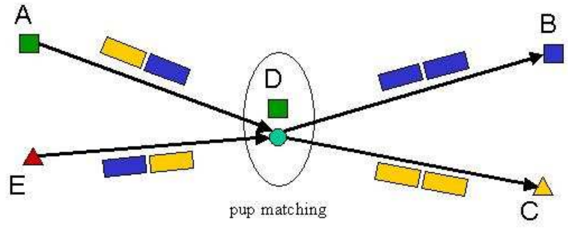

Figure 8: Pup Matching

In the first phase, we identify low-cost paths for each shipment meeting service requirements and resulting in high load factors. This phase is called the Freight Routing phase. In the second phase, given the shipment paths, we combine shipments into effectively packed loads that meet service requirements and determine feasible dispatch times for the loads. This phase is called the Trailer Assignment and Dispatch Timing phase. Both phases are discussed in detail in Chapter 2. At the end of the two phases, we have paths for the shipments, we know how shipments are combined into loads, and we have feasible dispatch times for the loads. *For instances obtained from a regional LTL carrier, we are able, on average, to increase the load factor by 10% while decreasing the total costs by 10%.*

In the third phase, we assign drivers to the loads that were created so as to minimize the number of drivers used while executing the loads within their feasible dispatch window. Driver duties have to satisfy hours of service constraints and company rules concerning their return to domicile. Some drivers have to return to their domicile every day while others have to return to their domicile every other day. This

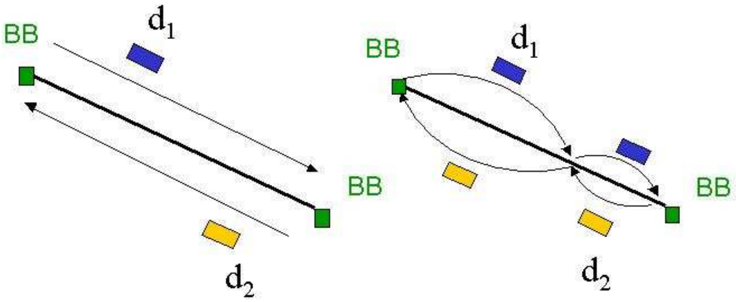

Figure 9: meet-and-turn operation

phase is called the Driver Assignment phase and is described in more detail in Chapter 3. *For instances obtained from a regional LTL carrier, we are able, on average, to reduce the number of drivers by 20% and the amount of empty travel by 9%.*

Effective use of meet-and-turns was crucial to reducing the number of drivers required to serve a set of loads. In Chapter 4, we study the value of meet-and-turns analytically. We develop bounds on the benefits that can be gained from using meet-and-turns. We show that in the best case the number of drivers can be reduced by 50% when considering a network consisting of a single leg. We extend this result to networks with multiple legs. We characterize how length of a leg and hours of service constraints affect the best case. We develop and provide performance guarantees for driver assignment heuristics exploiting meet-and-turns.

Summarizing, the main contributions of this thesis are the design and implementation of a set of effect and efficient decision models to support dynamic operations at a regional LTL carrier and an theoretical analysis of the value of meet-and-turns in driver management for LTL carriers.

CHAPTER II

FREIGHT ROUTING AND TRAILER ASSIGNMENT

2.1 Introduction

As mentioned in the previous chapter, regional LTL carriers often must adjust operating plans on a day-to-day basis in order to provide high levels of customer service with reasonable operating costs. This is quite different from the relatively static service network designs that are operated by national LTL carriers. The dynamic nature of operations at regional carriers makes the availability of effective decision support tools for daily operational planning activities very important.

Presently, most regional LTL carriers operate with fixed load plans and make minor changes in response to daily freight flow volumes, where such changes are determined in an ad hoc manner and at the discretion of local terminal managers at satellites and breakbulks. This process generally leads to freight meeting its service commitments, but, since decisions are based on local information only, may result in the use of more trailers and drivers than are actually needed to move the freight. Regional LTL carriers need centralized planning tools which can suggest how to build loads and route them through the system with assigned drivers, based on system-wide freight volume information. Such decision technology would not only be useful for managing daily demand fluctuations, but also would facilitate planning given seasonal variations in freight flow volumes.

In this dissertation, we decompose the centralized daily planning problem into two primary components. The first part, discussed in detail in this chapter, determines a set of loaded dispatches with dispatch time windows for a 24-hour planning period, such that all newly arriving freight and existing freight in the system is dispatched feasibly with respect to service commitments. The second part, discussed in detail in Chapter 3, determines a cost-effective assignment of drivers to the set of loaded dispatches.

2.2 Input Requirements for Daily Load Planning

The methods developed in this chapter are designed to be executed at the completion of daily freight pickup activities at all satellite terminals, such that the carrier now knows all new freight entering the system. For a typical regional LTL carrier, pickup and delivery operations are conducted during the day and the majority of linehaul operations occur during the evening and night. For shipments with a next-day service commitment, overnight dispatches allow the freight to arrive the following morning at the destination satellite for distribution via a delivery tour.

The 24-hour planning horizon that we use thus begins at a fixed time τ each day, where time τ is such that the regional LTL carrier has complete information regarding new freight entering the system on that day. In addition to this new freight, we assume also that the carrier has complete information on all freight in the linehaul network that has not yet arrived at its destination satellite terminal. For example, shipments with a two-day service commitment may only have reached an intermediate breakbulk terminal by time τ on day 1.

The following input data is assumed to be available:

Network Structure:

- Terminals (names, locations, types)
- Meet-and-turn locations
- Distances between all locations
- Travel times between all locations

Freight Information:

- Origin-destination flow volume (total trailerloads, earliest ready time, latest allowable arrival time)

We assume that the flow from an origin to a destination for a given day is comprised of potentially two components: new freight originating from pickup tours on that day, and existing freight that terminated at this terminal the previous day en route to its ultimate destination. Note that the latter freight type is only possible at breakbulk terminals, which also often have pickup and delivery operations like satellites. For a given origin-destination pair, we assume a single ready time, the earliest time that outbound freight for this pair is ready to be dispatched, and a single latest allowable arrival time computed by the service requirement for this lane.

Furthermore, another assumption of our approach is that all freight in the system (whether new or existing) has an overnight destination specified *a priori*. Thus, if new freight arriving has a two-day or three-day service commitment, we assume that the carrier has predetermined the breakbulk terminals (if necessary) to serve as en route intermediate destinations. Note that such freight with longer service commitments usually cannot be feasibly covered in one day. We recognize that this *a priori* specification is a disadvantage of a one-day planning horizon.

2.3 Problem Definition and Decomposition Approach

Given the inputs described in the previous section, the decision problem is to determine a set of loaded dispatches with dispatch time windows for the 24-hour planning period, such that all newly arriving freight and existing freight in the system is dispatched feasibly with respect to service commitments. Essentially, this is a *timed load planning problem*, similar but more complex than most service network design problems that have been studied in the literature (see Section 2.4. All service network design problems are concerned with determining the best manner to consolidate freight for transportation through a terminal network, given that transportation costs have some fixed component. In the regional LTL operational setting where service times are short, however, it is critical that the timing of such consolidation is explicitly modeled in order to develop realistic results.

In our operational load planning setting, we do not require that each origin-destination pair is served by a unique *path* of terminal to terminal dispatches. Indeed, lower cost solutions may result if such freight is split over multiple paths. Since the number of feasible flow paths between each origin-destination pair may be very large, solving a detailed timed load planning problem directly using an optimization approach is likely to be computationally intractable for realistic problem instances. We therefore adopt a two-phased solution approach. In the first phase, we solve a *freight routing problem* which ignores timing and attempts to identify a small set of candidate terminal to terminal dispatch paths for each origin-destination pair. In the second phase, we solve the more detailed problem of determining a timed set of loaded dispatches for each terminal to terminal leg in the network, where origin-destination flow is allocated to the paths determined in the first phase.

2.4 Related Literature

Previous research on operational load planning problems is limited, although much research has focused on the tactical design of service networks. Crainic [6] provides a good overview of the literature dealing with service network design. Cohn et al. [4] discuss an operational LTL dispatch problem, but they only consider management of equipment and load matching. Powell [13] discusses a heuristic for service network design, where he considers load planning models and the decision of when to use direct loading. Powell and Koskosidis [12] consider tree constraints in freight routing, assuming that the direct loading decisions are already available. Tree constraints assume that paths from all origins to a destination form a tree. They present local improvement heuristics and primal dual algorithms to solve this problem. Farvolden and Powell [8] discuss subgradient methods for the service network design problem, which is similar to the problem considered by Powell [13]. Existing methods have three primary deficiencies with respect to developing daily operational load plans for regional LTL carriers:

1. Most methods do not explicitly model the timing of freight arrivals into an LTL terminal network, and therefore the timing of freight consolidation

2. Most methods do not allow origin-destination freight to routed over multiple paths

3. No methods consider explicit service deadlines for freight

4. No methods provide capability for separately modeling the two primary different handling techniques for LTL freight: trailer unload/sort/reload, and trailer drop-and-hook relaying

5. No methods provide capability for modeling drop-and-hook relaying at non-terminal meet-and-turn locations

Addressing these concerns, we will develop in this chapter new approaches to operational load planning.

2.5 The Freight Routing Problem

The first phase of the two-phase solution approach to the timed load planning problem focuses on finding a small set of dispatch paths for each origin-destination freight flow pair, such that all flow is allocated to some path and total transportation and handling costs are minimized; we denote this the *Freight Routing Problem.* Importantly, this problem ignores all issues of timing, and thus likely overestimates the opportunities for consolidation at each breakbulk. We furthermore assume at this stage that an unlimited number of drivers and trailers are available, and that all terminal capacities are unconstrained.

The freight routing problem flows origin-destination freight through the network, such that total transportation and handling costs are minimized. We assume that the total flow from an origin to a destination can be split up, but we add constraints that bound this splitting. A single trailer type is considered, the pup. Transportation costs are computed as the number of loads (one or two pups) that are dispatched on each network leg, multiplied by the travel cost (driver plus fuel and maintenance costs) on that lane. While this cost therefore ignores empty dispatch costs, it is nonetheless a reasonable objective that will likely lead to low driver operating costs.

The model includes two major classes of freight handling costs. When trailers are unloaded and reloaded at a breakbulk terminal, a cost is incurred to account for these operations in addition to the sorting operation. Trailers may also be relayed through breakbulks, and such trailers incur a lower handling cost that reflects the drop-and-hook cost. Drop-and-hook costs are also incurred at meet-and-turn locations. Since we do not explicitly model drivers during this phase, this model does not look for meet-and-turn opportunities that produce only driver benefits (such as returning drivers to domicile, or maximizing use of allowable drive hours). Instead, meet-and-turns are selected that enable cost reductions due to freight rerouting that outweigh the drop-and-hook costs. The following subsection describes these ideas in more detail.

2.5.1 Relay and meet-and-turn operations

A major advance made in this dissertation in operational load planning is the explicit modeling of trailer relaying and drop-and-hook operations, both at terminals and also at off-terminal meet-and-turn locations. While relaying is very common in all LTL networks, drop-and-hook operations at meet-and-turns represents an interesting opportunity for regional carriers to reduce travel circuity.

In our model, relays and drop-and-hooks at meet-and-turns are modeled very similarly. Every breakbulk terminal is considered also as a *relay* terminal. Full truckloads can pass through breakbulk relays to go on to other terminals. In addition, both inbound and outbound relay trailers can be paired with trailers that are to be or have just been sorted at the breakbulk. For example, on a leg from A to B where B is a breakbulk, consider two trailers. One of these trailers needs to be dispatched through the relay with only drop-and-hook handling, whereas the other trailer needs to unloaded at the breakbulk. We allow this pair to be matched as a single load to be dispatched from A to B. Similar matches are allowed on legs departing breakbulks. Meet-and-turn locations also allow drop-and-hook operations for pup matching, however these operations can only be performed when trailers arrive and depart such that drivers are present continuously while trailers are present. Since the freight routing model ignores timing, this constraint will surface when we describe the timing model in Section 2.6.

Regional carriers do prefer to perform drop-and-hook operations at either breakbulk or satellite terminals, where the former would be used as a relay. Since the carriers have secured trailer yards at these locations, driver timing issues are not relevant. To encourage this, we use a smaller handling cost at relays or satellite meet-and-turns when compared to an off-terminal meet-and-turn.

Due to complexity of modeling drop-and-hook operations, we make some simplifying assumptions in our models. First, we do not allow trailers to move from one meet-and-turn or relay location directly to another meet-and-turn or relay location. While this obviously prohibits multiple overnight relays, such operations are unlikely in practice. Furthermore, at each meet-and-turn or relay, we define an immediate next terminal stop (breakbulk or satellite) for all freight with a common final destination d. All such freight using this meet-and-turn or relay must be routed next to this immediate next terminal. Essentially, this is a constraint required by our modeling approach, since a standard flow conservation constraint will not otherwise prevent disallowed freight sorting at the meet-and-turn or relay. However, this assumption is also not overly limiting given that most freight destined for d will likely follow a single best path.

For each destination d and meet-and-turn or relay m, the immediate next terminal k is selected *a priori* according to the criteria below. See Figure 10 for a graphical depiction of the ideas.

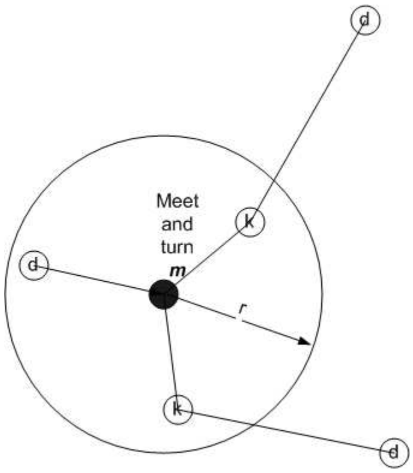

Figure 10: Selection of immediate next terminal for meet-and-turn m and destinations d

Case 1: If the destination terminal d lies within a certain predefined distance radius r of the meet-and-turn m, we select the intermediate next terminal k to be d.

Case 2: If the destination terminal d lies outside r, we select breakbulk terminal k as the immediate next terminal, where k lies within the radius and the distance $c_{mk} + c_{kd}$ is minimized over all breakbulks.

2.5.2 Mathematical model

We are now ready to present a mixed integer programming model for the Freight Routing Problem.

Define the following notation:

$\mathcal{B}$ is the set of breakbulk terminals

$\mathcal{E}$ is the set of satellite terminals

$\mathcal{R}$ is the set of relay locations, each co-located with a breakbulk $b \in \mathcal{B}$

$\mathcal{M}^E$ is the set of all meet-and-turn locations at satellite terminals

$\mathcal{M}^O$ is the set of all off-terminal meet-and-turn locations

$\mathcal{M}$ is the set of all meet-and-turn and relay locations, $\mathcal{M} \equiv \mathcal{R} \cup \mathcal{M}^E \cup \mathcal{M}^O$

$\mathcal{T}$ is the set of all terminals, $\mathcal{T} \equiv \mathcal{B} \cup \mathcal{E}$

$\mathcal{N}$ is the set of all locations, $\mathcal{N} \equiv \mathcal{T} \cup \mathcal{M}$

$k(m, d)$ is the unique next terminal $k \in \mathcal{T}$ to which freight with final destination d is routed after using meet-and-turn or relay $m \in \mathcal{M}$

$\mathcal{K}(m)$ is set of all possible next terminals k for meet-and-turn $m \in \mathcal{M}$: $\mathcal{K}(m) = \{k \in \mathcal{T} \mid k(m, d) = k \text{ for some } d \in \mathcal{T}\}$

q^{od} is the total freight flow, measured in fractional trailers, originating at the beginning of the planning period at terminal o and destined for terminal d, where $o, d \in \mathcal{T}$

c_{ij} is the travel cost on leg (i, j) for a dispatch of a load (one or two trailers), where $i, j \in \mathcal{N}$

H_b is the cost of unloading/sorting/reloading a trailerload of freight at breakbulk $b \in \mathcal{B}$

H_m is the drop-and-hook cost per trailer at $m \in \mathcal{M}$

$T(m)$ is the terminal (satellite or breakbulk) co-located with $m \in \mathcal{R} \cup \mathcal{M}^E$

$M(b)$ is the relay location co-located with breakbulk $b \in \mathcal{B}$, and $M(e)$ is the meet-and-turn co-located with satellite $e \in \mathcal{E}$

x_{ij}^{od} is the decision variable measuring total fractional trailerloads of freight from origin o to destination d dispatched from location i to location j, where $i, j \in \mathcal{N}$, and $o, d \in \mathcal{T}$. This is a non-negative continuous variable.

S_{ij} is the decision variable measuring the number of loads dispatched on leg (i, j), $i, j \in \mathcal{N}$. This is a non-negative integer variable.

n_{ij}^k is the decision variable measuring the number of trailers dispatched on leg (i, j), to be dispatched onward to immediate next terminal k, where $j \in \mathcal{M}$, and $i, k \in \mathcal{T}$. This is a non-negative integer variable.

2.5.2.1 Objective Function

The objective is to minimize the total travel costs and handling costs of all decisions. Travel costs are given by the number of loads dispatched on each leg, multiplied by the travel cost per load for that leg. Handling costs at breakbulks are given by the sum of all fractional trailerloads arriving at but not destined for a breakbulk, multiplied by that terminal's cost per trailerload, and summed over all breakbulks. Handling costs at meet-and-turns and relays are given by the sum of all trailers arriving at the meet-and-turn or relay multiplied by the unit handling cost, and summed over all such locations. Note that we do not consider handling costs at the origin and destination of freight, since such costs are constant regardless of freight routing.

$$min \sum_{i,j \in \mathcal{N}} c_{ij} S_{ij} \quad + \quad \sum_{b \in \mathcal{B}} H_b \left[\sum_{i \in \mathcal{N}} \sum_{o,d \in \mathcal{T}, d \neq b} x_{ib}^{od} \right] \quad + \sum_{m \in \mathcal{M}} H_m \left[\sum_{i \in \mathcal{T}} \sum_{o,d \in \mathcal{T}} x_{im}^{od} \right]$$

2.5.2.2 Constraints

Breakbulk Constraints

- Flow conservation constraints for freight for which the breakbulk is neither origin nor destination:

$$\sum_{i \in \mathcal{N}, i \neq b} x_{ib}^{od} = \sum_{k \in \mathcal{N}, k \neq b} x_{bk}^{od} \qquad \forall b \in \mathcal{B}, \ \ o,d \in \mathcal{T}, \ o,d \neq b \tag{1}$$

- Flow conservation constraints for freight originating at breakbulk b:

$$\sum_{j \in \mathcal{N}, j \neq b} x_{bj}^{bd} = q^{bd} \qquad \forall b \in \mathcal{B}, \ \ d \in \mathcal{T}, \ d \neq b \tag{2}$$

- Flow conservation constraints for freight destined for breakbulk b:

$$\sum_{i \in \mathcal{N}, i \neq b} x_{ib}^{ob} = q^{ob} \qquad \forall b \in \mathcal{B}, \ \ o \in \mathcal{T}, \ o \neq b \tag{3}$$

Satellite Constraints

- Flow conservation constraint for freight originating at satellite e:

$$\sum_{j \in \mathcal{N}, j \neq e} x_{ej}^{ed} = q^{ed} \qquad \forall e \in \mathcal{E}, \ \ d \in \mathcal{T}, \ d \neq e \tag{4}$$

- Flow conservation constraint for freight destined for satellite e:

$$\sum_{i \in \mathcal{N}, i \neq e} x_{ie}^{oe} = q^{oe} \qquad \forall e \in \mathcal{E},\ \ o \in \mathcal{T},\ o \neq e \tag{5}$$

Meet-and-Turn and Relay Constraints

- Flow conservation constraints balancing the arriving flow at meet-and-turn or relay m, with the departing freight to immediate next destination k:

$$\sum_{i \in \mathcal{T}} x_{im}^{od} = x_{m,k(m,d)}^{od} \quad \forall o, d \in \mathcal{T},\ \ m \in \mathcal{M} \tag{6}$$

- Flow using meet-and-turn or relay m destined for final terminal d must be packed into trailers that will be next dispatched to terminal k:

$$\sum_{o,d \in \mathcal{T} \,|\, k(m,d)=k} x_{im}^{od} \leq n_{im}^{k} \quad \forall i \in \mathcal{T},\ \ m \in \mathcal{M},\ \ k \in \mathcal{K}(m) \tag{7}$$

Trailer Flow Constraints at Satellites and Breakbulks

- Number of loaded dispatches on breakbulk-breakbulk leg enough to move all required trailers:

$$\sum_{o,d \in \mathcal{T}} x_{b_1 b_2}^{od} + \sum_{i \in \mathcal{T}} n_{i,M(b_1)}^{b_2} + \sum_{k \in \mathcal{K}(M(b_2))} n_{b_1,M(b_2)}^{k} \leq 2S_{b_1 b_2} \quad \forall b_1, b_2 \in \mathcal{B},\ b_1 \neq b_2 \tag{8}$$

- Number of loaded dispatches on breakbulk-satellite leg enough to move all required trailers:

$$\sum_{o,d \in \mathcal{T}} x_{be}^{od} + \sum_{i \in \mathcal{T}} n_{i,M(b)}^{e} \leq 2S_{be} \quad \forall b \in \mathcal{B},\ \ e \in \mathcal{E} \tag{9}$$

- Number of loaded dispatches on satellite-breakbulk leg enough to move all required trailers:

$$\sum_{o,d \in \mathcal{T}} x_{eb}^{od} + \sum_{k \in \mathcal{K}(M(b))} n_{e,M(b)}^{k} \leq 2S_{eb} \quad \forall b \in \mathcal{B},\ \ e \in \mathcal{E} \tag{10}$$

- Number of loaded dispatches on satellite-satellite leg enough to move all required trailers:

$$\sum_{o,d \in \mathcal{T}} x^{od}_{e_1 e_2} \leq 2S_{e_1 e_2} \quad \forall e_1 \in \mathcal{E}, \;\; e_2 \in \mathcal{E} \tag{11}$$

Trailer Flow Constraints at Meet-and-Turns

- Outbound loads to terminal k must be sufficient to serve all trailers scheduled next to terminal k:

$$\sum_{i \in \mathcal{T}} n^k_{im} \leq 2S_{mk} \quad \forall m \in \mathcal{M}^E \cup \mathcal{M}^O, \;\; k \in \mathcal{K}(m) \tag{12}$$

- Inbound loads to meet-and-turn m must be sufficient to serve all inbound trailers::

$$\sum_{k \in \mathcal{K}(m)} n^k_{im} \leq 2S_{im} \quad \forall i \in \mathcal{T}, \;\; m \in \mathcal{M}^E \cup \mathcal{M}^O \tag{13}$$

2.5.3 Restricting paths from origin to destination

The model described above does not place many restrictions on the allowable paths that an individual origin-destination freight flow may be split over. It reality, however, it may be essential that freight is not split up into very small portions. To prevent this occurrence, each of the x variables could be assigned a minimum value that it may take if it is not zero. In many optimization packages, such conditions are modeled with so-called *semi-continuous variables*: $x^{od}_{ij} \in 0 \cup [a_{od}, \infty)$, $\forall i \in \mathcal{N}, \; j \in \mathcal{N}, \;\; o, d \in \mathcal{T}$.

Semi-continuous variables may lead to excessive branching when the mixed integer program is solved. Therefore, it is best if we do not use more semi-continuous variables than necessary. To reduce their number while retaining the effect of limiting of the number of origin-destination paths, we use the following strategy. First, for origin-destination paths with large freight volumes, we use semi-continuous variables only for flows departing from o (i.e., x^{od}_{oj}) and flows arriving at d (i.e., x^{od}_{id}). Since most paths use fewer than four legs, this strategy is effective at preventing too many paths from origin to destination, since splitting is less likely at intermediate terminals. Second, for origin-destination pairs with low freight volumes, we actually remove the semi-continuous variables and add an $SOS1$ constraint for all flow variables out of the origin, and into the destination. Since $SOS1$ constraints only allow a single element in the set to be nonzero, the effect is that small flow volumes cannot be split out of the origin or into the destination, and the result is usually a single path for this freight.

2.5.4 Implementation considerations

To improve both the realism as well as the solvability of this model, we make several critical implementation decisions for this model, which we now describe. Importantly, these implementation considerations were required when we attempted to solve this model for a fairly large regional LTL carrier that used a large number of historical meet-and-turn locations.

- *Limiting Non-terminal Meet-and-Turn Options*: Most regional carriers that utilize meet-and-turns are likely to have many hundreds of historical locations where meet-and-turns have occurred. Not all are needed in a load planning model, since many are very near each other. We recommend reducing the set of possible non-terminal meet-and-turn locations to a reasonably small set, mainly by eliminating those that lie within a few miles of each other.

- *Limiting Routing Circuity*: To reduce the number of potential freight routing options and improve solution speed, we determine *a priori* a set of breakbulks, relays, and meet-and-turn locations that can be used by freight moving from origin o to destination d. These terminals are selected if they lie within a certain predefined ellipse around the origin-destination pair. The size of this ellipse is larger for selecting breakbulks and smaller for meet-and-turns, since we want to give more freedom to the model to make savings by consolidating whereas

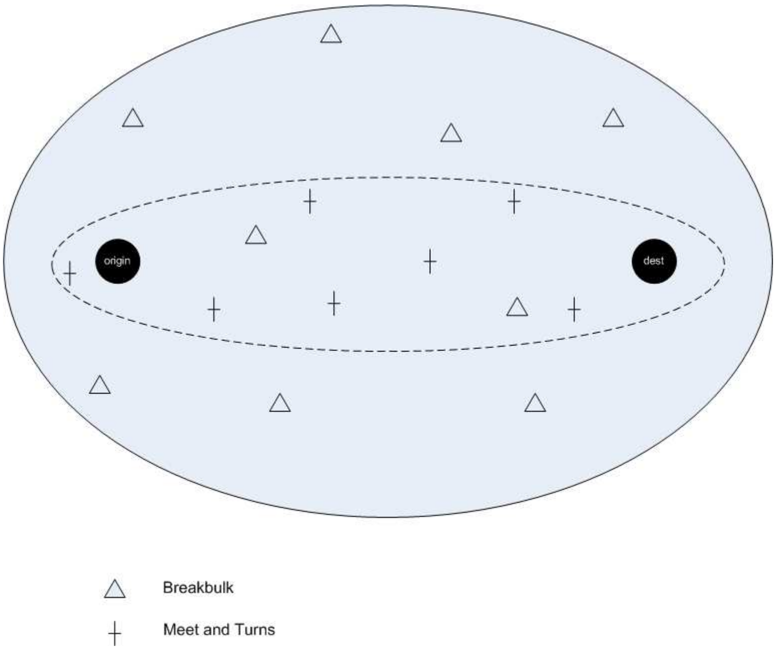

Figure 11: Allowable breakbulk and meet-and-turn locations for freight routed between a specific origin-destination pair

meet-and-turns should be used primarily when they reduce circuity (see Figure 11.

- *Eliminating Infeasible Flow Options*: To both improve computational performance as well as improve model results, we use some time considerations to reduce the number of variables generated. One important such consideration is the fact that the driving time between a terminal and an off-terminal meet-and-turn location must not be more than one half of the allowable driving time per driver shift, since each driver meeting at such a location will need to drive back to his start terminal. A second important consideration is the fact that the ready time and the latest allowable arrival time are both known for each origin-destination freight flow, and that assigning freight to paths which are infeasible with respect to these times is not useful. Therefore, we use the following specific steps for each origin-destination pair:

  - We generate x_{ob}^{od} and x_{bd}^{od} only if $t_{ob} + h_b + t_{bd} < due^{od} - ready^{od}$ where b is an allowable breakbulk for origin o and destination d, h_b is the handling time at b and t_{ib} and t_{bj} are the leg travel times (assumed to be allowable by maximum driving hours). Parameter due^{od} is the time when the freight is due at d, and $ready^{od}$ is the time when it is available for dispatch at o.

- We generate x_{om}^{od} and x_{md}^{od} only if $t_{om} + h_m + t_{md} < due^{od} - ready^{od}$ where m is an allowable meet-and-turn or relay for origin o and destination d, h_m is the handling time at m and t_{om} and t_{md} are the leg travel times, each no greater than one half the maximum driving hours).
- We generate $x_{b_1b_2}^{od}$ only if $t_{ob_1}+h_{b_1}+t_{b_1b_2}+h_{b_2}+t_{b_2d} < due^{od}-ready^{od}$ where b_1 and b_2 are both allowable breakbulks for this pair, h_{b_1} is the handling time at b_1, h_{b_2} is the handling time at b_2 and t_{ob_1}, $t_{b_1b_2}$, and t_{b_2j} are the leg travel times.
- We generate x_{bm}^{od} only if meet-and-turn or relay m is in the allowable set for this origin-destination pair, and furthermore only if $k(m,d) = d$ and $t_{ob} + h_b + t_{bm} + h_m + t_{md} < due^{od} - ready^{od}$, where h_b is the handling time at b, h_m is the handling time at m and t_{ob}, t_{bm}, and t_{md} are the leg travel times.

2.5.5 Solution strategies

To efficiently solve the freight routing mixed integer program, we decided to use a number of techniques designed to improve the quality of the best integer solution that can be found within a limited amount of computation time. Here, we briefly describe those strategies:

- *Branching Order*: We instruct the solver to branch first on the S variables, followed by the n variables, and then the x variables. This strategy reduces the model run-time.

- *Branching Up on Trailer Count Variables*: We instruct the solver to branch up on the n variables. Branching up ensures that a feasible solution is reached more quickly, since the "up" branch for these such variables allows the freight using these trailers to continue to use these trailers. Finding a feasible solution quickly helps prune the branch-and-bound tree more effectively earlier in the solution process.

- *Use of an A Priori Upper Bound*: We calculate an initial upper bound on the optimal solution, and add it to the formulation, again helping to prune the tree more effectively early in the process when no other integer solution has been found. This simple upper bound is calculated by assigning all origin-destination freight to a *primary path* specified by the carrier which specifies a sequence of breakbulks for transfers; no relaying is assumed.

- *Relative Stopping Criteria*: We use a relative stopping criterion that ensures that the branch-and-bound process will prune nodes with lower bounds greater than a value somewhat smaller than the current best integer solution value, rather than only pruning those with bounds no better than the current best integer solution value.

- *Aggressive Cut Generation*: We use aggressive cut generation to focus computational effort on improving lower bounds as quickly as possible.

2.5.6 Assigning O-D freight flow to paths

After the freight routing model is solved, there may exist multiple paths for each origin-destination freight flow. In addition, each such flow may additionally be comprised of both newly arriving freight as well as existing multi-day freight, and that some of that freight may be due at the destination early for distribution while other freight may be due later since the destination only represents an intermediate terminal. Thus, different portions of the flow may have separate ready and due times. We will call these different portions *origin-destination freight portions.* When we obtain multiple paths of flow for a particular O-D freight flow, we should assign the most time-constrained O-D freight portions to the shortest duration paths.

Let $\mathcal{S}(o,d)$ be the set of O-D freight portions for a particular origin o and destination d. Let $ready_s$ and due_s be the ready and due times of the O-D flow portion s, and let f_s be the fractional trailerloads of this portion. Let $\mathcal{P}(o,d)$ be the set of freight paths chosen for that O-D pair. Let w_p be the total fractional trailerloads assigned to path p by the freight routing model. The following algorithm is then used to assign portions to paths:

Algorithm 1 Assigning O-D freight portions to freight paths

```
for All O-D pairs $(o,d)$ do
  while Not all $s \in \mathcal{S}(o,d)$ have been completely assigned to paths do
    Find the most time-constrained unassigned O-D freight portion $s \in \mathcal{S}(o,d)$
    (i.e., that with the smallest difference between $ready_s$ and $due_s$)
    Find the path $p \in P$ with $w_p > 0$ with minimum duration
    if $f_s < w_p$ then
      Assign $s$ to $p$, $w_p = w_p$ - $f_s$
    else
      Assign the partial volume $w_p$ of $s$ to $p$, $f_s = f_s$ - $w_p$
    end if
  end while
end for
```

After this assignment, we might have freight portions that have been split among multiple paths. If a portion of the flow assigned to a particular path is late, then the entire O-D freight portion is considered to be late. To rectify, we now take all of the late O-D freight portions and assign them to the minimum duration path in $\mathcal{P}(o,d)$. Note that this results in a larger flow volume on that path than the value indicated by the solution to the freight routing model.

Once all assignments have been made, each path $p \in \mathcal{P}(o,d)$ will potentially carry some of the flow volume from origin o to destination d, and any positive volume will be comprised of one or more freight portions from $\mathcal{S}(o,d)$. For each such path p with positive volume, we create a *freight volume* v with a size equal to the sum of the assigned fractional trailerloads for this path. Let $\mathcal{V}^{od}$ be the set of all freight volumes for pair (o,d); note that the sum of all the sizes of path freight volumes $v \in \mathcal{V}^{od}$ is q^{od}. Let $\mathcal{V} = \cup_{o,d \in \mathcal{T}} \mathcal{V}^{od}$.

Finally, for each volume $v \in \mathcal{V}^{od}$ we determine a common ready time at o, and a due time at d. The ready time $ready_v$ is the latest ready time $ready_s$ among all freight portions comprising this volume. Similarly, the due time due_v is the earliest due time.

2.6 The Trailer Assignment and Dispatch Timing Problem

An implicit assumption during the freight routing phase is that, on a given day, any freight can be consolidated with any other freight that moves through the same terminal or drop-and-hook location. This assumption is also made by virtually all service network design approaches proposed in the literature to date.

In reality, some freight may arrive at consolidation points earlier than other freight, and may need to depart earlier as well (in order to meet service commitments). Consequently, more trailers and loads may be required than suggested by the solution to the freight routing problem. The primary objective, then, of the *Trailer Assignment and Dispatch Timing Problem* is to determine how to dispatch origin-destination freight through the network to again minimize total transportation and handling costs, while explicitly accounting for the following time constraints:

- Freight ready times and service deadlines
- Driver meet times at off-terminal meet-and-turn locations

After solving the freight routing problem, we have determined a set of paths with assigned volumes to be used for each origin-destination pair. Recall also that each volume $v \in \mathcal{V}^{od}$ may consist of several freight portions with different ready and due times, and that these portions have been split among paths.

In this approach, we will assume that all freight associated with a specific (o, d) volume v will be dispatched simultaneously, respecting the ready time $ready_v$ and due time due_v. Dispatch timing decisions require then that a dispatch time is selected for each volume $v \in \mathcal{V}$ for each *leg* of the path p associated with the volume. Joint dispatch time decisions for all volumes $v \in \mathcal{V}$ imply a number of trailers and loads moved on each network leg.

We consider several important timing considerations when selecting feasible dispatch times:

- the origin ready time $ready_v$ and destination due time due_v for each origin-destination freight volume v defines earliest and latest dispatch times for each leg of the path p;

- the actual dispatch times selected for v on each leg of the path p are linked by *precedence relations* (for example, outbound dispatch from a terminal cannot occur before the arrival of the freight inbound plus processing time); and

- drivers that are meeting for drop-and-hook operations at off-terminal meet-and-turn locations must arrive simultaneously.

Importantly, when timing considerations are explicitly modeled, some path choices made by the freight routing model may become costly; this is especially true in the case of freight routed through meet-and-turns and relays. Therefore, a key feature of our approach will be to model a limited set of *alternate paths* for certain freight volumes, and allow the model to select the best one.

Since dispatches on multiple legs in the network are linked via precedence relations and the requirements of meet-and-turns, a network-wide problem results. We choose a mixed integer programming approach to solve the problem, respecting all constraints generated by precedence relations and allowing some freight to be diverted onto paths different from those selected in freight routing. A difficult-to-solve time-expanded formulation results. We augment the branch-and-bound search with several heuristics to develop good solutions.

2.6.1 Input generation

Given the set $\mathcal{V}$ of freight volumes on paths, we first determine for each $v \in \mathcal{V}$ the earliest ready and latest cut times for dispatches along path legs. Note that this calculation is performed for the primary path, as well as any alternate paths. Let $ready_v^{p,\ell}$ and $cut_v^{p,\ell}$ be the ready and cut times for volume v for dispatch on leg ℓ when path p is followed. The distinction between a cut time and the volume due time is that the cut time is the latest dispatch time on a leg that allows a feasible arrival at the destination d by the due time due_v.

Earliest ready times $ready_v^{p,\ell}$ on origination legs (i.e., those outbound from the origin terminal o) are simply the freight ready times $ready_v$. Ready times on downstream legs in a path are simply imputed by adding the minimum transit (travel plus minimum handling) times to $ready_v$; this can be executed in a single forward pass for each v and p. Latest cut times $cut_v^{p,\ell}$ are similarly computed via a backward pass, starting at the freight destination, where the cut time on the final leg of the path is simply due_v minus the leg travel time.

Second, we note that a subset of the freight volumes $\mathcal{V}$ need not be considered in this model, since they have a trivial timing solution: volumes assigned to direct paths from one satellite terminal to another (with no relays). Assignment and timing is simple for such volumes since by definition they may have only one freight portion (corresponding to arriving freight with a next-day service commitment); the number of trailers required is simply the freight flow volume rounded up, and those trailers may be dispatched any time between the ready and cut time for the single path leg.

2.6.2 Relay and meet-and-turn alternate paths

As mentioned earlier, we introduce for some freight volumes v alternate paths in order to give the timing model some flexibility to potentially reduce costs if freight routing choices prove costly.

For each volume v assigned to a primary freight path p that includes one or more relay or meet-and-turn operations, a single alternate path p' is generated. When selecting alternate paths, our objective is to minimally affect the other portions of the network while offering an option which avoids the meet-and-turn or relay. The following cases describe how we assign the alternates:

Alternate freight paths

- **Case 1**: If path p does not use a meet-and-turn or relay, no alternate path is specified.

- **Case 2**: If path p contains a single meet-and-turn location, then we define the alternate path p' by skipping the meet-and-turn. For example, a path of $A{-}M{-}B{-}C$, where M is the meet-and-turn, will have alternate path $A{-}B{-}C$. Note that since $A - M$ and $M - B$ are both feasible legs, leg $A - B$ is also feasible.

- **Case 3**: If path p uses a single relay, then we define the alternate path p' to replace the relay m with the breakbulk $T(m)$. For example, a path $A - M - C$ where M is a relay yields an alternate path of $A - T(M) - C$. This freight would now be unloaded, sorted, and reloaded with other freight using $T(M)$.

- **Case 4**: If path p uses more than one meet-and-turn or relay, then we define the alternate path p' as the one that applies the Case 2 and Case 3 rules to all such locations. For example, freight path $A - M_1 - C - M_2 - D$ where M_1 is a relay and M_2 is a meet-and-turn is given alternate path $A - T(M_1) - C - D$.

2.6.3 Mathematical model

We use a time-discretized mixed integer programming model for the trailer assignment and dispatch timing problem. The one-day planning horizon is discretized into time buckets of one hour duration. Since we explicitly model trailer drop-and-hook operations at relays and meet-and-turns, we again must account for trailer flows (in addition to load flows) on certain network legs, and much of the model complexity results from this necessity.

Define the following notation, in addition to that of the earlier model:

$\mathcal{U}$ is the set of all time buckets

$\mathcal{V}$ is the set of all origin-destination freight volumes assigned to primary paths

$\mathcal{L}$ is the set of network legs (i, j), where $i, j \in \mathcal{N}$

$\mathcal{P}(v)$ is the set of paths for each $v \in \mathcal{V}$, where $|\mathcal{P}(v)| \leq 2$ since each volume may have at most one alternate path

f_v is the volume in fractional trailerloads for $v \in \mathcal{V}$

τ_ℓ is the total time required for leg $\ell \in \mathcal{L}$, including handling time at the end terminal

c_ℓ is the travel cost on leg $\ell \in \mathcal{L}$

H^p is the total handling cost per trailerload for path p, where the total is the sum of all breakbulk, relay, and meet-and-turn handling costs for the intermediate locations visited by the path

We also use the following additional notation, only for simplicity of presentation:

$\mathcal{P}(v, \ell)$ is the set of paths for each $v \in \mathcal{V}$ that include leg ℓ

$\mathcal{L}(p)$ is the ordered set of network legs in path p, and $\mathcal{L}'(p)$ is the same set excluding the first leg

$\mathcal{L}(X, Y)$ is the set of all network legs connecting any location in set X with any location in set Y

$\mathcal{K}$ is the set of next terminals from all meet-and-turns, $\{\mathcal{K}(m) \mid m \in \mathcal{M}\}$

$\mathcal{V}(m, k)$ is the set of all freight volumes whose primary path includes relay or meet-and-turn m immediately followed by terminal k

$e(t)$ is the actual time corresponding to the start of time bucket t, while $e^{-1}(t)$ is the time bucket within which actual time t is contained

$\mathcal{U}(v, \ell)$ is the set of feasible time buckets for the dispatch of freight volume v on leg ℓ, i.e. any bucket t where $ready_v^{p,\ell} \leq e(t) \leq cut_v^{p,\ell}$ for some path $p \in \mathcal{P}(v)$

$tail(\ell)$ is the tail (from) location for leg ℓ

$head(\ell)$ is the head (to) location for leg ℓ

$prev(\ell, p)$ is the previous leg to ℓ in the path p

Decision Variables

y^v_p is 1 if volume v is dispatched on path $p \in \mathcal{P}(v)$

$y^v_{t,\ell}$ is 1 if volume v is dispatched during time bucket t on leg ℓ

$n^k_{t,\ell}$ is the number of trailers dispatched loaded for destination k on leg ℓ during time bucket t, for legs ending at meet-and-turns and relays ($head(\ell) \in \mathcal{M}$)

$n_{t,\ell}$ is the number of trailers dispatched on leg ℓ during time bucket t, for legs outbound from meet-and-turns and relays ($tail(\ell) \in \mathcal{M}$)

$S_{t,\ell}$ is the number of trailer pairs (loads) dispatched during time bucket t on leg ℓ

2.6.3.1 Objective Function

The objective is to minimize the total travel costs and handling costs of all dispatch decisions; note that if alternate paths were not considered, handling costs could be ignored since they are fixed given paths. The first sum in the expression below represents the total transportation cost, while the second represents the handling costs at breakbulks, meet-and-turns and relays.

$$min \sum_{t \in T} \sum_{\ell \in \mathcal{L}} c_\ell S_{t,\ell} + \sum_{v \in \mathcal{V}} \sum_{p \in \mathcal{P}(v)} H^p f_v y^v_p \tag{14}$$

2.6.3.2 Constraints

Path and Dispatch Constraints

- Each freight volume v is dispatched on a single path:

$$\sum_{p \in \mathcal{P}(v)} y_p^v = 1 \quad \forall v \in \mathcal{V} \tag{15}$$

- Each freight volume v is dispatched on each leg of its path during exactly one time bucket; note that the sum on the right-hand side is necessary since a single leg ℓ may be in more than one path in $\mathcal{P}(v)$:

$$\sum_{t \in \mathcal{U}(v,\ell)} y_{t,\ell}^v = \sum_{p \in \mathcal{P}(v,\ell)} y_p^v \qquad \forall v \in \mathcal{V}, \;\; \ell \in \cup_{p \in \mathcal{P}(v)} \mathcal{L}(p) \tag{16}$$

- Each freight volume may not be dispatched on leg ℓ of selected path p until it has completed leg $prev(\ell, p)$, including necessary handling time between the dispatches:

$$\begin{gathered} \sum_{u \,|\, e(u) \leq e(t) - \tau_{prev(\ell,p)}} y_{u,prev(\ell,p)}^v \geq y_{t,\ell}^v - (1 - y_p^v) \\ \forall \; v \in \mathcal{V}, \;\; p \in \mathcal{P}(v), \;\; \ell \in \mathcal{L}'(p), \;\; t \in \mathcal{U}(v,\ell) \end{gathered} \tag{17}$$

Load Counting Constraints

- For legs connecting satellite terminals, loads are counted simply using assigned dispatch times:

$$\sum_{v \in \mathcal{V}} f_v y_{t,\ell}^v \leq 2S_{t,\ell} \quad \forall \; t \in \mathcal{U}, \;\; \ell \in \mathcal{L}(\mathcal{E}, \mathcal{E}) \tag{18}$$

- For legs ending and starting at meet-and-turn locations, loads are imputed from trailer counts:

$$\sum_{k \in \mathcal{K}(tail(\ell))} n_{t,\ell}^k \leq 2S_{t,\ell} \quad \forall \; t \in \mathcal{U}, \;\; \ell \in \mathcal{L}(\mathcal{T}, \mathcal{M}^E \cup \mathcal{M}^O) \tag{19}$$

$$n_{t,\ell} \leq 2S_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{M}^E \cup \mathcal{M}^O, \mathcal{T}) \tag{20}$$

- For legs connecting breakbulk terminals with satellites, loads are imputed from an aggregation of relay trailers and sorted trailers:

$$\begin{aligned} \sum_{k \in \mathcal{K}(M(head(\ell)))} n^k_{t,\ell'} + \sum_{v \in \mathcal{V}} f_v y^v_{t,\ell} \leq 2S_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{E}, \mathcal{B}),& \\ tail(\ell') = tail(\ell),\; head(\ell') = M(head(\ell))& \end{aligned} \tag{21}$$

$$\begin{aligned} n_{t,\ell'} + \sum_{v \in \mathcal{V}} f_v y^v_{t,\ell} \leq 2S_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{B}, \mathcal{E}),& \\ head(\ell') = head(\ell),\; tail(\ell') = M(tail(\ell))& \end{aligned} \tag{22}$$

- For legs connecting breakbulk terminals, loads must be imputed from freight that is sorted in both locations in addition to freight using one of the relay options:

$$\begin{aligned} n_{t,\ell'} + \sum_{k \in \mathcal{K}(M(head(\ell)))} n^k_{t,\ell''} + \sum_{v \in \mathcal{V}} f_v y^v_{t,\ell} \leq 2S_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{B}, \mathcal{B}),& \\ head(\ell') = head(\ell),\; tail(\ell') = M(tail(\ell))& \\ tail(\ell'') = tail(\ell),\; head(\ell'') = M(head(\ell))& \end{aligned} \tag{23}$$

Trailer Counting Constraints

- Trailers inbound to a meet-and-turn or relay are packed for an immediate next destination k, where they will be unloaded:

$$\sum_{v \in \mathcal{V}(head(\ell),k)} f_v y^v_{t,\ell} \leq n^k_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{T}, \mathcal{M}),\;\; k \in \mathcal{K}(head(\ell)) \tag{24}$$

- Trailers outbound from a meet-and-turn or relay are determined by the dispatch decisions, since waiting is allowed at terminal locations:

$$\sum_{v \in \mathcal{V}} f_v y^v_{t,\ell} \leq n_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{M}, \mathcal{T}) \tag{25}$$

Meet-and-Turn and Relay Constraints

- For each outbound destination from a meet-and-turn or relay, the total number of trailers dispatched over the horizon must be equal to the inbound number of trailers:

$$\sum_{\ell' \,|\, head(\ell') = tail(\ell)} \sum_{t \in \mathcal{U}} n^k_{t,\ell'} = \sum_{t \in \mathcal{U}} n_{t,\ell} \quad \forall\, \ell \in \mathcal{L}(\mathcal{M}, \mathcal{T}) \tag{26}$$

- At off-terminal meet-and-turns, inbound trailers may not wait before outbound dispatch:

$$\sum_{\ell' \,|\, head(\ell') = tail(\ell)} y^v_{e^{-1}(e(t) - \tau_{\ell'}),\ell'} = y^v_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; \ell \in \mathcal{L}(\mathcal{M}^O, \mathcal{T}), \quad v \in \mathcal{V}(tail(\ell), head(\ell)) \tag{27}$$

- At off-terminal meet-and-turns, the number of inbound loads (and hence drivers) must equal the number of outbound loads in any given time period:

$$\sum_{\ell' \,|\, head(\ell') = m} S_{e^{-1}(e(t) - \tau_{\ell'}),\ell} = \sum_{\ell \,|\, tail(\ell) = m} S_{t,\ell} \quad \forall\, t \in \mathcal{U},\;\; m \in \mathcal{M}^O \tag{28}$$

2.6.4 Implementation considerations and solution strategies

As with the freight routing model, we must be careful about the size of actual instances of the trailer assignment and dispatch timing model proposed in the previous section. Clearly, instance size increases with the number of O-D freight volumes, the number of network legs, and the number of meet-and-turn or relay locations. Another consideration is that, for greater accuracy, we would like the time buckets to be of smallest duration possible; instance size grows significantly also with the number of time buckets. In this section, we propose strategies to reduce the size of actual instances so that short time buckets can be used for high model accuracy:

- *Limit Dispatch Time Choice Set*: As proposed above, we use $\mathcal{U}(v, \ell)$ to define the set of feasible time buckets for the dispatch of volume v on load ℓ in any path in $\mathcal{P}(v)$. When generating the model, then, we need only generate variables $y^v_{t,\ell}$ for time periods $t \in \mathcal{U}(v, \ell)$, and eliminates the need for constraints ensuring that ready and cut times are satisfied.

- *Simple Upper Bounds for Count Variables*: We generate simple upper bounds for variables n^k, n, and S by assuming that on each leg ℓ, each flow volume v is served by its own trailers and loads. Given fixed values for the path variables, these bounds are then easily computed.

- *Branching Order*: When branching, we give priority to the load count variables S over the dispatch time selection variables $y^v_{t,\ell}$. Since they are aggregate values, the linear programming relaxations are good estimates of the S values. Branching on these variables first reduces the number of options to be considered downstream in the branch.

2.6.5 Feasible solution heuristics

Given the model and solution strategies outlined above, our experience with practical instances indicates that feasible solutions are not generated quickly by the branch-and-bound process. Thus, we implement heuristic methods that generate good feasible solutions given the linear relaxation solution available at each node in the branch and bound tree. The methodology is now outlined.

We generate the full integer programming model and begin the branch-and-bound solution process. Observe that the $y^v_{t,\ell}$ variables represent dispatch times for volume v, and y^v_p represent path selection. If all such values have integer values at some point, then all decisions regarding trailer and load counts represented by the n and S variables can be computed simply by rounding up. Thus, at each branch-and-bound node we check whether all the y variables are integer and if so generate this full solution by rounding. Note that when applying this strategy at meet-and-turns, it is first necessary to round up the inbound n^k trailer counts, then the outbound n trailer counts to generate first cuts at the S variables. Since load conservation must also be preserved, we may then adjust some S values (upward) to again ensure load conservation at every time period.

This simple method for generating heuristic solutions is implemented at each branch and bound node, and results in many more feasible solutions (and thus upper bounds). Providing such upper bounds makes the pruning of nodes more efficient, and allows the solver to explore the tree more effectively in less time.

2.6.6 Feasible solution improvement heuristic

We have also developed a heuristic to improve feasible solutions constructed by the rounding procedure. Although the proposed heuristic could be executed on every feasible solution produced, in our computational study we use the approach only for the best rounded solution found overall by the branch-and-bound process. Note that since we do not let the branch-and-bound run to completion, the best feasible solution is usually not provably optimal, and thus can be improved.

The improvement heuristic uses the following information, available with any feasible solution generated by rounding:

- $S_{t,\ell}$, the number of loads dispatched at each time period on each leg;
- The contents of these loads, in terms of trailer information in the n variables

and information in the y variables;

- Given the selected dispatch times $y_{t,\ell}^{v}$ for each freight volume v on each leg ℓ, we can define new ready and cut times for each (v, ℓ) such that any joint dispatch choice for all (v, ℓ) in these new windows remains time feasible; and
- Given the new ready and cut time windows for each (v, ℓ), we can impute ready and cut times for trailers, and loads.

The improvement heuristic uses a greedy search to attempt to reduce the number of loads dispatched on each network leg. Note that the heuristic is applied separately for each leg. There are two ways to reduce the number of loads:

1. Remove a dispatch (a load or loads moving at a specific time bucket) altogether. This requires reassigning all affected freight volumes v to other (existing) dispatches on that leg; and
2. Reduce the number of loads in a dispatch by reassigning some of the relevant freight volumes v to other (existing) dispatches on that leg.

After each leg has been processed, we have new exact dispatch times of each O-D flow path on all legs. As earlier, we can again use these to calculate all the ready and cut times. So, whenever we make changes on a leg, since we work within the time bounds, we make sure that none of the cut times are violated. Note that, now we can work with actual times instead of the time buckets which were used in the IP model. On each leg:

- Select the dispatch with the lowest volume.
- Try putting all the O-D freight portions/trailersn in that dispatch into other dispatches on the leg. We also allow the other dispatches on the leg to be moved forward or backward in time to accomodate the O-D freight portions

in the present dispatches. In all movements forward and backward, we ensure that the ready and cut times of all O-D freight portions within the dispatch are valid. While trying to put O-D freight portions into other dispatches, we try to put them in where the number of loads in that dispatch does not increase.

- Repeat for all the loads in the leg in increasing order of volume.

We do this for the satellite-breakbulk legs first followed by the breakbulk-breakbulk legs, again followed by the breakbulk-satellite legs. We do not perform the improvement on the meet-and-turn legs as moving O-D freight portions around might result in the meet-and-turn operation becoming infeasible.

In the second part of the heuristic, instead of moving entire dispatches, we try to move O-D freight portions from dispatches to other dispatches such that the number of loads required for this dispatch decreases. Again, we only move the present O-D freight portion to dispatches where the addition of this O-D freight portion does not require the addition of another load. And again we consider moving the dispatches forward or backward to accomodate the O-D freight portion.

2.7 Statistical comparison with real-life data

2.7.1 Information available from a major regional LTL carrier

We obtained data for a week from a major LTL carrier. This data contained total shipment flow data, the ready times of the shipments and the due times and due dates. We also obtained the network information. Driver information was also available but we will discuss that in greater detail when we discuss Driver Assignment. Finally, the LTL carrier also provided us with information about how they delivered the shipments including the flow paths and the trailers assigned. To process these shipments for use first by the freight routing model, and then by the trailer assignment and dispatch timing model, we have to make a few modifications. We first remove all shipments greater than three days because more than 99% of the shipments are one, two or three day shipments. For converting this to fit the input requirements for our model, we break it up into one-day data using the instance generation method which will be discussed later. We do this by finding the nearest intermediate breakbulk which can be reached in one day. Our breakdown into one-day shipments is sometimes aggressive and may improve service requirements significantly. We then consolidate all shipments from one origin to a destination into an *O-D freight flow.*

We notice that some of the regional LTL carriers have several meet-and-turn locations throughout the network, some of them very near each other. When we replicate the satellite locations also as meet-and-turn locations, the number increases further. To prevent the number from increasing we delete locations which are very near other meet-and-turn locations or relays.

The LTL carrier does not cost and time data for all the legs in their network. Using the shortest path algorithm, we generate the shortest path between all the locations (terminals and meet-and-turns). In the models, we do not have loads traveling on legs whose information was not provided. However, we use this information to create one-day instances.

2.7.2 Creating one-day instances

The data available from the regional carriers is usually not in 1-day format. They have a majority of their shipments flowing on a one-day route but not all. We have to use some strategies to convert all of the shipments into 1-day shipments. The shipment data obtained is primarily comprised of 1,2,3 day shipments. There are very few 4 and 5 day shipments. We consider the routing of the 1-3 day shipments in our model. The 3-day shipments are broken up into 3 parts, each a 1-day shipment. So, basically we have to find 2 intermediate breakbulks where the shipment has to be dispatched to on the first and the second day. The destination for the third day will be the final destination. For a 2-day shipment, we have to find one intermediate location. Before the freight routing, all the O-D volumes generated above are combined to produce one consolidated O-D freight flow. The ready time chosen for this O-D freight flow is the latest ready time and the cut time chosen is the earliest cut time. The ready and cut time information is not available for the intermediate terminals and we have to generate those ourselves. We assume that the shipments need to arrive at the breakbulks by 3 PM so that they are ready to be dispatched after 3 hours of handling time at 6 PM. This is based upon data observed from the LTL carrier who had the earliest ready times of all shipments starting at 6 PM. So the ready time at the intermediate breakbulks is 6 PM and the cut time is 3 PM.

For 2-day freight, the intermediate location chosen is a breakbulk location A which is a breakbulk greater than a specified driving time from the origin (in this case 8 hrs) and within a single day's driving from the origin (11 hrs) and also minimizes the distance $O - A + A - D$ for any given origin destination pair O and D.

For 3-day freight between origin O and destination D, we first find an intermediate location A which is a breakbulk which is greater than a specified driving time from O (8 hrs) and within a single day's driving distance from O and also minimizes the distance $O-A+A-D$ for any given origin destination pair O and D. Then assuming that $A - D$ is a two-day shipment we calculate an intermediate terminal for it using the method described for splitting two-day shipments.

2.7.3 Comparison with the LTL carrier

We make comparisons on various parameters such as most used lanes, load factors and the number of legs used per freight flow path, but the main comparison is based on cost. We consider the transportation and handling costs in this phase. The driver costs will be considered but in the next chapter. The load factors are calculated as

$$\frac{Total\ Weight\ of\ O - Dflowpaths}{Total\ Weight\ available\ because\ of\ the\ loaded\ trailers\ and\ the\ vans}$$

Since we assign intermediate locations to develop a one-day model we have to add the handling costs at the one-day destination for the two-day and the three-day data. We calculate them at the beginning when we break the O-D flow paths into one-day data. We add his number to our net handling costs. The dispatches are available, so the transportation costs are the sums of the mileage for the dispatches times the cost of 1.1$ per mile.

We have shipment data available for one week, ie. all the shipments for Monday through Friday. We will split all the 2-day shipments and 3-day shipments into daily shipments. The number of shipments with delivery deadline more than or equal to 4 days is very small, so we do not consider them in the planning process. We observe that the data for Monday will be split into portions on Monday, Tuesday and Wednesday, data for Tuesday will be split into portions on Tuesday, Wednesday and Thursday and so on. So, the Monday shipment data will only have one-day shipment data for Monday. The Tuesday shipment data will contain 1-day Tuesday shipments and 2-day Monday shipments. The Wednesday shipment data will contain 1-day Wednesday shipment, 2-day Tuesday shipments and 3-day Monday shipments. Similarly for Thursday and Friday shipment data. This implies that the Wednesday, Thursday and Friday shipment data will have 1,2 and 3-day shipments. These will be the most complete instances for comparison purposes. The other instances, ie. Monday, Tuesday (week 1) and Monday, Tuesday (week 2) will not have the complete data and hence will not have the same number of consolidation opportunities as the complete datasets. We discard all the datasets for our analysis and use just the Wednesday, Thursday and Friday datasets.

2.7.4 Computational results

Table 1 summarizes the instance characteristics summed over the three instances solved, ie. Wednesday, Thursday and Friday instances. Note that the number of

O-D pairs, 3356 is much larger than the total volume, 2379.32. This implies that most of the origin destination freight is less than a trailer. This is very interesting to note considering that most national carriers probably have more daily flows between a given O-D pair. There is a fair share of each kind of flow, between breakbulks, between satellites, and between breakbulk and satellites. The satellite terminals are denoted by the letter E in this table as in following tables.

Table 1: Instance data for our models consolidated over 3 days

OD pairs	3356
Total Volume	2379.32
Total BE O-D flows	1534
Total EE O-D flows	830
Total BB O-D flows	992

Table 2 gives a comparison of the performance of the solution techniques developed by us against the actual data observed from the regional LTL carrier. The main comparisons we need to look at it is how the Major LTL carrier column compares with the column on Dispatch Timing. The other comparison we can see in the table is how the solution changes in terms of load factors, costs and other parameters from the Freight Routing problem to the Dispatch iming problem. First, we will compare our solutions against the LTL carrier's implementation for the same instances. We calculate all the parameters based upon the averages for the Wednesday, Thursday and Friday instances.

Observe that the number of paths used by our Dynamic Decision Planning Technology (DDPT) is higher than that used by the LTL carrier. This happens because DDPT allows O-D freight flows to be split more often than most carriers would. We also allow freight flows to split into 3 or 4 portions which hardly happens at LTL carriers. The average number of legs in DDPT is significantly smaller, primarily because DDPT will try to send as much freight direct as possible at the cost of lower load factors on the direct legs, because this avoids handling costs. Note that sending shipments direct will also result in better service levels, which is an added incentive. DDPT sends several Satellite-Satellite O-D flows direct from origin to destination even with low load factors, which would probably never happen at an LTl carrier unless the O-D flow is almost equal to a truckload. This observation is further validated in the Satellite-Satellite Load Factor (EE LF in the table), which is significantly higher for the LTL carrier. We also observe that DDPT tries to send much more freight through relays than the LTL carrier. This is also primarily because sending freight through the Relay results in lower handling costs. So, DDPT takes lower handling costs while letting the load factors dip a bit. This can be validates too by looking at Relay load factors which are lower for DDPT as compared to the carrier. Another validation is looking at the flow through breakbulks in either of the solutions. DDPT decreases the flow through breakbulk and assigns a lot of that to the relays to reduce costs whereas the carrier sends more through the breakbulks and less through the relays. The Breakbulk-Breakbulk Load factor and the Breakbulk-Satellite load factors are higher for DDPT indicating the efficient packing methodology. The total Load factor for DDPT is also higher. The carrier chooses to use several meet-and-turns while DDPT avoids these and again probably routes the freight through the relays. Relays offer more flexibility in terms of timing than meet-and-turn locations. DDPT produces almost 9% savings in cost as compared to the solution implemented by the LTL carrier.

Now, we will compare the solutions produced by Freight Routing problem and the Dispatch Timing problem. Since freight routing is a timing-relaxed version of Dispatch timing, we would expect the total costs, the total trailer pairs (loads) and the transportation and handling costs to go down, which we observe as well. Since Dispatch timing does not allow further splitting of O-D freight portions, the average number of paths can only go down and we observe that it goes down by a bit because

Table 2: Comparison of Freight Routing, Dispatch Timing and LTL carrier solutions

	Major LTL carrier	Freight routing	Dispatch Timing
Avg Legs	2.08	1.78	1.74
Avg Paths	1.17	1.24	1.21
Load Factor (LF)	0.70	0.76	0.75
BB LF	0.84	0.91	0.87
BE LF	0.70	0.72	0.74
EE LF	0.56	0.43	0.44
Relay LF	0.76	0.79	0.72
Total Flow through relays	358.00	540.41	476.80
Total Flow handled at breakbulks	628.88	367.54	431.15
Number of meet-and-turns used	37	9	7
Flow through meet-and-turns	162.11	31.60	23.94
Total Costs	795,569	713,495	725,382
Handling Costs	130,245	117,296	119,467
Transportation Costs	665,324	596199	605,915
Total Trailer pairs used	N/A	2621	2689

some of the alternate paths for O-D freight portions might be the same as the recommended path for another portion for the same O-D flow. This is again reflected in the average number of legs going down in the Dispatch Timing model because alternate paths for O-D freight portions flowing through meet-and-turns will have fewer legs and selecting that would imply a decrease in the average number of legs. Flow through relays and meet-and-turns goes down because freight flowing through these might have been assigned to alternate paths while flow through breakbulks will increase because the same alternate paths for the relays will pass through a breakbulk. The load factors are also affected by the use of alternate paths. Because of relay O-D freight portions being moved into alternate paths through breakbulks, the load factors from Breakbulk-Breakbulk decreases as the extra flow added through the breakbulk may not fit very well. Because of the same reason, the load factors pertaining to flow through relays also goes down.

Run times and Optimality Gaps are shown in Table 3 and 4 respectively. These are again values averaged over the three instances (Wednesday, Thursday and Friday). Since, this is a daily planning model, we do not want the run times to be very high. Something within a duration of about 45 mins will be acceptable. The main contributions to the total run time are from the Model generation and the Integer

Program run-time. We used several techniques (which have been discussed in sections 2.5 and 2.6), to make the model smaller and hence reduce the model generation times. To limit the total Integer Program solution time, after some analysis, we found that a run time of 10 mins for the Freight Routing problem and 15 mins for the Trailer Assignment and Dispatch Timing problem produced reasonably good solutions. Freight routing produces multiple integer solutions in the 10 mins allowed. Dispatch timing produces just one or two integer solutions in the 15 mins allowed. However, the rounding heuristic is able to provide further integer solutions in that time period. Both Freight Routing and Dispatch Timing produce solutions which are in the range of 5% within optimality. The improvement heuristic for the Dispatch Timing problem helps in further reducing the optimality gap. Notice that it matters very little when we stop the Integer Programs. If we look at the first integer solutions as shown in Table 4, these are also quite close to optimality and the final solutions where we stop at are not too much better than the first integer solutions produced.

Table 3: Run Times

	Total Time	IP Time	Heuristic Time
Freight Routing	21 mins	10 mins	N/A
Dispatch Timing	37 mins	15 mins	10 mins

Table 4: Optimality Gaps

	Optimality Gap	Total Integer Solns	Optimality Gap of first Integer Soln
Freight Routing	4.53 %	2.7	6.76 %
Dispatch Timing	5.23 %	1.2 (IP), 26.5(Rounding)	6.53 %
Improvement Heuristic	5.14 %	N/A	N/A

The performance of the improvement heuristic is shown in Table 5. As expected, the heuristic does not affect the handling costs. It only affects the transportation costs which it manages to decrease by about 2 %.

Table 5: Heuristic Solution

Total Costs before Heuristic	Total Costs after Heuristic
740,187	725,382
620,720	605,915
119,467	119,467

2.8 Building Loads

After solving the Freight Routing and Dispatch Timing problems, we have for each origin-destination flow, the paths along which the flow is sent together with the corresponding volumes and we know for each leg how the freight that flows over that leg is packed into dispatches. The final task is to convert this information into actual loads with appropriate dispatch windows, where a load is a set of shipments and the earliest dispatch time of the load is the latest ready time among the shipments in the load and the latest dispatch time of the load is the earliest cut time among the shipments in the load. This involves assigning shipments to flow paths and splitting dispatches on legs into loads.

2.8.1 Assigning shipments to flow paths

For each origin-destination flow, we have the paths along which the flow is sent together with the corresponding volumes, which we have called O-D freight portions. The origin-destination flow represents a number of shipments with their volumes. The two main parameters for packing shipment into trailers is volume and weight. We will use the term *volume* as a proxy for the size of the shipment. In actual implementation, we can either use weight or volume for our planning models but not both. As a first step into building actual loads, we assign each shipment to an O-D freight portion. For simplicity, we have assumed in the freight routing model that shipments can be split. In reality this is not allowed, therefore we assign shipments to a unique O-D freight portion. There is a possibility that this increases the number of dispatches on certain legs.

For each origin-destination flow, we have the set of O-D freight portions P, the flow f_p on path for the O-D freight portion p ($p \in P$), the set of shipments S, and the volume f_s of shipment s ($s \in S$). As a flow path of an O-D freight portion typically consists of multiple legs, O-D portion p is part of a number of dispatches. Let r_p be the minimum extra space available among all the dispatches that contain O-D freight portion p. Therefore, the volume of O-D freight portion p can be increased by r_p without increasing the number of dispatches. Note that r_p may change as a result of shipments being assigned to O-D freight portions involving common dispatches.

We assign shipments to O-D freight portions using a 2-phase approach. Each phase involves the solution of a bin packing problem, where the O-D freight portions form the bins. In Phase 1, the capacities of the bins are set to f_p, the volume of the O-D freight portions. The objective function is to pack as many shipments in the bins as possible (i.e., pack as much volume in the bins as possible). In Phase 2, the capacities of the bins are set to r_p plus what ever capacity remained at the end of Phase 1. The objective to pack the remaining shipments in the bins while minimizing the maximum capacity violation.

2.8.2 Splitting dispatches into loads

After assigning shipments to O-D portions, we know for each dispatch which shipments are involved. Next, we need to decide how to partition the set of shipments in a dispatch into loads and to compute for each of the loads the earliest and latest possible dispatch times. We assume that any set of shipments with a combined volume that is less than or equal to the available capacity fits. In practice, this may be too optimistic as it is difficult, due to stacking and packing issues, to completely fill up trailers. For dispatches on legs involving a relay or a meet-and-turn location, i.e., for dispatches with trailer level information, we partition shipments of a dispatch into trailers; for all other legs we partition shipments of a dispatch into trailer pairs.

For each dispatch, we know the set of shipments S, the volume f_s of shipment s ($s \in S$), the ready time r_s of shipment s ($s \in S$), and the cut time c_s of shipment s ($s \in S$). Since all shipments can be feasibly dispatched at the dispatch time, regardless of how we partition the shipments into loads, each load will have at least one feasible dispatch time. We have two objectives when assigning shipments to loads:

- minimizing the number of loads, and
- maximizing the minimum flexibility (or maximizing the sum of the flexibilities) of the loads, where the flexibility of a load is the difference between the latest and earliest dispatch time.

We handle the first objective by enumerating over the number of loads, starting with the smallest possible number of loads, and increasing that number until we reach feasibility. In each iteration we solve an optimization problem that focuses on the second objective.

Let K denote the set of loads we are trying to create. Define the following variables: r_l, the earliest possible dispatch time of load l, c_l the latest possible dispatch time of load l, and x_l^s indicating whether or not shipment s is assigned to load l. The optimization problem can now be formulated as follows:

$$\max C \text{ or } \max \sum_{l=1}^{K} (c_l - r_l)$$

$$\sum_{l=1}^{K} x_l^s = 1 \qquad \forall s \in S$$

$$r_l \geq x_l^s r_s \qquad \forall l \in 1 \ldots K, \forall s \in S$$

$$c_l \leq x_l^s c_s \qquad \forall l \in 1 \ldots K, \forall s \in S$$

$$c_l - r_l \geq C \qquad \forall l \in 1 \ldots K$$

Trailers going to a meet-and-turn are matched according to their destination. Shipments going to the same destination have to be packed together and shipments going to different destinations have to be packed into different trailers. Therefore, we have to set up an optimization problem for each destination. Once the solution to these optimization problems has been obtained, we have to pair the trailers into loads. In doing so, we again try to maximize flexibility.

2.8.3 Determining earliest and latest possible dispatch times

The earliest and latest possible dispatch times for a load when derived simply from the ready and cut times of the shipments in the load may not be realizable (and thus too optimistic) due to interactions with other loads as a result of precedence relations between dispatches. Algorithm 2 will update and correct the earliest and latest possible dispatch times for the loads and will properly account for any precedence relations. Algorithm 2 will call the recursive functions 3 and 4. The algorithms basically go through the set of loads updating their ready and cut times. However, each update of the ready time for a load involves updating the ready times for all the loads succeeding that load. This is where the recursive call to function 3 comes. Function 3 will recursively update ready times till all succeeding loads have been updated. Similarly, when cut time of a load is changed, the cut times for all the preceding loads have to be changed which is done by function 4.

At the end of this load generation process, we have obtained a set of loads with ready and cut times and a list of shipment contents. At this stage, we are ready to assign drivers to the set of loads. The driver assignment process will be discussed in Chapter 3.

Algorithm 2 Generating Implied Ready and Cut Times

```
Ready Time of all dispatches = 100000
Cut Time of all dispatches = -1
for Shipment $s \in S$ do
  for Dispatch/Load $d$ which is part of the path of shipment $s$ do
    Calculate the ready time implied on that leg by $s$ which is the ready time at origin + handling and transportation times
    if Ready Time implied by $s >$ ready time of dispatch/load which $s$ is part of then
      Update the ready time of this dispatch/load. Let $r$ be the ready time
      for All other shipments $s_2$ which are part of this dispatch/load do
        Find the dispatch of which $s_2$ is part of next. Call it $d_2$
        if $d_2$ exists then
          Calculate the ready time at $d_2$ which is $r$ + transportation time on $d$ + handling time on $d_2$. Call this $r_2$
          Call function CalculateReady($d_2$, $r_2$) given in Algorithm 3
        end if
      end for
    end if
    Calculate the cut time implied on that leg by $s$ which is the cut time at destination - handling and transportation times
    if Cut Time implied by $s <$ cut time of dispatch/load which $s$ is part of then
      Update the cut time of this dispatch/load. Let $c$ be the cut time
      for All other shipments $s_2$ which are part of this dispatch/load do
        Find the dispatch of which $s_2$ is part of before this. Call it $d_2$
        if $d_2$ exists then
          Calculate the cut time at $d_2$ which is $c$ - transportation time on $d_2$ - handling time on $d$. Call this $c_2$
          Call function CalculateCut($d_2$, $c_2$) given in Algorithm 4
        end if
      end for
    end if
  end for
end for
```

Algorithm 3 Recursive Function CalculateReady

```
Function CalculateReady(d,r)
if $r >$ ready time of $d$ then
    Update ready time of $d$
    for All shipment $s$ part of $d$ do
        Find the next dispatch which $s$ is part of. Call it $d_2$
        if $d_2$ exists then
            Calculate the ready time at $d_2$ which is $r$ + transportation time on $d$ + handling time on $d_2$. Call this$r_2$
            Call function CalculateReady($d_2$, $r_2$)
        end if
    end for
end if
```

Algorithm 4 Recursive Function CalculateCut

```
Function CalculateCut(d,c)
if $c <$ cut time of $d$ then
    Update cut time of $d$
    for All shipment $s$ part of $d$ do
        Find the previous dispatch which $s$ is part of. Call it $d_2$
        if $d_2$ exists then
            Calculate the cut time at $d_2$ which is $c$ - transportation time on $d_2$ - handling time on $d$. Call this$c_2$
            Call function CalculateReady($d_2$, $c_2$)
        end if
    end for
end if
```

CHAPTER III

DRIVER ASSIGNMENT

3.1 Introduction

After the Freight Routing and Dispatch Timing models have been solved, the loads for the upcoming period have been built. That is, all shipments have been assigned to loads along a path from their origin to their destination and the dispatch windows of these loads ensure that all shipments reach their destination on time. To complete the schedule for the upcoming period, driver duties need to be constructed such that all loads are moved and such that hours of service regulations and company rules and policies are respected.

Scheduling drivers is challenging because of the restrictions imposed by the Department of Transportation, i.e., the hours of service regulations, as well as those imposed by the operating policies of the carrier. A driver is allowed to drive for up to 11 hours and work for up to 14 hours in a duty, where work includes short rest time and time spent waiting. We denote the driving time limit by t_{drive} and the duty time limit by t_{duty}. Different from national, unionized LTL carriers, which must manage driver bids which further restrict feasible driver duties, most regional LTL carriers have substantial flexibility when building driver duties. The regional carrier that motivated our research and that provided historical driver data employed two types of drivers. The first type, whom we will refer to as a *domicile driver*, needs to return to his domicile or home location every night. The second type, whom we will refer to as a *layover driver* can spend one night away from his domicile, but needs to return to his domicile every other day. Typically, there are no restrictions on where a lay-over driver spends his night away from the domicile.

3.2 Related Literature

There are several papers in the operations research literature which focus on resource allocation in the LTL trucking industry, but most of them focus on the management of tractors and trailers. Crainic and Roy ([5]), Powell ([11]), and Caliskan and Hall ([3]) do specifically consider drivers and focus on returning drivers to their domicile. However, these articles ignore hours of service regulations and other rules that may restrict driver duties.

There is a significant body of research literature focused on crew scheduling and rostering problems for transportation systems operating fixed schedules, such as passenger airlines, transit systems, and passenger rail services. Again, most research addresses tactical planning problems and uses a set covering or set partitioning model to choose a subset of partial schedules. The models are solved exactly or heuristically by enumeration or column generation. Barnhart et al. ([2]) provides a thorough overview of airline crew scheduling. Barnhart et al. ([1]) further discuss airline crew scheduling techniques. Recent advances in this field focus on tactical crew planning under uncertainty (see e.g., Schaefer et al. ([14]). Importantly, the solution times required by these approaches make it difficult to apply them in an operational setting with dynamically changing data. Advances in solution speed using specialized solution techniques still lead to long computation times. Elhallaoui et al. ([7]) reports computation times greater than an hour for scheduling problems with more than 1,500 tasks.

There is lot of research focusing on dynamic resource/asset allocation problems, typically in the context of managing drivers for truckload transportation firms. The body of research by Powell and co-authors focuses primarily on methods for handling data uncertainty. The adaptive dynamic programming approach applied to problems with a discretized time dimension outlined in the paper by Godfrey and Powell ([9]) appears to be a promising approach for solving this type of problem. Yang et al. ([15])

consider a dynamic truckload assignment problem and show, using a simulation, that myopic rolling horizon reoptimization policies can perform quite well when compared to an a posteriori optimization. Their work, however, ignores the resource time constraints crucial for driver scheduling decision-making. It should also be noted that truckload driver management problems are quite different from the LTL driver management problems, since load requests do not require movement between distinct terminals and tend to arrive with less predictability than LTL loads.

Powell et al. ([10]) consider a deterministic LTL driver scheduling problem quite similar to the operational problem that we study. After concluding that integer programming techniques are computationally prohibitive due to instance sizes, the authors instead apply an approximate dynamic programming methodology similar to those developed for stochastic problems. This approach yields promising results with computation times in the range of an hour when all constraints are included. It relies on a discrete representation of time (with computation times that depend on the fineness of the discretization) and a discrete representation of the state of a driver, including the remaining drive hours.

3.3 The Driver Assignment Problem

As mentioned above, after the Freight Routing and Dispatch Timing models have been solved, all shipments have been assigned to loads along a path from their origin to their destination and the dispatch windows of the loads ensure that all shipments reach their destination on time. More specifically, for each load l, we have a leg leg_l, a ready time at the origin r_l, a cut time at the destination c_l, and a set of preceding loads PL_l.

In addition to the load information, we also have information on the drivers available to perform loads. Three types of drivers are available: drivers who start from and return to their domicile, drivers who start away from their domicile, but have to return to their domicile, and drivers who start at their domicile and do not have to return to their domicile. Therefore, at the start of the duty generation process we have the following information for each driver d: domicile dom_d, starting location $init_d$, ready time $tstart_d$, compulsory end time $tend_d$, and ending location $final_d$; the ending location need not be specified for all drivers.

Finally, there is the network itself. We have a set of legs with for each leg le an origin $orig_{le}$, a destination $dest_{le}$, a travel time tt_{le}, and a set of possible meet and turn locations M_{le}; the set of possible meet-and-turn locations is empty on legs which do not involve meet-and-turns.

3.4 Solution Methodology

The basic idea of our solution approach is to create feasible driver duties and match these up with drivers using a set partitioning model. Unfortunately, the number of feasible driver duties or driver tours is huge and solving a set partition problem with so many tours is (too) time consuming. In Table 6, we show how the number of tours is affected by the number of loads. The loads selected for this study were chosen randomly from the available loads. The number of tours increases exponentially with the number of loads.

Table 6: Number of tours

Loads	Tours
200	23,434
400	46,534
600	75,323
800	153,992
1000	288,787
1200	472,122
1400	1,123,215
1600	3,624,761

The number of drivers for the regional LTL carrier that motivated our research is around 1000 and the number of loads that needs to be moved daily is around 1500.

Because of the huge number of feasible driver tours, we need to carefully control their generation. We have chosen to solve the driver duty generation problem in phases. By doing so, we give up some in overall quality, but we gain enormously in computational efficiency. The division in phases is guided by tour types. In each phase, we solve a set packing problem.

3.4.1 Algorithm

As mentioned above, we generate driver duties in phases; five to be precise. In each phase, we find drivers for certain types of tours. In all phases, we use the ratio of the driving time in a tour and the driving time available for a driver as the quality of a tour/driver combination. That is, we consider a tour/driver combination to be of high quality if most of the available driver hours are used up when performing the tour.

3.4.1.1 Phase 0: Perfect meet-and-turn Tours

In Phase 0, we assign drivers to "perfect meet-and-turn tours." A perfect meet-and-turn tour occurs on a very long leg (a leg with a travel time of more than 10 hours) when there are loads in both directions and when the dispatch windows of the loads allow drivers from both endpoints to meet at the midpoint, exchange trucks, and return to their initial location. We look for such opportunities and assign drivers. No optimization problem is solved in this case.

3.4.1.2 Phase 1: Perfect Domicile Tours

In Phase 1, we assign drivers to "perfect domicile tours." A perfect domicile tour starts and ends at the same terminal, does not involve any empty movements, and has high quality, i.e., quality greater than or equal to 0.85. The tours may contain any number of loaded moves as long as they satisfy the limit on driving hours. The tours are perfect in the sense that they do not involve any empty travel and start and end at the same location. (Note that loads with a travel time greater than $t_{drive}/2$ hours can never appear in a perfect domicile tour.) We consider both domicile drivers and layover drivers at their domicile for these tours.

Since drivers assigned to perfect domicile tours return to their domiciles, we prefer to use drivers who have to return to their domicile in this phase. Therefore, in the optimization, if two drivers can perform a particular tour, we include an incentive for using a driver who has to return to his domicile. A perfect domicile tour can be viewed as a sequence of loads that can be performed in order without violating any of the load dispatch windows.

We generate perfect domicile tours using recursive Algorithm 5. For each tour, we save its start time, its latest feasible start time, its loads and the total wait time required.

The latest feasible start time is updated each time we add a load to the tour.

Algorithm 5 Generating Tours

```
for all $l$ in loads do
    CreateTours($l, 0, 10000, 0, 0, [\,], orig_l$)
end for
```

For convenience, we refer to the origin of a load (*orig*), the destination of a load (*dest*), and the travel time of a load (*tt*) although these terms were defined for a leg and a load only specifies an associated leg.

The parameters of the function **CreateTours**, shown in Algorithm 6, are: (1) a load (l) to be appended, if possible, to the partial tour, (2) the start and end time of the dispatch window for the tour (t_{start} and t_{end}), which will change as loads are appended to the tour, (3) the total drive time and the total wait time of the current partial tour ($drive_t$ and $wait_t$), (4) the set of loads already in the partial tour (A), and (5) the beginning and ending location of the partial tour (*domicile*). The total

Algorithm 6 Function CreateTours($l, t_{start}, t_{end}, drive_t, wait_t, A, domicile$)

```
for all $l_2$ in loads such that $l_2 \neq l$ and $orig_{l_2} = dest_l$ and $drive_t + tt_{l_2} < t_{drive}$ and $t_{start} + drive_t + wait_t + tt_{l_2} \leq c_{l_2}$ do
  if $t_{end} + drive_t + wait_t \leq r_{l_2}$ then
    $wait = r_{l_2} - (t_{end} + drive_t + wait_t)$
    $start = r_{l_2}$
    $t_{start} = t_{end}$
    $t_{end} = t_{end}$
  else if $t_{start} + drive_t + wait_t \leq r_{l_2} \leq t_{end} + drive_t + wait_t$ then
    $wait = 0$
    $start = r_{l_2}$
    $t_{start} = r_{l_2} - drive_t - wait_t$
    if $c_{l_2} \geq t_{end} + drive_t + wait_t + tt_{l_2}$ then
      $t_{end} = t_{end}$
    else
      $t_{end} = c_{l_2} - tt_{l_2} - drive_t - wait_t$
    end if
  else
    $wait = 0$
    $start = t_{start} + drive_t + wait_t$
    $t_{start} = t_{start}$
    if $c_{l_2} \geq t_{end} + drive_t + wait_t + tt_{l_2}$ then
      $t_{end} = t_{end}$
    else
      $t_{end} = c_{l_2} - tt_{l_2} - drive_t - wait_t$
    end if
  end if
  if $drive_t + wait_t + tt_{l_2} + wait \leq T_{duty}$ then
    if $dest_{l_2} = domicile$ then
      Save tour
    else
      Add the load $l_2$ to A
      CreateTours($l_2, t_{start}, t_{end}, drive_t + tt_{l_2}, wait_t + wait, A, domicile$)
    end if
  end if
end for
```

driving time and total waiting time of the current partial tour can be calculated from the information in A, but are given here to make the pseudo code more explicit and readable.

At the end of the tour generation process we have a set of driver tours T. For each tour $t \in T$, we check if it satisfies the hours of service limits and, if so, we determine the earliest dispatch time rt_t and the latest possible dispatch time ct_t. Finally, we introduce indicator values $lt_{l,t}$ with value 1 if load l is part of tour t and 0 otherwise.

A complication that may occur when two perfect domicile tours are selected by the optimizer is that precedence relations between loads cause infeasibility, i.e., the tours cannot feasibly be executed at the same time. Therefore, we analyze each pair of selected tours and check if they give rise to any violated precedence constraints. For a given tour t, we can use rt_t and ct_t to determine a dispatch window for each of the loads. Once dispatch windows of the loads in a tour are computed, it becomes easy to check whether there exist feasible start times for a pair of tours that will not lead to any precedence constraint violations. Let qt_{t_1,t_2} be 1 if tour t_1 and tour t_2 do not result in a precedence constraint violation and 0 otherwise.

To complete the information used by the optimization model, let D denote the set of drivers and let $tstart_d$ and $tend_d$ define the start and end time of the dispatch window for the driver. Let $D_A \subset D$ be the set of drivers that have to return to their domicile.

Note that based on the dispatch window of driver d and the dispatch window of tour t it is trivial to determine if driver d can be assigned to tour t. (Of course the domicile of the driver also has to be the same as the start and ending location of the tour.) Let $dt_{d,t}$ be 1 if driver d can be assigned to tour t and 0 otherwise.

We are now ready to formally define the optimization problem. Decision variable $x_{d,t}$ will be 1 if driver d is assigned to tour t and 0 otherwise. (Variable $x_{d,t}$ will be set to zero whenever $dt_{d,t}$ is zero.)

Objective function

$$\max \sum_{d \in D} \sum_{t \in T} \sum_{l \in L} x_{d,t} lt_{l,t} tt_l + 0.1 \sum_{d \in D_A} \sum_{t \in T} x_{d,t}$$

The first part of the objective function represents the total load miles whereas the second part of the objective function represents an incentive for using drivers who need to return to their domicile.

Constraints

- Each driver cannot be assigned to more than one tour.

$$\sum_{t \in T} x_{d,t} \leq 1, \qquad \forall d \in D \tag{29}$$

- Each load can be assigned at most once.

$$\sum_{t \in T} \sum_{d \in D} x_{d,t} lt_{l,t} \leq 1 \qquad \forall l \in L \tag{30}$$

- Certain tours cannot be selected together.

$$\sum_{d \in D} x_{d,t_1} + x_{d,t_2} \leq 1 + qt_{t_1,t_2} \qquad \forall t_1, t_2 \in T \tag{31}$$

The set of constraints (31) may be huge and may make solving the optimization problem computationally prohibitive. Therefore, we solve the problem without constraints (31) and then, if needed, we remove some tours from the solution to satisfy constraints (31). Even after removing some of the tours to satisfy constraints (31), there is no guarantee that all precedence constraints are satisfied. We have only verified pairs of tours to see if precedence relations would result in infeasibility. This of course does not guarantee that a set of three selected tours is always feasible.

The complication pointed out above is a result of the fact that the tours have dispatch windows for each of the loads in the tour as opposed to fixed dispatch times. The advantage of this approach is that it keeps the number of tours reasonable; the disadvantage is that accounting for precedence relations exactly is difficult.

One way around this is to work with timed copies of a tour, i.e., for each tour construct a number of copies, each with specific feasible dispatch times for the loads in the tour. Once we have tours with specific dispatch times for the loads, precedence violations can be detected (or prevented) by pairwise comparisons of tours. Of course there is a heavy price to pay: many timed copies may have to be generated for a single tour.

Therefore, we decided to simply solve the integer program and then to remove tours to restore feasibility, if necessary. This approach was effective as only few tours had to be removed (around 5 %). Algorithm 7 describes the procedure in more detail. We assign dispatch times to the loads in order of their ready times, always dispatching them as early as possible (i.e., when the load is ready, when the preceding loads have been completed, and when the driver is available). Because the precedence graph is acyclic, assigning dispatch times in this order results in a feasible schedule if one exists. If we encounter a load where the dispatch time violate the cut time, the tour covering this load is removed (and therefore also all the loads covered by that tour).

Algorithm 7 Evaluating Tour Feasibility

```
for all $t$ in tours do
  for all $l$ in $t$ do
    Dispatch $l$ as early as possible (i.e., when the load is ready, when the preceding
    loads have been completed, and when the driver is available).
    if Dispatch time for $l$ is greater than cut time for $l$ then
      Remove $t$ from the solution (and all loads in $t$)
    else
      Fix the present dispatch time for $l$ and continue
    end if
  end for
end for
```

3.4.1.3 Phase 2: Good Layover Tours

In Phase 2, we assign drivers to tours which include long loads, i.e., loads with $tt_l \geq \frac{t_{driver}}{2}$. These tours cannot be performed by domicile drivers and have to be assigned to layover drivers (either starting from their domicile or on their way back to their domicile). To increase tour quality we try to add (short) loads to the tour. Different from Phase 1, we allow empty moves as part of the tour, i.e., we allow moving empty trailers from the destination of one load to the origin of a subsequent load. Tours are created according to *templates.* A template is another mechanism to control the size of the optimization problem that needs to be solved. Because we allow empties, the number of feasible tours may be huge. However, many of them are not likely to have high quality. Therefore, instead of generating a huge number of tours and then finding that most of them have low quality and can be discarded, we restrict the generation of tours to tours with "structures" that are likely to result in high quality.

For notational convenience, we always allow drivers to drive empty from their present location to the origin of the first load and also to drive empty from the destination of the last load to the domicile (in the case the driver needs to return to his domicile). To be considered, a tour also has to satisfy driving and duty time limits and has to have a minimum quality. Let L denote a long load, let S denote a short load, and let E denote an empty move. The following tour templates are used in this phase:

- L
- L-S-S
- S-L-S
- S-S-L
- S-E-L
- L-E-S

Since, we need to get layover drivers that slept away from their domicile back to their domicile, we try to assign as many layover drivers away from their domicile as possible in this phase. (This way they will be moving at least one long load.) Therefore, we set the minimum required tour quality for layover drivers away from their domicile lower than the minimum required tour quality of layover drivers at their domicile. Furthermore, we provide an incentive in the objective function for utilizing drivers who are away from their domicile.

The preprocessing and postprocessing of ready and cut times and checking of precedence works the same as in Phase 1. Note that after Phase 1, dispatch times have been assigned to several loads, so we have to keep these in mind when calculating the ready and cut times for loads in Phase 2.

3.4.1.4 Phase 3: Good meet-and-turn Tours

In Phase 3, we look at legs which are more than 40 % of t_{drive} and have loads in both directions which can be paired to use a meet-and-turn.

These are combined with other loads so that the resulting tour/driver combinations will have high quality. If $L1$ and $L2$ are the loads on the meet-and-turn leg, then the following templates involving $L1$ are used:

- L1-S-E
- L1-E-S
- S-L1-E
- S-E-L1
- E-S-L1
- E-L1-S
- L1

Similar templates involving $L2$ are used. After generating tours involving $L1$ and generating tours involving $L2$, we check which tours for $L1$ and $L2$ can be paired together feasibly, i.e., satisfying driver availability and load ready and cut times. Two drivers have to be assigned to each pair. A set partitioning formulation is solved to assign drivers to tours. Only domicile drivers are allowed in this phase. The combined driving time of the tours has to satisfy a quality constraint in this phase. The quality constraint used in this phase is 0.65.

To create dispatch times, we first make a decision as to the exact time when the meet-and-turn operation occurs as that can be readily computed. From there we work backwards and forwards to calculate actual dispatch times of all the loads.

3.4.1.5 Phase 4: Any Tours

In Phase 4, we generate tours with the following templates. Here X is any load and E denotes an empty.

- X
- X-X
- X-X-X
- X-X-X-X
- X-E-X
- X-X-E-X
- X-E-X-X

The goal in this phase to assign drivers in such a way that *all* loads are moved.

3.5 Computational Study

We have load and driver data available for one day. We use drivers who have to return to domicile daily and drivers who return every alternate day. We solve the problem instance using the methodology proposed above. Table 7 summarizes the results obtained when solving a one-day instance using the above methodology.

Table 7: Overall driver assignment statistics

Total Loads	1161
Loads covered	1156
Total drivers	715
Drivers used	622
Average tour quality	0.81
Empty travel percentage	12

The instance has 1161 loads and 715 drivers. The loads were generated by the Freight Routing and Dispatch Timing models based the data provided by the carrier. The drivers were copied from a master data file used by the carrier. The carrier also employs weekly drivers and sleeper team drivers, which were not considered in our instance. The initial locations for layover drivers were generated randomly, with the domicile have probability 0.5 of being selected, and the other terminals having probability $\frac{0.5}{n-1}$ of being selected (where n is the total number of terminals).

Of the 1161 loads, 1156 are dispatched feasibly by the algorithm. For the remaining 5 loads no drivers could be found (which may of course be a result of the random placement of layover drivers). The algorithm utilizes 622 of the 715 drivers. It is important to note that there are many drivers among the 715 who are away from domicile and have to return to their domicile during the current planning period. Some of these drivers return to their domicile empty. It is also worth observing that the number of drivers considered by our models is far fewer than the total driver pool, which has more than 900 drivers. Still, the algorithm uses even fewer drivers; almost

15 % fewer.

Table 8 presents more detailed statistics of the driver assignment methodology. In Phase 0, we search for perfect meet and turn opportunities and impose a high tour quality (0.87). The average tour quality obtained is 0.92. In Phase 1, we search for tours starting and ending at the same location and impose a high tour quality (0.85). The average tour quality obtained is 0.94. Phase 1 is one of the most productive phases in terms of the number of loads covered as more than 40 % of the loads are covered in this phase. We prefer to use domicile drivers in this phase and this is reflected in the results as 152 of the 154 drivers used are domicile drivers. In Phase 1, only short loads are covered. In Phase 2, we aim to cover many long loads. We are successful as the majority of long loads is indeed covered. Since the loads are long and we are not using meet-and-turns, getting drivers back to their domicile not possible; only layover drivers can be used. We impose two tour quality limits in this phase: one for drivers at their domiciles (0.75) and one for drivers away from their domiciles (0.60). Drivers away from their domiciles must get back to their domiciles, loaded or empty, so for them we allow lower quality tours. However, Phase 2 still produces high-quality tours as the average tour quality is 0.84. This is the first phase in which we allow empty travel and it represents 13 % of the total miles traveled. In Phase 3, we search for meet-and-turn opportunities. We want to utilize domicile drivers in this phase because meet and turns will help them return to domicile. As we are nearing the end of the driver assignment process, a lower tour quality limit is imposed (0.65). However, the average tour quality obtained is still good at 0.76. Empty travel is even slightly lower than in Phase 2 at 12 %. Finally, in Phase 4, we try to cover all remaining loads and do not impose any tour-quality limit. It is interesting to observe that even without imposing any restrictions we obtain an average tour quality of 0.64 and an empty travel percentage of 20 %. This indicates that the objective function of maximizing loads covered does a good job of tour selection. In Phase 4, we cover

237 loads while utilizing mostly domicile drivers and some layover drivers. Phase 4 also ensures that all drivers who must get back to their domicile do so.

Overall, we see that as the tour quality imposed decreases, the observed tour quality also decreases. The empty mileage percentage is 11%, which is small for regional LTL carriers, which typically operate with 15-25 % empty mileage.

Table 8: Driver assignment statistics by phase

	Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
Total loads covered	30	476	343	69	237
Long loads covered	30	0	204	24	103
Short loads covered	0	476	139	45	134
Total drivers used	30	154	204	42	192
Domicile drivers used	30	152	0	42	168
Layover drivers at domicile used	0	2	103	0	0
Layover drivers at another location used	0	0	101	0	24
Tour quality imposed	0.87	0.85	0.75, 0.60	0.65	0
Tour quality obtained	0.92	0.94	0.84	0.76	0.64
Average loaded driving minutes per driver	607.21	617.43	556.26	498.57	425.39
Average empty driving minutes per driver	0	0	82.20	68.57	112.26
Fraction of empty travel	0	0	0.13	0.12	0.20

Table 9 summarizes the run times and solution quality of the different phases. Phase 1 is the only phase in which we do not produce a proven optimal solution.

Table 9: IP Run Times and Solution Quality

Phase	Run Time	Solution Quality
Phase 0	1 mins	N/A
Phase 1	9 mins	Within 1.2%
Phase 2	8 mins	Optimal
Phase 3	6 mins	Optimal
Phase 4	5 mins	Optimal

3.6 Contributions

The main contribution of this chapter is a daily driver planning methodology for regional LTL carriers which takes into account detailed driver decisions such as meet-and-turn usage while taking into account driver constraints such as duty hours and drive time hours. Using meet-and-turns in driver management is a new area of study and as we will discuss in Chapter 3, it can reduce driver requirements significantly. A staged approach for generating tours while using templates to create tours is a new contribution in the area of driver management in trucking. Using templates helps us to model the typical types of duties that a driver may perform and hence helps create driver tours which are easily implemented.

CHAPTER IV

VALUE OF MEET-AND-TURNS IN DRIVER MANAGEMENT

4.1 Introduction

As indicated in the previous chapters, meet-and-turns are used in practice to reduce the number of drivers required to move a set of loads and to reduce the layover costs incurred by drivers resting away from their domicile. A basic meet-and-turn operation occurs when two drivers moving loads in opposite directions on the same leg meet somewhere along the leg and exchange their loads and return to their starting location. A basic meet-and-turn operation is shown in Figure 12. The primary use of such meet-and-turn operations is to enable drivers to get back to their domicile at the end of the day. This reduces layover expenses for the carriers and improves the quality of life for the driver. In this chapter, we establish analytical bounds on the benefits of using meet-and-turns. Furthermore, we design and implement heuristics for the effective use of meet-and-turns when assigning drivers to loads.

We begin by observing that the use of a meet-and-turn affects the drivers involved in two ways: the location where they end up changes and the remaining driving and duty hours upon arrival changes. There are two ways in which drivers d_A located at A and driver d_B located at B can move loads along the leg AB (see Figure 12):

- Driver d_A moves a load from A to B and driver d_B moves a load from B to A.

- Driver d_A and d_B meet at C, exchange their loads and return to their starting location.

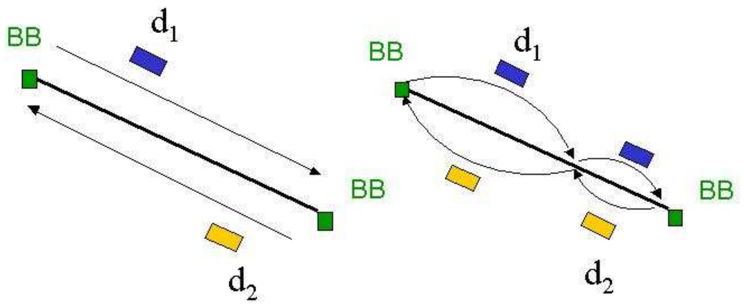

Figure 12: meet-and-turn operation

In both cases, the number of drivers available at A and B after the drivers complete their moves is the same. The use of the meet-and-turn, however, changed the "attributes" of the drivers available at A and B. For example, without a meet-and-turn, driver d_A will be available at B after completing his move, whereas with a meet-and-turn driver d_B will be available at B after completing his move. Also, the drivers will have different remaining drive times when compared to not using the meet-and-turn at C. The relevant attributes of a driver are

- domicile
- drive hours left
- duty hours left

To more clearly illustrate the effects of using a meet-and-turn operation consider Figure 13. Such figures will be used several times in this chapter. Time is presented on the y-axis and increases in the downward direction. This allows us to depict the movement of the drivers not only in terms of geography, but also in terms of time.

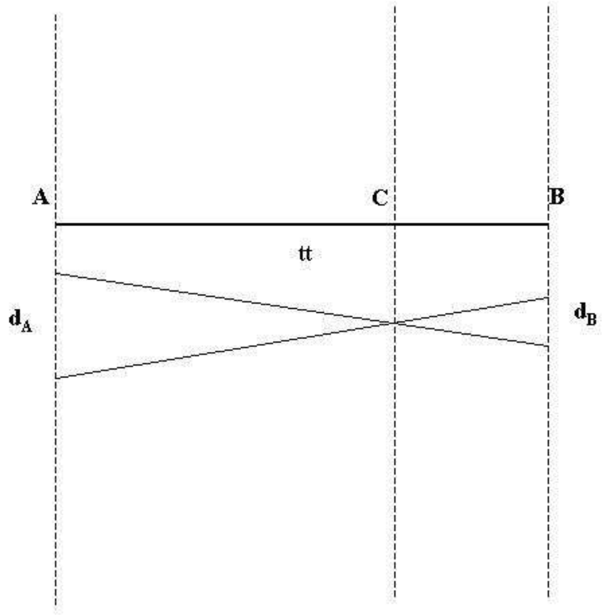

Figure 13: A meet-and-turn operation on a single leg.

We observe that the use of a meet-and-turn operation may affect all three attributes. It is important to note that these three attributes form the basis of any reduction in the number of drivers needed due to the use of meet-and-turns operations.

In the remainder of the chapter, for ease of analysis, we consider only two and three terminal problems, i.e., one and two leg problems. We focus primarily on whether or not the use of meet and turns can reduce the number of drivers required to serve a set of loads. In our analysis, we take into account ready and due times of loads as well as driving and duty time restrictions. We assume that a load can be involved in at most one meet-and-turn operation and that meet-and-turn operations are instantaneous. Note that, without loss of generality, we may assume that a driver does not wait at a meet-and-turn location. Waiting at a meet-and-turn location can always be replaced by waiting at the origin terminal.

Before studying the benefits of meet-and-turns in a few specific situations, we present a number of general observations.

Lemma 4.1.1 *The time at which the exchange of loads takes place in a meet-and-turn operation is completely determined by the dispatch times of the loads.*

Proof Consider the lane shown in Figure 15. Assume that l_A is dispatched at t_A and l_B is dispatched at t_B. Let us assume that the two drivers meet at a location which is drive time x away from A. Because the drivers do not wait at the meet-and-turn location, they have to arrive at the meet-and-turn location at the same point in time. So $t_A + x = t_B + tt - x$, which implies $x = \frac{tt - t_A + t_B}{2}$. ▮

Next, we derive conditions on the dispatch times that must be satisfied if a meet-and-turn is to take place. Since $0 < x < tt$, the following feasibility condition has to be satisfied:

$$|t_A - t_B| < tt. \tag{32}$$

The inequality has to be strict because otherwise one of the terminals becomes the meet-and-turn location. Furthermore, the times t_A and t_B are constrained by the ready time of the loads at the origin terminals, the due times of the loads at the destination terminals and the driver availability. Let l_A have ready and due times r_A and s_A and let l_B have ready and due times r_B and s_B. Also, let driver d_A have duty period $[u_A, v_A]$ and let driver d_B have duty period $[u_B, v_B]$. Then the following constraints need to be satisfied for t_A and t_B:

$$\begin{aligned} r_A \le t_A &\le s_A - tt \\ r_B \le t_B &\le s_B - tt \\ u_A \le t_A &\le v_A - 2x \\ u_B \le t_B &\le v_B - 2(tt - x) \end{aligned}$$

Finally, we observe that the combined drive time of two drivers involved in a meet-and-turn operation is $2tt$. The first driver covers $2x$ and the second driver covers $2(tt - x)$ for a total of $2tt$.

Lemma 4.1.2 *If $tt \leq t_{drive} < 2tt$, then a driver d with remaining drive time less than $2(tt - t_{drive}/2)$ cannot move a load.*

Proof There are two ways in which a driver can move a load:

- Move the load from one end of a leg to the other end the leg. This requires that $tt < 2(tt - t_{drive}/2)$, which implies $tt > t_{drive}$; a contradiction.
- Move the load to a meet-and-turn location and return with another load. This requires that a driver q meets driver d at the meet-and-turn location. Together, the drivers consume $2tt$ time. Let us assume that d consumes t_d and driver q consumes t_q. Then, $t_d + t_q = 2tt$. Since $t_d < 2(tt - t_{drive}/2)$, we must have $t_q > t_{drive}$; a contradiction. ▮

Theorem 4.1.3 *If $tt < t_{drive} < \frac{3}{2}tt$, then two drivers d_A and d_B starting their duty at opposite terminals and performing a meet-and-turn operation can cover the most loads when they use a meet-and-turn location at distance $t_{drive}/2$ from one of the terminals.*

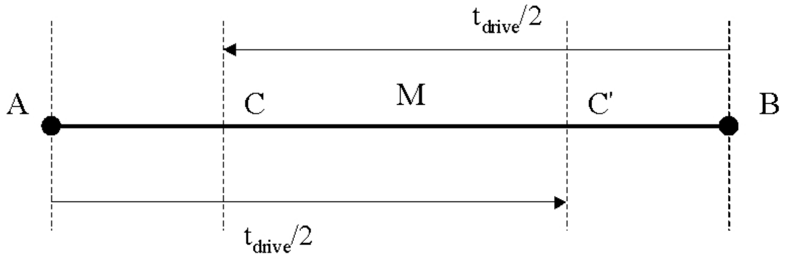

Figure 14: Best meet-and-turns

Proof Consider the situation depicted in Figure 14. Because $tt < t_{drive} < \frac{3}{2}tt$, the drivers will always be able to meet at some meet-and-turn location. Both drivers cover their first load using meet-and-turn location M. We want to show that the drivers can move the maximum number of loads if M is at C or C'.

First, we prove that the maximum number of loads moved is three, i.e., it is not possible that both drivers move more than one load. From Lemma 4.1.2 it follows that a driver needs to have at least a remaining drive time of $2(tt - t_{drive}/2)$ to be able to move a load. If both drivers have to move a load, then both of them have to be able to drive at least $2(tt - t_{drive}/2)$ when they move their second loads. To move the first two loads, the drivers together used a drive time of $2tt$. The total drive time available for the two drivers is $2t_{drive}$. Therefore, we must have $2tt + 4(tt - t_{drive}/2) \leq 2t_{drive}$. For this to happen, t_{drive} has to be greater than or equal to $\frac{3}{2}tt$, which would violate the assumption of the theorem. Therefore, at most three loads can be moved. For that to happen, it is best to choose the meet-and-turn location in such a way that one driver consumes all its available drive time, ensuring that the other driver has the largest possible remaining drive time when he returns to his origin terminal. This implies the use of either meet-and-turn location C or C'. ▮

Next, we formalize the intuitive idea that there is no need to consider meet-and-turn operations when the drivers are unconstrained.

Theorem 4.1.4 *When there are no drive time restrictions, no duty time restrictions, and no rest location restrictions, then no benefits can arise from using meet-and-turn operations.*

Proof Consider the situation depicted in Figure 15. There are two drivers d_A and d_B at terminals A and B both available at time 0. Consider a pair of loads l_A at A and l_B at B and assume that the loads are covered using a meet and turn operation and that their dispatch times are t_A and t_B, respectively. We observe that driver d_A will be available at A at time $t_B + tt$ and that driver d_B will be available at B at time $t_A + tt$. If no meet-and-turn operations is used, then d_A will be available at B at time

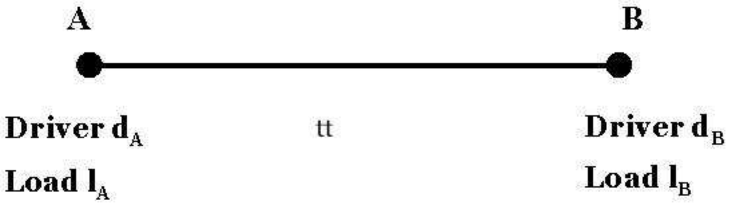

Figure 15: 2 terminal problem

$t_A + tt$ and d_B will be available at B at time $t_B + tt$. Because there are no restrictions limiting drivers, driver d_A and d_B are indistinguishable. Thus, the situation arising after covering loads l_A and l_B using a meet and turn operation can also be reached without using a meet-and-turn operation. Consequently, there is no advantage to using meet and turn operations. ▮

Theorem 4.1.4 indicates that drivers have to be restricted in some way before meet-and-turn operations may have benefits. In the remainder of the chapter, we focus on the situation in which the drive time of a driver is restricted, but there are no duty time limits and rest location restrictions.

4.2 Drive time restrictions

In this section, we analyze the potential benefits of using meet and turns in terms of the number of drivers required to cover a given set of loads when drivers can drive for at most t_{drive}. That is, we consider the situation in which a driver once he has used its available drive time (t_{drive}) can never be used again; the driver does not renew his available drive time by resting. We show that the benefits of using meet-and-turns depends on the length tt of the lane.

4.2.1 Lanes with $t_{drive} < tt$

None of the lanes can be covered irrespective of whether we use meet-and-turn operations or not.

4.2.2 Lanes with $t_{drive} = tt$

When $t_{drive} = tt$, there is no benefit to using a meet and turn operations. Since, the total drive time required to do a meet and turn operation would always be $2tt$, in this case if a meet and turn is used, each driver will have to drive exactly tt because they cannot drive more than tt since $tt = t_{drive}$. So at the end of a trip, each driver will be back at his domicile with no drive time available, which is also the result when the two drivers do not meet but go to their destinations. In the latter case, the drivers available at the terminals A and B will be different from the previous one, but that does not make a difference since in both cases the drivers are out of drive time and have to go to rest.

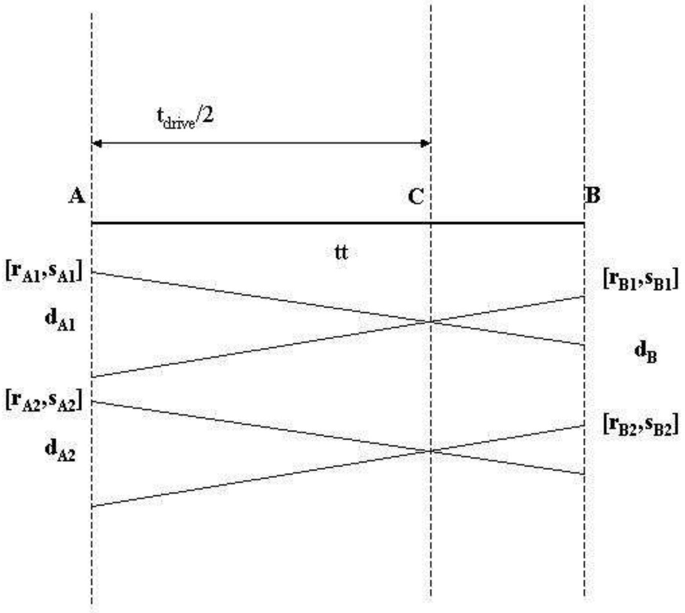

Figure 16: 2 terminal problem with 4 loads

4.2.3 Lanes with $tt < t_{drive} < 4tt/3$

Note that because $t_{drive} < 2tt$, a driver will not be able to cover two loads by himself. Consider the situation depicted in Figure 16, which represents the "best case scenario." We show that even in this case, no benefits arise from using meet and turn operations. Driver d_{A1} uses his full drive time to get to the meet-and-turn location C and back. So the meet-and-turn location is $t_{drive}/2$ away from A. Driver d_B has to travel $tt - t_{drive}/2$ one way which makes $2tt - t_{drive}$ total. The drive time left for d_B after this is $2t_{drive} - 2tt$. For d_B to be useful for at least one more load d_B has to be able to travel at least to C (and back) such that a new driver d_{A2} available at A can carry load l_{A2} to C and exchange with d_B. So that would mean that $2t_{drive} - 2tt$ has to be greater than or equal to $2tt - t_{drive}$. But that would require $t_{drive} > 4tt/3$ which is not true in this case. d_{A2} will carry l_{A2} but l_{B2} will not be covered.

The same situation also occurs when we do not use a meet-and-turn. Driver d_A covers load l_{A1} and d_B covers load l_{B1} and both of them cannot be used for any other purpose later on. Driver d_{A2} will carry l_{A2} but l_{B2} will not be covered.

4.2.4 Lanes with $4tt/3 \leq t_{drive} < 3tt/2$

Because $t_{drive} < 2tt$, a driver will not be able to cover two loads by himself. Furthermore, because $tt > \frac{1}{2}t_{drive}$, if meet-and-turn operations are not considered all loads have to be covered by a different driver. Consider, again, the situation depicted in Figure 16, which represents the "best case scenario." Since $4(tt - \frac{1}{2}t_{drive} \leq 3t_{drive} - 2t_{drive} = t_{drive}$, a single driver can perform two "short turns" of a meet-and-turn operation. As a result, the four loads can be covered with 3 drivers when meet-and-turn operations are used, but require 4 drivers when no meet-and-turn operations are used.

4.2.5 Lanes with $3tt/2 \leq t_{drive} < 2tt$

Type $A - B - A$ tours will not be feasible in this case either. The advantage in this case occurs because one driver uses his full drive time to use the meet-and-turn and the other driver goes back having enough drive time left to cover the whole length tt once. The example demonstrated in the previous subsection will be feasible in this

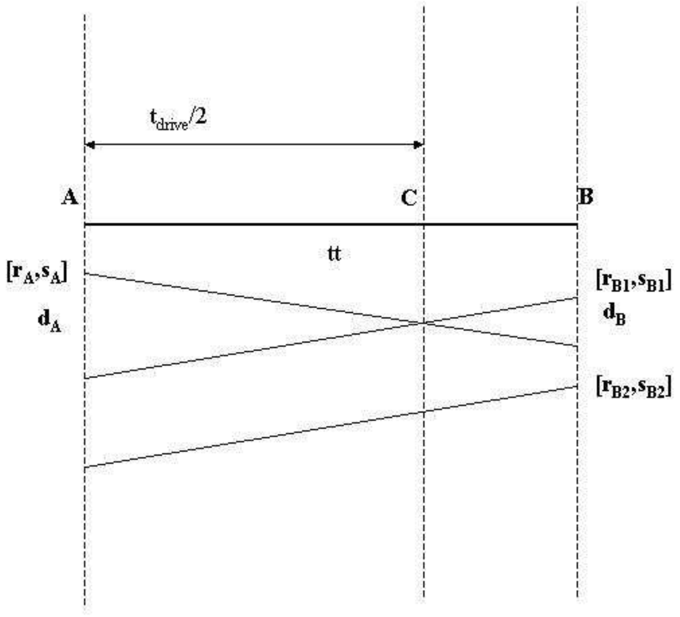

Figure 17: A 2-terminal problem with 3 loads

case also, but here we can use a simpler example using 3 loads and 2 drivers as shown in Figure 17.

Without using the meet-and-turn, all the 3 loads will not be feasibly covered because there are 2 drivers and drivers cannot do $A - B - A$ type tours.

Driver d_A will carry l_A till C. Driver d_B will carry l_{B1} till C. Drivers d_A and d_B will exchange loads and return. Now, d_B will carry l_{B2}. d_A uses exactly t_{drive} hours of driving. d_B uses $2tt - t_{drive} + tt$ which is less than or equal t_{drive} because $3tt \leq 2t_{drive}$.

4.2.6 Lanes with $2tt \leq t_{drive} < 3tt$

The situation we present here is somewhat different from the situations above. Because $2tt \leq t_{drive}$, a driver may be able to cover two loads without the use of meet-and-turn operations, but it depends on the origins of the loads. Consider the situation depicted in Figures 18 and 19. The first figure shows how four loads can be covered with two drivers if a meet and turn operation is used. The second figure shows that without the use of a meet-and-turn, three drivers will be needed. In this example, there are three loads going from A to B and only one load going from B to A. In a sense, the use of a meet-and-turn operation avoids empty relocation.

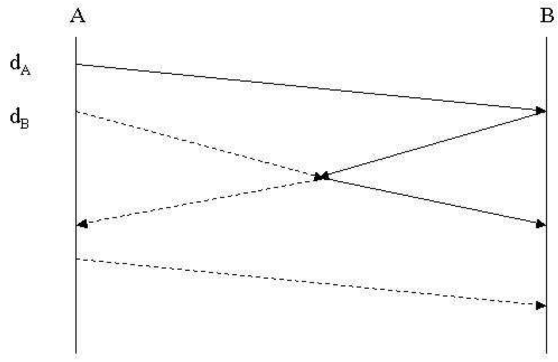

Figure 18: 2 terminal problem with meet-and-turns $2tt \leq t_{drive} < 3tt$

4.2.7 Best Case Analysis

In the previous section, we have given examples where the use of meet-and-turn operations provides advantageous. Next, we bound the maximum benefit of the use of meet-and-turn operations, i.e., the maximum number of drivers that can be saved using meet-and-turn operations when covering a given set of loads.

The bound depends on the length of the lane. Let n_{mt} denote the number of drivers required to cover the given set of loads when meet-and-turn operations are allowed and n_d denote the number of drivers needed when meet-and-turn operations are not allowed.

Observation 1 Given a set of loads L, the minimum number of drivers required to cover the loads is greater than or equal $\lceil \frac{|L|tt}{t_{drive}} \rceil$.

For lanes with $tt < t_{drive} < 2tt$, a slightly stronger statement can be made.

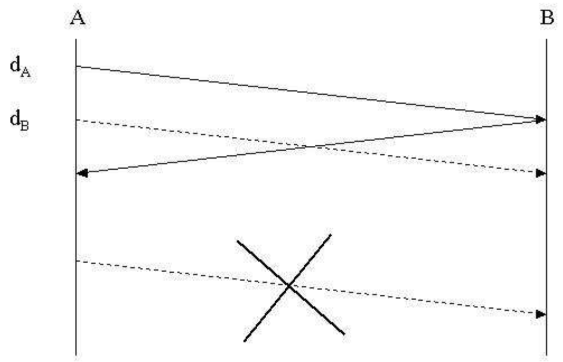

Figure 19: 2 terminal problem without meet-and-turns $2tt \leq t_{drive} < 3tt$

Observation 2 Given a set of loads L, a set of drivers D, and a lane with length $tt < t_{drive} < 2tt$, a driver assignment in which all loads are covered and in which all but one of the drivers use their drive time t_{drive} and the remaining drive time of the other driver is insufficient to reach the nearest meet-and-turn location is best possible.

Lanes with $t_{drive} < 4tt/3$

There is no benefit from using meet-and-turn operations.

Lanes with $4tt/3 \leq t_{drive} < 3tt/2$

Each driver can do a maximum of two trips to the meet-and-turn location, which leads to the example shown in Figure 20. Observation 2 shows that such an example is the best possible, i.e., in the best case $n_{mt} = 3/4n_d$.

Lanes with $3tt/2 \leq t_{drive} < 8tt/5$

Each driver can do a maximum of three trips to the meet-and-turn location, which leads to the example shown in Figure 21. In the best case, we have $n_{mt} = 4/6n_d$.

Lanes with $8tt/5 \leq t_{drive} < 10tt/6$

Each driver can do a maximum of four trips to the meet-and-turn location, which leads to the example shown in Figure 22. In the best case, we have $n_{mt} = 5/8n_d$.

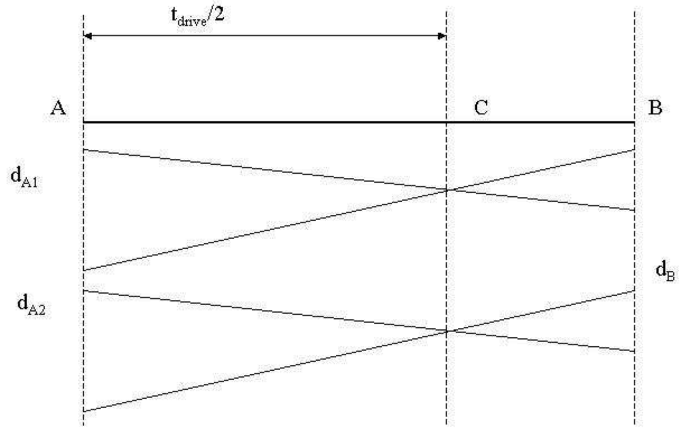

Figure 20: Best case scenario for $4tt/3 \leq t_{drive} < 3tt/2$

The pattern is clear. We end with the situation in which k trips to the meet-and-turn location are possible.

Lanes with $2ktt/(k+1) \leq t_{drive} < 2(k+1)tt/(k+2)$

Each driver can do a maximum of k trips to the meet-and-turn location, which leads to the example shown in Figure 23. In the best case, we have $n_{mt} = \frac{(k+1)}{2k} n_d$. As k grows, the ratio $\frac{n_{mt}}{n_d} = \frac{k+1}{2k}$ tends to a $\frac{1}{2}$.

The above shows that there are examples where the benefit of using meet-and-turn operations is 100%, the number of drivers required is reduced by a factor 2. Next, we show that this is indeed the maximum possible reduction.

Theorem 4.2.1 *The number of drivers n_d required to cover a given set of loads L on a single leg without using meet-and-turn operations is never more than twice the number of drivers n_d required to cover L when meet-and-turn operations are being used, i.e.,*

$$\frac{n_d}{n_{mt}} \leq 2$$

Proof Consider a solution covering the loads in L with a minimal number of drivers

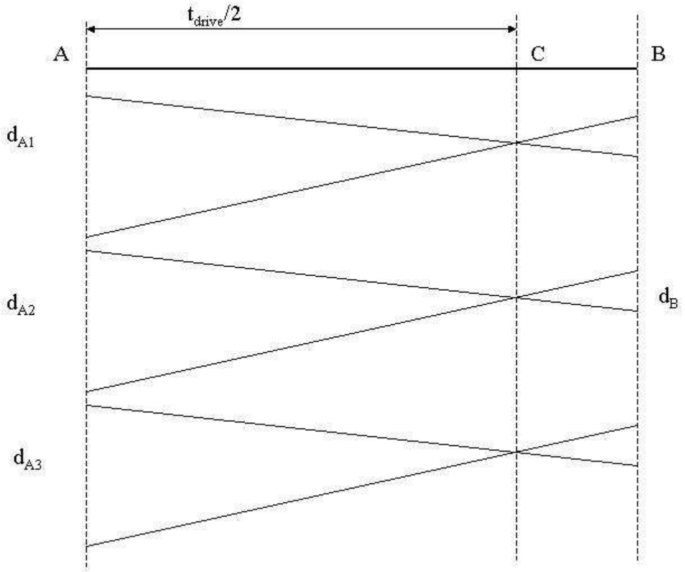

Figure 21: Best case scenario for $3tt/2 \leq t_{drive} < 8tt/5$

(n_{mt}), i.e., a solution in which meet and turn operations are allowed. Loads can be divided into two categories: loads involved in a meet-and-turn operation and loads not involved in a meet-and-turn operation. We show that a solution without meet and turn operations can be constructed that uses no more than $2n_{mt}$ drivers. For each driver that covers a "long turn" of a meet-and-turn operation, create a "clone." Obviously, the clone can handle the "short turn" of the meet-and-turn operation. There are three types of drivers:

- Drivers covering long turns of meet-and-turn operations, but no loads that are not involved in meet-and-turn operations - Clearly, the drivers and their clones can cover the same loads without meet-and-turn operations; see Figure 24.
- Drivers covering long turns of meet-and-turn operations as well as loads that are not involved in meet-and-turn operations - Again, the drivers and their clones can cover the same loads without meet-and-turn operations; see Figure 25. In this case, one of the drivers will have to drive empty when the other driver is covering a load not involved in a meet-and-turn operation.

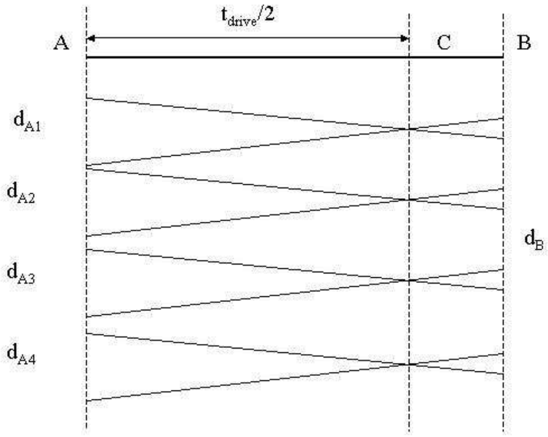

Figure 22: Best case scenario for $8tt/5 \leq t_{drive} < 10tt/6$

- Drivers covering only loads that are not involved in meet and turn operations - No need for clones.

We see that all the loads can be covered without meet-and-turn operations no driver has been cloned more than once. Hence, when meet-and-turns are not allowed the number of drivers cannot more than double. ▌

We elaborate a little on the above theorem by providing an alternate tight situation where the driver who does the small turns of the meet-and-turn operations also moves a load across the entire leg as shown in Figure 26. The drivers involved in each of the larger parts use up all of their driving time (t_{drive}).

Let $m = \frac{t_{drive}-(\frac{t_{drive}}{2}+\epsilon)}{2\epsilon} = \frac{\frac{t_{drive}}{2}-\epsilon}{2\epsilon}$. The number of drivers required for the case where meet-and-turn operations are employed is

$$n_{mt} = m + 1 = \frac{t_{drive}}{4\epsilon} - \frac{1}{2} + 1 = \frac{t_{drive}}{4\epsilon} + \frac{1}{2}. \tag{33}$$

The number of drivers required for the case where meet-and-turn operations are not allowed is

$$n_d = 2m + 1 = 2(\frac{t_{drive}}{4\epsilon} - \frac{1}{2}) + 1 = \frac{t_{drive}}{2\epsilon}. \tag{34}$$

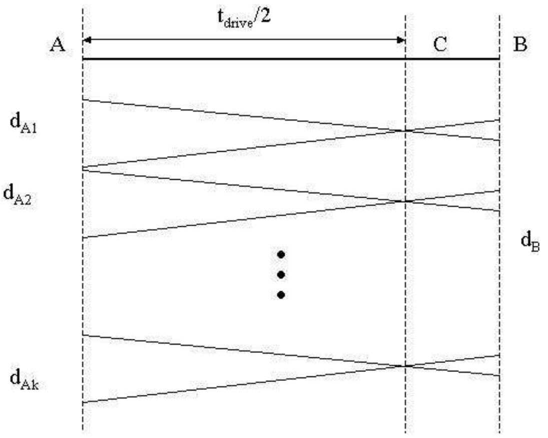

Figure 23: Best case scenario for $2ktt/(k+1) \leq t_{drive} < 2(k+1)tt/(k+2)$

The ratio is

$$\frac{n_d}{n_{mt}} = \frac{\frac{t_{drive}}{2\epsilon}}{\frac{1}{2} + \frac{t_{drive}}{4\epsilon}} == \frac{\frac{2t_{drive}}{2\epsilon}}{1 + \frac{t_{drive}}{2\epsilon}}. \tag{35}$$

In the limit this ratio approaches two,

$$\lim_{\epsilon \to \infty} \frac{n_d}{n_{mt}} = \lim_{\epsilon \to \infty} \frac{2}{1 + \frac{2\epsilon}{t_{drive}}} = 2. \tag{36}$$

4.2.8 Best case analysis for the two-leg case

Conjecture 4.2.2 *The number of drivers required to carry a set of loads when meet and turns are not allowed n_d is at worst less than twice the number of drivers required when meet-and-turns are allowed n_m for the case where there are two legs in the network.*

Even though, we do not have a proof for the above conjecture, we can show that it is true under certain conditions. Proving that it is not possible to ever achieve a larger benefit is more difficult in this case than it was in the single leg case. The difficulty arises because it appears as if a driver may have to be cloned on both legs. Consider, for example, the situation depicted in Figure 27. Drivers at B (the middle terminal)

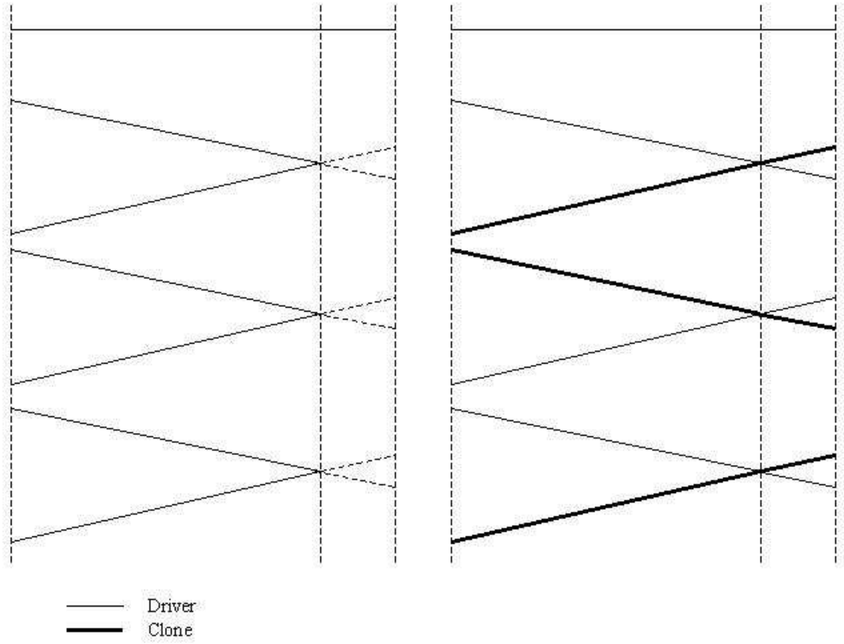

Figure 24: Driver and clone can cover all loads

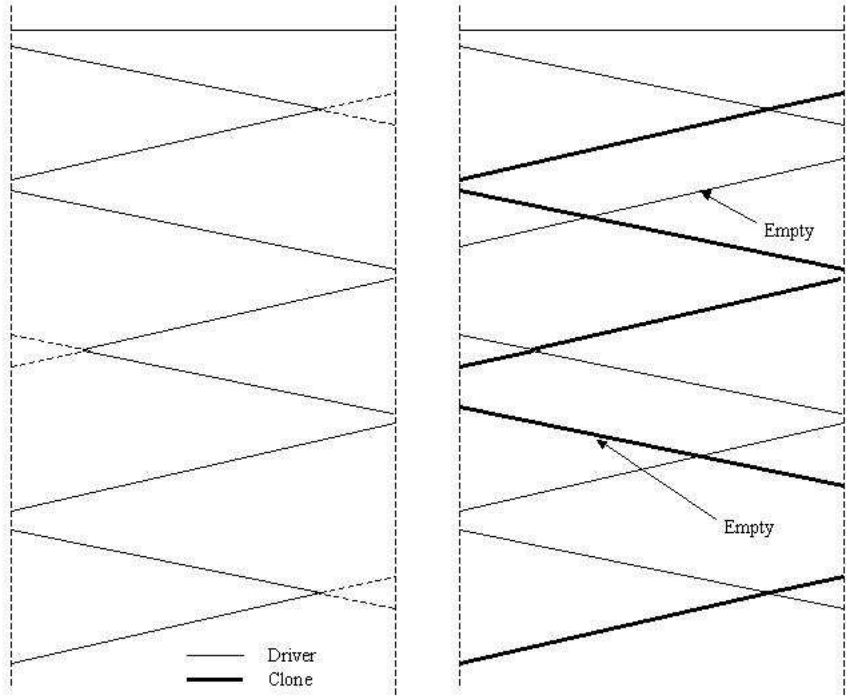

Figure 25: Driver and clone can cover all loads with directs

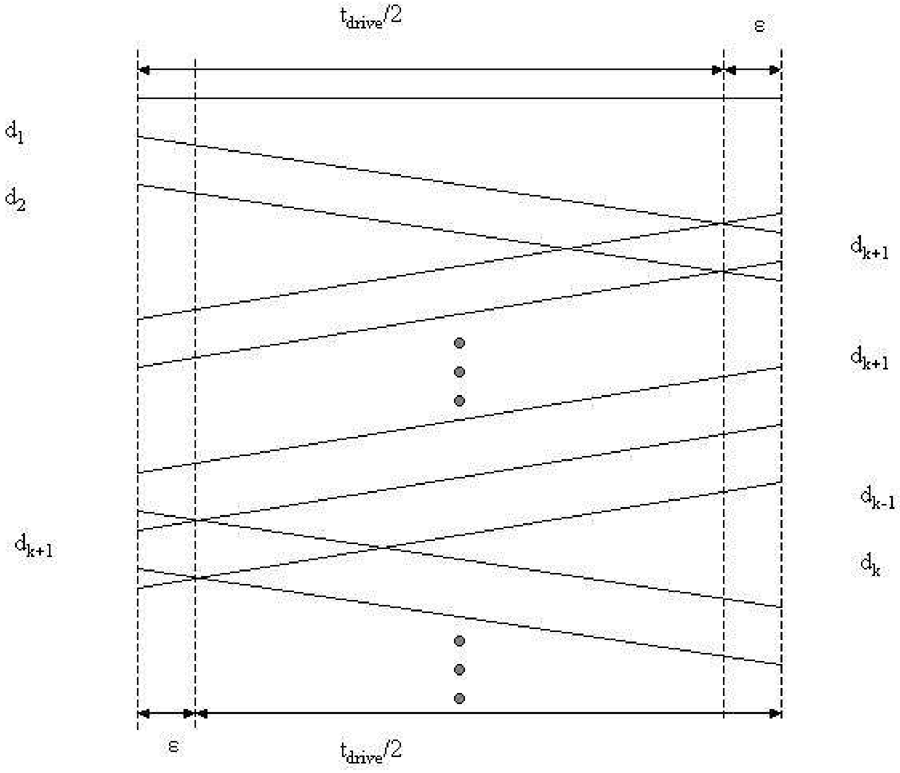

Figure 26: Alternate tight situation

perform two long turns, one on each leg, and the drivers at A and C perform only short turns. Again, a benefit of 100% is realized when meet and turn operations are considered.

We will consider situation depicted in Figure 27 and consider conditions under which the example will indeed produce a tight bound.

First, it is important to note that in such an example both the legs cannot be long ($tt > \frac{t_{drive}}{2}$). If both the legs are long, then the driver wont be able to do large meet-and-turn operations on both sides. There are two cases to consider:

- both legs are short, and
- one leg is short and the other long.

Both legs are short

When the legs are short, two drivers can take care of at least two meet-and-turn operations on each leg. Let us assume that there are m meet-and-turn operations on each leg. The number of drivers needed when meet-and-turns are allowed is $m + 2$. When meet and turns are not allowed, two drivers can take care of two meet and turn operations. So, the number of drivers on each leg is $2\lceil\frac{m}{2}\rceil$. So, the total drivers required is $4\lceil\frac{m}{2}\rceil$ which is less than or equal to $4\frac{(m+1)}{2} = 2(m + 1)$ which is less than $2(m + 2)$. Hence the number of drivers required, n_d, is strictly lesser than twice n_{mt}. We will not have a tight example in this case.

One leg is short and one leg is long

When one leg is short and the other is long, then we will have a tight case. Since a driver can do the larger operation on the longer leg as well as the larger operation on the shorter leg but cannot do the longer operation on the long leg and the shorter operation on the short leg, hence $l_1 > \epsilon_1 > l_2 > \epsilon_2$. So, we can never have $\epsilon_1 = \epsilon_2$ in this case. The tight case will involve the driver doing the smaller operations on the long leg (ϵ_1) exhausting his driving hours. This would mean that since $\epsilon_2 < \epsilon_1$ and

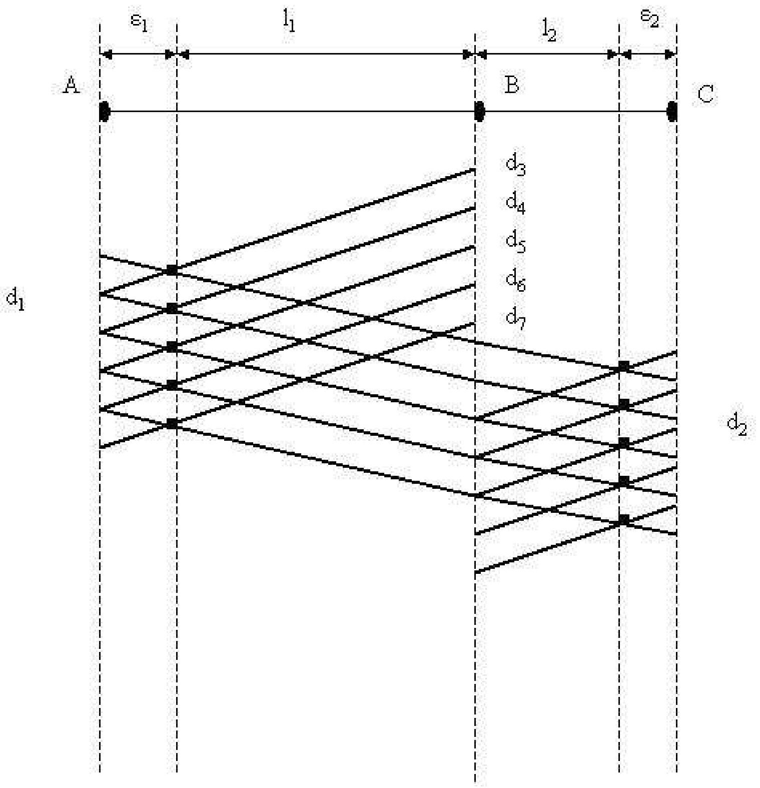

Figure 27: Best case scenario on two legs

$l_2 < \epsilon_1$, $\epsilon_2 + l_2 < 2\epsilon_1$, so two drivers would be able to do all the operations on the shorter leg without exhausting their driving hours. On the longer leg each operation would require two drivers. Hence, the number of drivers required will be exactly twice the original number used.

4.3 An Integer Programming Formulation

We will present an integer programming formulation for finding the minimum number of drivers required to cover a set of loads L with fixed dispatch time t_l ($l \in L$) on a single leg of length tt. Drivers can driver for at most t_{drive} and are not allowed to drive empty. Meet-and-turn operations may be exploited, but a load can be involved in at most one meet-and-turn operations.

For ease of notation, we define the set of operations (or pieces of work) as all the possible operations a driver can perform, i.e., covering a load without a meet-and-turn operation, covering the long turn of a meet-and-turn operation, and covering the short turn of a meet-and-turn operation.

Let O be the set of possible operations, let $D = 1 \ldots n$ be the set of drivers, and let $a_{o,l}$ indicate whether load l is part of operation o as a load not involved in a meet in turn operation ($a(o,l) = 1$), as a load involved in a meet in turn operation ($a(o,l) = \frac{1}{2}$), or not ($a(o,l) = 0$). Furthermore, let tr_o denote the duration of operation o and let q_{o_1,o_2} indicate whether o_1 and o_2 have to be done together ($q_{o_1,o_2} = 1$), i.e. they are part of the same meet and turn operation.

We introduce the following binary decision variables: $x_{d,o}$ equal to 1 if driver d covers load o, and $y^d_{o_1,o_2}$ equal to 1 if driver d does operation o_2 immediately after operation o_1, i.e., without any other operations in between o_1 and o_2. We will only generate y variables for which operations o_1 and o_2 can indeed be performed in sequence. This includes checking for time feasibilities and whether the destination of the first operation is the same as the origin of the second. This will be done as part of the preprocessing. Finally, z_d equal to 1 if driver d is used.

The objective function is to minimize the number of drivers used:

$$\sum_{d \in D} z_d \tag{37}$$

Various sets of constraints ensure a feasible solution:

- Each load has to be covered.

$$\sum_{d \in D} \sum_{o \in O} x_{d,o} a_{o,l} = 1 \qquad \forall l \in L \tag{38}$$

- Driver cannot drive for more than t_{drive} hours

$$\sum_{o \in O} x_{d,o} tr_o \leq t_{drive} \qquad \forall d \in D \tag{39}$$

- Not more than one operation can follow another operation on the driver's schedule.

$$\sum_{o_2 \in O} y^d_{o_1,o_2} \leq 1, \qquad \forall o_1 \in O, d \in D \tag{40}$$

- Not more than one operation can precede another operation on the driver's schedule.

$$\sum_{o_1 \in O} y^d_{o_1,o_2} \leq 1, \qquad \forall o_2 \in O, d \in D \tag{41}$$

- Two parts of a meet-and-turn operation have to be done together

$$\sum_{d \in D} x_{d,o_1} = \sum_{d \in D} x_{d,o_2} \qquad \forall o_1 \in O, o_2 \in O | q_{o_1,o_2} = 1 \tag{42}$$

- If an operation follows another operation on a driver's schedule, then both operations have to be covered by the driver

$$2y^d_{o_1,o_2} \leq x_{d,o_1} + x_{d,o_2} \qquad \forall d \in D, o_1 \in O, o_2 \in O \tag{43}$$

- Link x- and y-variables

$$\sum_{o_1 \in O} \sum_{o_2 \in O} y^d_{o_1,o_2} = \sum_{o \in O} x_{d,o} - 1 \qquad \forall d \in D \tag{44}$$

- Operations can only be performed when a driver is used

$$x_{d,o} \leq z_d \qquad \forall d \in D, o \in O \tag{45}$$

4.3.1 Implementation

The model was implemented using Xpress-Mosel. Several tricks were employed to speed up the solution process.

- A simple lower bound on the number of drivers was introduced ($\lceil \frac{Ltt}{t_{drive}} \rceil$).
- To eliminate symmetry, constraints were added to ensure that Driver 1 drives more than Driver 2, and so on.

4.4 A heuristic for assigning drivers for the one leg problem with meet-and-turns

We develop a heuristic based on approaches to solve the bin packing problem. The drive time allowed for a drive (t_{drive}) is like a bin of size t_{drive} and we have to assign operations to it. This is the same idea as bin packing. However, items (i.e., operations) may not fit into a bin (driver's schedule) because of reasons other than size of item. Each operation has the following information:

- Total travel time involved
- Start time
- Beginning terminal
- Ending terminal
- Type of operation, i.e., meet and turn operation or a direct load
- If meet-and-turn operation, then there is also an associated partner operation

The heuristic first creates all possible operations and then sorts them in decreasing order of size and uses a best fit approach to assign them to drivers. Only one driver is assigned at a time and even for a meet-and-turn operation, one part is assigned to a driver first. We ensure that the same driver does not get assigned the other part but we also ensure that the other part is covered by some driver. The heuristic described in Algorithm 8.

4.4.1 Performance guarantee for the Single Driver Heuristic

We will assume that we have a problem with L loads and we can use as many drivers as required. We know that for any problem, the number of drivers required will be less than or equal to L. We also know that the best case would be when each driver uses exactly t_{drive} amount of time. The lower bound for the number of drivers used

Algorithm 8 Single Driver Heuristic

```
Sort operations according to decreasing order of size
Cut Time of all dispatches = -1
for All operations in the sorted list do
  if Operation is allowed, ie. it has not been previously deleted then
    for All drivers do
      if Operation fits in the driver's schedule and driver is allowed to do operation then
        if Remaining drive time on driver's schedule after inserting operation is smaller than current minimum then
          Make this driver the driver which fits the operation best
        end if
      end if
    end for
    Assign operation to driver with best fit
    Delete all other operations which correspond to loads part of chosen operation
    if Chosen operation is part of a meet-and-turn then
      Other part of meet-and-turn cannot be done by same driver
    end if
  end if
end for
```

will be $\lceil \frac{Ltt}{t_{drive}} \rceil$. If we denote n^* as the number of drivers required in the optimal case and n^H as the number of drivers required with the heuristic solution. We know that $n^* \geq \lceil \frac{Ltt}{t_{drive}} \rceil$ and $n^H \leq L$, and thus $\frac{n^H}{n^*} \leq \frac{L}{\lceil \frac{Ltt}{t_{drive}} \rceil}$.

We will do the analysis for the longer legs, i.e., with $tt > \frac{t_{drive}}{2}$, because these are the cases where meet and turns are most useful. It was shown that the benefit of using a meet-and-turn is dependent on the length of the lane. We assume here that the lower bound for n^* is $\lceil \frac{Ltt}{t_{drive}} \rceil$. However, the total benefit of using meet and turns has been discussed in previous sections. So for different lane lengths, the benefits are different. The savings can be found in Table 10.

We will denote the worst case ratio of $\frac{n^H}{n^*}$ by α_H. We know that $\alpha_H \leq \frac{L}{\lceil \frac{Ltt}{t_{drive}} \rceil}$. We will show examples where α_H is exactly equal to the above bounds. Let us look at examples where t_{drive} is 10 hours. In Figs 30 and 31, the example with 4 loads, the optimal solution uses 3 drivers while the heuristic solution uses 4 drivers. In Figs 28

Table 10: Meet-and-turn benefit by lane length

Lane Length (tt)	Savings (β_{tt})
$\frac{3}{4}t_{drive} \leq tt \leq t_{drive}$	1
$\frac{4}{6}t_{drive} \leq tt \leq \frac{3}{4}t_{drive}$	$\frac{3}{4}$
$\frac{5}{8}t_{drive} \leq tt \leq \frac{4}{6}t_{drive}$	$\frac{4}{6}$
$\frac{6}{10}t_{drive} \leq tt \leq \frac{5}{8}t_{drive}$	$\frac{5}{8}$
$\vdots$	$\vdots$
$\frac{k+2}{2k+2}t_{drive} \leq tt \leq \frac{k+1}{2k}t_{drive}$	$\frac{k+1}{2k}$

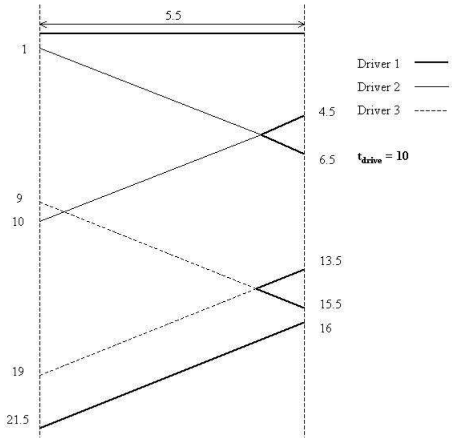

Figure 28: Optimal solution for example with 5 loads

and 29, the example with 5 loads, the optimal solution again uses 3 drivers while the heuristic solution uses 5 drivers. This is because the heuristic selects the operation with the largest meet-and-turn first.

But we know for example that any solution with $tt > \frac{3}{4}t_{drive}$ uses exactly L drivers because there is no benefit from using meet-and-turns. We have to factor into the ratio, the fact that the benefit from using meet-and-turns has discrete values. Therefore we can claim that the ratio α_H for a given value of tt and L is given by $\alpha_H \leq \beta_t t \frac{L}{\lceil \frac{Ltt}{t_{drive}} \rceil}$. This is not a tight bound since this goes to 2 as tt gets close to 5 and we do not have an example where the heuristic uses twice as many drivers as the optimal.

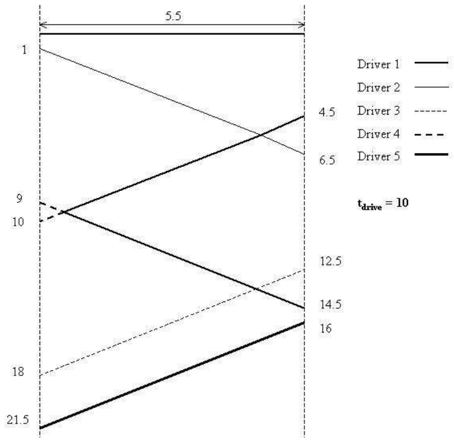

Figure 29: Heuristic solution for example with 5 loads

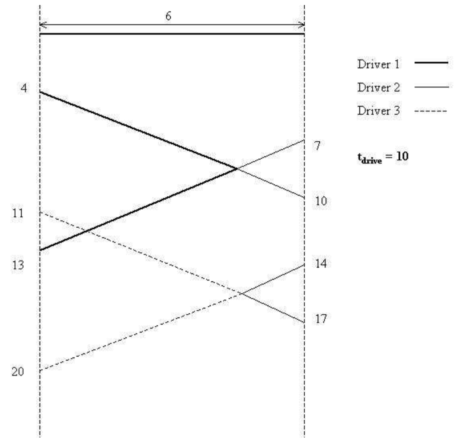

Figure 30: Optimal solution for example with 4 loads

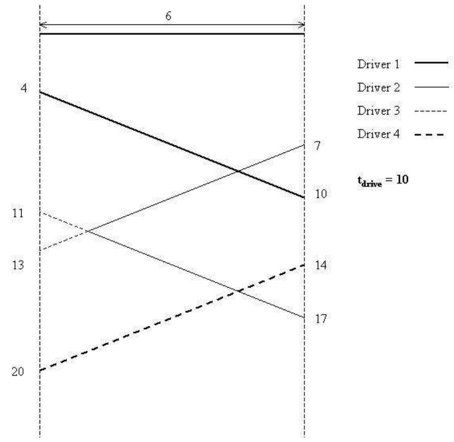

Figure 31: Heuristic solution for example with 5 loads

4.5 Computational Study

We have conducted a short computational study to analyze the performance of the integer program, in terms of the size of instances that can be solved, and the heuristic, in terms of the quality of solutions produced. We created random test instances with an equal number of loads at the terminals at the endpoints of the leg. The dispatch times of the loads at a terminal are distributed uniformly across a specified time period. The length of the time period is set to half the number of loads being dispatched. We assume $t_{drive} = 10$ for all the instances.

We perform three types of analyses. First, we analyze the impact of the use of meet-and-turn operations on driver requirements, i.e., the total number of drivers required to serve a given set of loads. We examine how the lane length affects the importance of meet and turn operations. More specifically, we solve 100 instances for each lane length from 1 to 10 with 16 loads, 8 dispatched from each terminal. The lane length plus the ratio of the number of drivers required when meet-and-turn operation are allowed and the number of drivers required when meet-and-turn operations are not allowed are shown in Table 11.

Table 11: Benefits of meet-and-turn operations for different lane lengths

Lane length	Average benefit of meet-and-turn operations
1	1
2	1
3	0.96
4	0.92
5	0.88
6	0.91
7	0.96
8	1
9	1
10	1

We observe that the most benefits are achieved for lane lengths around half the total drive time. This matches with the analytical results obtained earlier, where we proved that the most benefits are obtained when the lane length is just over $\frac{t_{drive}}{2}$ and that the benefits decrease as the lane length approaches t_{drive}. Also note than when the lane length is 5 ($\frac{t_{drive}}{2}$), we reduce the number of required drivers by about 12%; far less than the maximum possible reduction of 50%.

Next, we analyze the performance of the integer program and the heuristic. As before, we create instances with an equal number of loads departing from both terminals. Having an equal number of loads departing from both terminals increases the opportunities for performing meet-and-turn operations, but, as a result, it also increases the the number of integer variables (and constraints) in the integer program. We vary the size of the instances, in terms of number of loads having to be dispatched, and compare the performance of the integer program and the heuristic. We solve 100 instances for a given size and present the average optimality gap and the average run time. We limit the run time for the integer program to 500 seconds. We compute the optimality gap for the heuristic based on the lower bound generated by the integer program. The results are shown in Table 12. The integer program becomes very hard to solve when the number of loads is more than 20. The heuristic is, of course, much faster and performs well.

Table 12: Run times and optimality gaps for the integer program and the heuristic

Number of loads	Run time (IP)	Optimality Gap (IP)	Run time (H)	Optimality Gap(H)
10	56.5s	Opt	12.3s	Opt
15	63.2s	Opt	13.6s	1.3%
20	132.7s	1.2%	14.9s	1.6%
30	500s	5.21%	17.8s	1.8%
40	500s	2.79%	21.2s	1.7%
50	500s	3.45%	24.6s	1.5 %

The heuristic continues to produce high quality solutions even for large instances, see Table 13. For instances with 10, 20 and 50 loads, we solve 100 instances to calculate the average run times and optimality gaps; for instances with 100, 150 and 200 loads, we solve only three instances to calculate the average run times and optimality gaps. We observe that the heuristic is quite fast and produces good solutions. The optimality gap remains only a few percentage points, and this may be due more to the quality of the lower bound than the quality of the heuristic solution.

Table 13: Run times and optimality gaps for the heuristic

Number of loads	Run Time (s)	Optimality Gap
10	12.3s	Opt
20	13.6s	1.3%
50	24.6s	1.5 %
100	37.9	2.8 %
150	48.6	4.2 %
200	57.8	4.3 %

4.6 Contributions

This chapter has provided a comprehensive study of the benefits of meet-and-turn usage and provided some theoretical bounds on the benefits. An Integer Programming formulation and a heuristic for meet-and-turn usage in driver planning are also new contributions in the area of meet-and-turn planning.

REFERENCES

[1] Barnhart, C., Johnson, E. L., Nemhauser, G. L., and Vance, P. H., "Crew scheduling," in *Handbook of Transportation Science* (Hall, R. W., ed.), Kluwer Academic Publishers, 1999.

[2] Barnhart, C., C. A. J. E. L. K. D. N. G. L. and Vance, P., "Airline crew scheduling," in *Handbook of Transportation Science* (Hall, R. W., e., ed.), pp. 517–560, 2003.

[3] Caliskan, C. and Hall, R. W., "A dynamic empty equipment and crew allocation model for long-haul networks," *Transportation Research Part A*, vol. 37, no. 7, pp. 405–418, 2003.

[4] Cohn, A., R. S. W. A. and Mohr, D., "Integration of the load matching and routing problem with equipment balancing for small package carriers," *Techincal Report number 05-04*, 2005.

[5] Crainic, T. G. and Roy, J., "Design of regular intercity driver routes for the ltl motor carrier industry," *Transportation Science*, vol. 26, no. 4, pp. 280–295, 1992.

[6] Crainic, T. G., "Long-haul freight transportation," in *Handbook of Transportation Science* (Hall, R. W., ed.), Kluwer Academic Publishers, 1999.

[7] Elhallaoui, I., V. D. S. F. and Desaulniers, G., "Dynamic aggregation of set-partitioning constraints in column generation," *Operations Research*, vol. 53, pp. 632–645, 2005.

[8] Farvolden, J. M. and Powell, W. B., "Subgradient methods for the service network design problem," *Transportation Science*, vol. 28, no. 3, 1994.

[9] Godfrey, G. and Powell, W. B., "An adaptive dynamic programming algorithm for single-period fleet management problems i: Single period travel times," *Transportation Science*, vol. 36, pp. 21–39, 2002.

[10] Powell, W. B., S. J. A. and Simao, H. P., "An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem," *Transportation Science*, vol. 36, pp. 231–249, 2002.

[11] Powell, W. B., "A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers," *Transportation Science*, vol. 30, no. 3, pp. 195–219, 1996.

[12] Powell, W. B. and Koskosidis, I. A., “Shipment routing algorithms with tree constraints,” *Transportation Science*, vol. 26, no. 3, 1992.

[13] Powell, W. B., “A local improvement heuristic for the design of less-then-truckload motor carrier networks,” *Transportation Science*, vol. 20, no. 4, 1986.

[14] Schaefer, A. J., J. E. L. K. A. J. and Nemhauser, G. L., “Airline crew scheduling under uncertainty,” *Transportation Science*, vol. 39, pp. 340– 348, 2005.

[15] Yang, J., J. P. and Mahmassani, H. S., “Real-time multivehicle truckload pickup and delivery problems,” *Transportation Science*, vol. 38, p. 135148, 2004.

VITA

Prashant Warier was born in Bhilai, India on October 29, 1979. He received his Bachelor's degree in Manufacturing Science and Engineering from the Indian Institute of Technology, Delhi in 2001. He received his Master's degree in Operations Research from the School of Industrial and Systems Engineering at Georgia Tech in 2003. Prashant is presently working for SAP in Scottsdale, AZ. His primary research interests are in Transportation and Logistics and in Price Optimization.