Refined Execution Cost Estimation
for LTL Load Plans

Alan Erera, Michael Hewitt, Martin Savelsbergh, Yang Zhang
School of Industrial and Systems Engineering
Georgia Institute of Technology

Abstract

A load plan specifies how freight is routed through a linehaul terminal network operated by a less-than-truckload (LTL) motor carrier. Determining the design of the load plan, a problem of service network design, is critical to effective operations of such carriers. Operations research techniques currently support load plan design, and all models in use or proposed today approximate transportation costs by using costs per trailer dispatched between terminals; often, these costs are determined by multiplying a per trailer-mile cost by the mileage between terminals. Furthermore, empty transportation costs are determined by solving a trailer re-balancing problem, either within the load planning model or as a post-processing step. These approximations ignore two important ideas: (1) short trailers, commonly known as pups, can be moved behind tractors in trains of two or three trailers, and the cost of moving a trailer train is not linear in the number of trailers; and (2) drivers must be scheduled for each dispatch, and driver rules may introduce travel in addition to that minimally required for trailer balance. In this paper, we present technology designed to more accurately estimate the operational execution costs of a load plan. A computational study demonstrates that our technology produces accurate operational execution costs estimates, typically within 2% of actual incurred costs.

1 Introduction

The trucking industry provides an essential service to the U.S. economy by transporting goods from business to business and from business to consumer. Less-than-truckload (LTL) transportation represents an important, but relatively small, segment of the trucking industry serving businesses that ship quantities ranging from 150 lbs to 10,000 lbs, i.e., less-than-truckload quantities. A typical shipment occupies only 5-10% of the trailer capacity. LTL carriers therefore collect and
consolidate freight from multiple shippers, and route shipments through a terminal network of cross-
dock transfer terminals. Consolidation of shipments leads to higher trailer capacity utilization, but
also to increased travel distance (referred to in the industry as circuity) and freight handling costs.

How freight is routed through the terminal network, and thus where opportunities for consol-
didation occur, is determined by a so-called load plan, which specifies a fixed sequence of terminal
transfer points for each shipment entering the system. Consequently, load plan design is critical to
effective operations of an LTL carrier. Freight fluctuations, whether seasonal or caused by chang-
ing economic conditions, force LTL carriers to regularly review and adjust their load plans; small
percentage gains in trailer utilization can lead to significant cost savings.

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

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

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

We have conducted a computational study of the proposed approach using an actual load plan
and actual shipment data from a super-regional LTL carrier operating in the continental U.S. We compare the execution cost estimates of the load plan from two different approaches with the actual linehaul costs incurred in practice when executing the load plan. The first estimate comes from SuperSpin (Manhattan Associates (2010)), the current industry standard software for load plan design. The second estimate is taken from the technology presented in this paper. The results show that SuperSpin tends to underestimate actual costs, between 88.8% to 90.6% of actual, while our technology provides accurate cost estimates, between 99.6% and 101.7% of actual.

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

- improves load plan execution cost estimates; accuracy improvements on the order of 10-15% are shown for a super-regional carrier,
- builds a set of dispatches and generates a set of cost-effective driver duties and tours covering these dispatches; driver duties and tours satisfy government regulations and union and/or company rules and can thus be used in practice, and
- solves real-life instances efficiently; less than 2 hours for instances representing a week of data with over 140,000 shipments, which equates to over 10,000 loads and approximately 6,000 driver duties.

The remainder of the paper is organized as follows. In Section 2, we describe LTL operations, discuss load plan design, and present the need for more accurate load plan cost estimation. In Section 3, we review relevant literature. In Section 4, we formally state the load plan costing optimization problem and discuss the modeling issues and choices. In Section 5, we introduce our solution approach. In Section 6 we present the results of an extensive computational study using historical data from a super-regional LTL carrier in the U.S. Finally, in Section 7, we present some concluding remarks.

2 LTL System Description

LTL carriers operate fixed terminal networks to provide service. During the day 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 an end-of-line terminal. End-of-line terminals serve as sorting centers and loading facilities for outbound and inbound freight for the geographic area. There usually is not enough freight collected at an end-of-line terminal to build full or nearly full trailerloads to other end-of-lines, hence another layer of consolidation is introduced.

Outbound freight from an end-of-line is sent to a breakbulk terminal where it may be consolidated with freight from other end-of-lines. By consolidating freight from and destined for many locations,
breakbulk terminals handle enough freight to build and dispatch cost-effective trailerloads. Consolidation at a breakbulk requires freight to be cross-docked which requires time and results in handling cost.

A typical shipment may travel from an origin end-of-line to an origin breakbulk, then to a destination breakbulk, and eventually to a destination end-of-line. Once it arrives at the destination end-of-line, it is loaded on to a delivery truck and sent to the consignee.

Figure 1: A typical LTL network

We focus on the freight movements between terminals, i.e., the linehaul operations of an LTL carrier. A load plan determines how freight is routed through the linehaul network by specifying a sequence of transfer points for each shipment entering the system. Each terminal-to-terminal trailer movement without intermediate handling is referred to as a direct move. A load plan, therefore, determines a unique path of direct moves for all shipments entering the linehaul system at a specific origin end-of-line terminal and destined for a specific destination end-of-line terminal. To keep cross-dock operations simple, traditional load plans have a tree-like structure; at any terminal, freight (either originating or transferring) with the same final destination will always be loaded to the same next terminal.

LTL carriers pack shipments into 28-foot trailers known as pups, or 53-foot vans. Typically, one tractor pulls either a single van or two pups in a so-called dispatch. A trailer direct move may be executed by a single dispatch, or may require multiple dispatches along the legs of the trailer path associated with the direct move. In case a trailer path consists of multiple legs, the freight is relayed at the intermediate terminals. Relaying is frequently necessary because of the limitations imposed upon drivers by regulations and work rules. In the U.S., regulations dictate that a driver is not allowed to drive for more than 11 hours or be on duty for more than 14 hours before requiring a rest period of at least 10 hours. Therefore, when the travel time between the origin and destination of a direct move is more than a single driver can cover without a rest period, one or more relays
are introduced to ensure timely transfer of the freight. Usually, trailer relay operations occur at breakbulk terminals, although they may also occur at special relay facilities. At a relay point, the load is transferred to another driver and continues with minimal delay. For example, a direct move for one LTL carrier from Dallas to San Francisco requires 27 hours of drive time and involves two relays and three drivers. It happens frequently that different direct moves may include common legs in their trailer paths. For example, both the Dallas-to-San Francisco and the Dallas-to-El Paso direct moves include the Dallas-El Paso leg in their respective trailer paths.

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

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

3 Related Literature

Load plan design leads to challenging optimization problems. Early research in this area focused on relatively simple local search heuristics for models based on static networks that do not explicitly capture service standard constraints or the timing of consolidation opportunities (see Powell and Sheffi (1983), Powell (1986), Powell and Sheffi (1989), and Powell and Koskosidis (1992)). In Powell and Farvolden (1994), a dynamic model is presented that more accurately models consolidation timing, and an alternative heuristic is developed that relies on determining service network arc subgradients by solving large-scale multi-commodity network flow problems. The approach, however, allows freight for a specific origin-destination pair to be moved over multiple paths, and does not model empty equipment balancing decisions. Jarrah et al. (2009) develop a column generation approach to create load plans where columns represent freight path trees into a destination. A slope scaling heuristic is used to linearize costs when generating columns. The approach explicitly models consolidation timing, but only uses a single time point per day per terminal, so it cannot capture many tightly constrained freight paths useful for 1- and 2-day service offerings.
Erera et al. (2009) develop optimization technology to create improved load plans, including day-differentiated plans, for a major super-regional U.S. LTL carrier. The approach uses a detailed time-space network representation to accurately model consolidation in time. However, it relies on several important simplifying assumptions for tractability:

1. Each day in the planning horizon is discretized into 8 decision epochs, varying in length from 2 hours to 5 hours;
2. Loaded transportation costs are assumed to be linear in the number of trailers required for each direct move; and
3. Empty transportation costs are assumed to be linear in the number of empty trailers dispatched between terminals, and empty dispatches are determined only to correct trailer imbalances created by loaded trailer movements.

These assumptions, while reasonable, still provide only a rough approximation of true transportation costs. In the research described in this paper, we model LTL operations at a finer level of detail.

Problems also related to the one we consider in this paper are the focus of Cohn et al. (2007) and Root and Cohn (2008), in which solution approaches are developed that integrate empty balancing with a pup matching and routing for small package express carrier operations. The proposed set partitioning model uses composite variables that define complete paths for one or more trailers, and employs templates to limit the set of such composites generated. While we also consider a pup loading and matching problem, our matching problem is simple since we assume that the best trailer path is known for each direct. Furthermore, we estimate transportation costs more precisely since we construct feasible driver tours to cover loads. Erera et al. (2008) employ greedy approaches to construct driver tours that cover dispatches; in this paper, we develop an optimization-based set covering heuristic.

4 Model Formulation

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

1. The problem is formulated on a time-space network. Flat network representations, i.e., networks without an explicit time dimension, such as the ones used in Powell and Sheffi (1983), Powell (1986), Powell and Sheffi (1989), and Powell and Koskosidis (1992), are based on two important assumptions: (1) the total trailer loads needed on a direct during the planning horizon can be determined by assuming that all freight traveling at any time within the planning
horizon can be consolidated; and (2) service standard constraints can be modeled by using a proxy, e.g., by ensuring a minimum trailer frequency on a direct per day. In today’s LTL market where 1-day and 2-day service have become the norm, these assumptions are no longer valid. It is necessary to use a representation that can explicitly represent time. A detailed time-space network model allows consolidation timing and service standards to be modeled accurately. Given a time discretization of the planning horizon, multiple nodes are created for each terminal, one for each time point, so that each node represents a location and a point in time. For each leg in the linehaul network, we create multiple transportation arcs in the time-space network, each representing the possibility to move freight at a particular time. Each node in the time-space network is connected with an arc to the node representing the same terminal at the next time point, thus modeling the possibility to hold freight at a terminal. See Figure 2 for an illustration.

![Figure 2: Time-Space Network](image)

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

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

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

3. *Time is discretized in hours.* Time must be modeled at a fine level of granularity for two reasons: (1) to be able to accurately model the driver rules discussed in Section 2, and (2) to be able to properly model freight paths between origin-destination pairs with tight service standards. Consider the freight path encountered at a super-regional carrier for freight available in Lexington, KY at 7 pm and due in Grayling, MI at 8 am the next day shown in the top part of Table 1.

<table>
<thead>
<tr>
<th>Time Event</th>
<th>Location 1</th>
<th>Location 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leave Lexington, KY</td>
<td>19:00</td>
<td>21:00</td>
</tr>
<tr>
<td>Finish handling at Cincinnati, OH</td>
<td>21:30</td>
<td></td>
</tr>
<tr>
<td>Leave Cincinnati, OH</td>
<td>22:00</td>
<td>01:45</td>
</tr>
<tr>
<td>Finish handling at Toledo, OH</td>
<td>02:15</td>
<td></td>
</tr>
<tr>
<td>Leave Toledo, OH</td>
<td>03:00</td>
<td>07:27</td>
</tr>
</tbody>
</table>

Table 1: Modeling Freight Paths

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

4. *Freight enters the linehaul system at 7 p.m. and leaves the linehaul system at 8 a.m.*. All freight picked up during the day is assumed to be ready to be send into the linehaul system at 7 p.m. local time. All freight to be delivered during the day must arrive at its destination terminal at 8 a.m. local time. Thus, we model the freight that enters the linehaul network at terminal $t_1$ on day $d_1$ and is due at terminal $t_2$ on day $d_2$ as originating in the time-space network at node $n_1 = (t_1, 7 \text{ p.m.} \ d_1)$ and is destined to node $n_2 = (t_2, 8 \text{ a.m.} \ d_2)$. 
5. **Handling 1-day service freight takes 30 minutes and handling all other freight takes 2 hours.**

A certain amount of time is required for handling freight at intermediate breakbulks. Special handling procedures are generally used at breakbulks to prioritize the processing of 1-day freight to ensure that it can meet its service expectation. Therefore, a short handling time for 1-day freight is appropriate. We note that handling of freight can only occur during business hours, which typically start at 12 am on Monday morning, and end Saturday at noon. The terminals are, however, accessible to drivers arriving/departing all weekend long.

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

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

### 5 Solution Approach

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

#### 5.1 Phase I: Loading and Matching Trailers

In the first phase, we determine a timed path for each commodity to generate loaded trailers and dispatches. As an approximation, we assume that the freight for each commodity can be split arbitrarily across multiple trailers; this assumption is not overly optimistic since the total flow for a commodity is comprised of many shipments, and since in practice this flow is rarely split across more than two trailers.

We develop a GRASP heuristic to determine the timed paths. The GRASP heuristic chooses a path for each commodity sequentially using a shortest path algorithm; note that since the load plan is fixed, determining the path for a commodity is simply selecting the dispatch times for each
of the direct moves in the load plan path (represented by arcs in the time-space network). The sequential nature of the search enables us to estimate the marginal cost of adding the commodity under consideration to all possible dispatch arcs, and thus to minimize the marginal cost increase that results from selecting a set of feasible dispatch times.

Dispatch costs for purchased transportation direct moves are linear in the number of trailers loaded. For regular moves, dispatch costs are linear in the number of two-pup trains loaded along each leg of the trailer path. We now illustrate how to compute the marginal cost of adding a commodity of size $c$ (measured in fractional trailerloads) to arc $a$ representing a leg of the trailer path of non-purchased direct $d_0$; a similar computation is used for computing the marginal cost on a purchased transportation direct. Suppose that $d_0, \ldots, d_m$ are the directs whose trailer paths include arc $a$ as a leg and that the dispatch cost (per trailer pair) on arc $a$ is $p_a$. Furthermore, let $w_0$ be the existing freight (measured in fractional trailerloads) on arc $a$ for direct $d_0$ and let $e_i$ be the current number of trailers on leg $a$ for direct $d_i$, $i = 0, \ldots, m$. Finally, let $U$ be the capacity of a trailer. Adding commodity $c$ to arc $a$ changes the required number of trailers on leg $a$ for direct $d_0$ from $e_0$ to $\lceil \frac{w_0 + c}{U} \rceil$. Hence the marginal dispatch cost is

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

(1)

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

Note that it is simple to determine the actual dispatches moving on each arc representing a leg in time, given the method of selecting the timed paths. Since we assume that each commodity is splittable across trailers, we simply create trailers (and two-pup trailer trains) whenever a trailer (or train) becomes full in the process above. These trailer pairs (and single trailers, when necessary)
Algorithm 1 GRASP for Trailer Loading and Matching

Sort the commodities in order of increasing slack time. In case of ties, sort the commodities in order of decreasing weight.

for $i = 1$ to $N$ do

Create a copy of the commodity list

while the list is not empty do

Select a commodity from the list biased towards the top, i.e., the $k$-th commodity $c_k$ with probability $\lambda \cdot (1 - \lambda)^{k-1}$, $k = 1, 2, \ldots$

Find the least-marginal-cost path for $c_k$ that conforms to the load plan, using the cost in (1)

Remove $c_k$ from the commodity list

end while

if an improved solution is found then

Update the best solution

end if

end for

will be referred to as dispatches, and the commodities contained within each dispatch will be used in Phase II to determine its time flexibility (if any).

5.2 Phase II: Determining Dispatch Windows

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

Let $P$ be the set of dispatches built in Phase I corresponding to purchased transportation. Since purchased transportation takes place on fixed schedules (recall that purchased transportation is typically provided by railroad companies), these dispatches are not altered. Let $L$ be the set of dispatches, or loads, built in Phase I corresponding to transportation provided by company drivers. Our goal is to find for each load $i \in L$ an earliest and a latest dispatch time, denoted by $\alpha_i$ and $\beta_i$ respectively, such that if all loads are dispatched between their earliest and latest dispatch times, all freight can be moved feasibly from origin to destination, i.e., every shipment reaches its destination at or before its due time and can make feasible connections at transfer and relay points. See Figure 3 for an illustration of a dispatch window.

We introduce some additional notation before discussing a linear programming model for this
Figure 3: Dispatch Window

problem. Let \( o_c \) and \( d_c \) denote the ready time at the origin and the deadline at the destination for commodity \( c \). Furthermore, let \( p_{c1}, \ldots, p_{cn_c} \) denote the sequence of dispatches for commodity \( c \). Finally, let \( dt_{pc}, tt_{pc}, ht_{pc}, \) and \( ft_{pc} \), be the dispatch time, travel time, handling time, and finish time, respectively, of dispatch \( p_{cd} \), where the dispatch time and the finish time are given by the timed path selected for commodity \( c \) in Phase I. Our goal is to determine load dispatch windows that provide the most flexibility, and therefore the objective function of the linear program must be chosen consistent with this goal. We have chosen to maximize the sum of the widths of the individual dispatch windows for each load. An alternative is to maximize the minimum width of any dispatch window. However, since there typically are a few dispatches without any flexibility, this objective does not produce useful information in this case.

The linear program to determine dispatch windows is as follows

Maximize \[ \sum_{i \in L} (\beta_i - \alpha_i) \]

subject to

1. \[ \alpha_{p_{ci}} \geq o_c \quad \forall c, p_{ci} \in L \] (2)
2. \[ \beta_{p_{cn_c}} + tt_{p_{cn_c}} \leq d_c \quad \forall c, p_{cn_c} \in L \] (3)
3. \[ \beta_{p_{cil}} + tt_{p_{cil}} + ht_{p_{cil}} \leq \alpha_{p_{cil+1}} \]
   \[ \forall c, 1 \leq l \leq n_c - 1, p_{cil} \in L, p_{cil+1} \in L \] (4)
4. \[ \beta_{p_{cil}} + tt_{p_{cil}} + ht_{p_{cil}} \leq dt_{p_{cil+1}} \]
   \[ \forall c, 1 \leq l \leq n_c - 1, p_{cil} \in L, p_{cil+1} \in P \] (5)
5. \[ ft_{p_{cil}} \leq \alpha_{p_{cil+1}} \]
   \[ \forall c, 1 \leq l \leq n_c - 1, p_{cil} \in P, p_{cil+1} \in L \] (6)
6. \[ \alpha_{p_{cil}} \leq dt_{p_{cil}} \leq \beta_{p_{cil}} \quad \forall c, 1 \leq l \leq n_c, p_{cil} \in L \] (7)

Constraints (2) ensure that the first dispatch occurs no earlier than the latest ready time at the
origin of all of the constituent commodities, and constraints (3) ensure that last dispatch is such that all freight arrives at the destination before its deadline. Constraints (4), (5), and (6) ensure feasible connections at transfer and relay points; note that all commodities carried within the dispatches must make feasible connections. Finally, constraints (7) forces the dispatch times on the timed path selected in Phase I to be feasible.

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

**Algorithm 2 Post-Processing Dispatch Windows**

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

5.3 Phase III: Constructing Driver Tours

In the third phase, we determine driver tours to cover all dispatches in the set \( L \); note that the dispatches in the purchased transportation set \( P \) are easy to cost, and thus are ignored in this phase. Each driver tour can be performed by a single driver. It begins and ends at a driver domicile, and thus forms a timed cycle, and consists of one or more duties. A duty is a feasible sequence of timed dispatches that can be performed in a single 24-hour period and conforms to hours-of-service regulations. If a tour contains multiple duties, the duties are separated by a required rest period of
appropriate duration. We use the current (2009) U.S. hours-of-service regulations that impose the following restrictions on drivers: a driver is allowed to drive up to 11 hours in a duty, a duty must not exceed 14 hours, and a driver must rest for at least 10 hours between duties.

Note that duties may include empty dispatches. If we assume that drivers are always dispatched with two trailers (either loaded or empty), empty trailer balance over time is implied at all terminals. For this reason, neither Phase I or Phase II above includes any consideration of empty trailer balancing decisions.

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

Since we want to build company driver tours in this phase, we modify the time-space network by removing all purchased transportation arcs, and by adding lay-down arcs from every node to the nodes representing the same terminal 10 to 14 hours into the future. This network is then used to generate columns for a set-covering model. For each \( i \in L \), let \( A(i) \) be the subset of arcs in the time-space network associated with load \( i \) that fall within its dispatch window. Since \( A(i) \) and \( A(i') \) with \( i \neq i' \) may contain common arcs, an arc does not uniquely identify a load. For each arc \( a \), let \( L(a) = \{i \in L \mid a \in A(i)\} \) be the set of loads that can potentially use \( a \).

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

\[
\begin{align*}
\text{Minimize} & \quad \sum_{t \in T} c_t x_t \\
\text{subject to} & \quad \sum_{t \in T} a_{it} x_t \geq z_i \quad \forall i \in L \\
& \quad x_t \in \mathbb{Z}^+ \quad \forall t \in T
\end{align*}
\]

Since the set of tours is too large in practice to consider explicitly, we rely on column generation to solve the linear programming relaxation. Given a dual solution \( \pi \) to the linear programming relaxation of a restricted master problem, the pricing problem seeks to identify a tour with negative reduced costs. More specifically, the pricing problem seeks a tour minimizing \( \sum_{a \in L} (c_a - \max_{i \in L(a)} \pi_i) \).
Note that because multiple loads may be covered with the same arc, we must use \( \max_{i \in L(a)} \pi_i \) to determine the dual value to use for an arc. The pricing problem is a resource-constrained shortest path problem with arc cost \( c_a - \max_{i \in L(a)} \pi_i \).

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

<table>
<thead>
<tr>
<th>Resource limits at a node</th>
<th>Initial value</th>
<th>Resource extension functions</th>
<th>Resource limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duty time</td>
<td>0</td>
<td>+ ( [d_a] ) + 1</td>
<td>reset to 0</td>
</tr>
<tr>
<td>Driving time in duty</td>
<td>0</td>
<td>+ ( d_a ) unchanged</td>
<td>reset to 0</td>
</tr>
<tr>
<td>Num of dispatches in duty</td>
<td>0</td>
<td>+ 1</td>
<td>reset to 0</td>
</tr>
<tr>
<td>Num of lay-downs in duty</td>
<td>0</td>
<td>unchanged</td>
<td>+ 1</td>
</tr>
<tr>
<td>Cost</td>
<td>0</td>
<td>+ ( (c_a - \max_{i \in A(a)} \pi_i) ) unchanged</td>
<td>+ ( H )</td>
</tr>
</tbody>
</table>

Table 2: Resource Extension Functions and Resource Limits

We round up the driving time when updating the duty time label because dispatches only occur at points in the time discretization, i.e., whole hours. Note that the duty time label is only used in the dynamic programming algorithm. Actual duty times may be slightly smaller, and these are also computed and used when reporting duty times in our computational study.

We solve the resource-constrained shortest path problem using a standard dynamic programming approach (see Irnich and Desaulnier (2005) and Desrosiers et al. (1995) for discussions of dynamic programming approaches for constrained shortest path problems). One slight modification is that in the path extension step, a waiting arc and a lay-down arc are not allowed to immediately follow each other in order to prevent undesirable long rests.

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

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

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

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

The algorithm to find a low-cost set of driver tours covering all dispatches is provided in detail in Algorithm 3.

Algorithm 3 Load Covering

Generate a set of initial columns

repeat

Solve the linear programming relaxation of the restricted master problem

Retrieve the dual prices

Sort the loads in increasing order of \((c_i - \pi_i)\)

for \(i = 1\) to \(|L|\) do

if load \(i\) is not excluded from consideration then

Invoke a dynamic programming search for a tour that starts with load \(i\)

while a tour with a negative cost is not found and there are feasible extensions do

Perform a dominance check and a path extension

end while

if a desirable tour is found then

Add a column representing the tour

break loop

else

Exclude load \(i\) from consideration for \(M\) iterations

end if

end if

end for

until a column with negative reduced cost is not found

Solve a set covering problem over the columns in the restricted master problem

5.3.1 Meet-and-Turns and Initial Columns

It is well-known that a good set of initial columns can reduce the running time of a column generation algorithm. However, before presenting our approach for creating initial columns, we must describe meet-and-turns, which are used by LTL carriers on long legs to reduce lay-down costs. A meet-and-turn can occur when two drivers move loads in opposite directions on a leg that is longer than half of the maximum allowed driving time in a duty, i.e., 5.5 hours. Without intervention, the drivers moving these loads will be unable to return to their domiciles at the end of the day because they would violate the driving time limit. A meet-and-turn, illustrated in Figure 4, instead has the
two drivers meet at a location along the leg, exchange their loads, and then return to their respective starting location. This ensures that both loads arrive at their destination on time, and both drivers get back to their domiciles at the end of the day. A parking lot or a rest area usually suffices as a meet-and-turn location. Executing meet-and-turns reduces lay-down expenses for the carriers and improves the quality of life for the drivers.

![Figure 4: Meet-and-Turn](image)

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

### 5.3.2 Short Driving and Duty Times

Hours-of-service regulations, which are motivated by safety considerations, only restrict maximum driving and duty times. Therefore, short driver duties with short driving times are legal, but may not be cost-effective. When non-unionized carriers plan short driver duties, they often attempt to do so by allowing for “dual-use” of the drivers by employing them also for loading and unloading trailers (so-called dock work), and by reducing the staff of dedicated dock workers accordingly. In general, short driver duties are therefore acceptable, but typically less desirable than longer driver duties.

In this section, we propose a penalty-based approach that enables the analysis of the tradeoff between the quality of the tours (in terms of duty time and driving time) and the execution costs. The approach penalizes short duties with a term in the objective function that is proportional to the difference between the actual and the maximum allowed driving time in a duty, i.e., 11 hours. The reason that we penalize short driving times rather than short duty times is because waiting between dispatches is counted towards duty time; penalizing short duty times would create incentives for adding waiting time to duties, and this should not be encouraged.

Let $n_t$ be the number of duties in tour $t$ (either 1 or 2), let $\alpha$ be a parameter indicating the weight assigned to the penalty term, and let $LD$ be the set of lay-down arcs. Then the cost of a
Let $SL$ contain loads that require less than or equal to 5.5 hours of driving and let $LL$ contain loads that require more than 5.5 hours of driving.

**Algorithm 4 Creation of Initial Columns**

for all $i, j \in SL$, $i \neq j$ do
  if $i$ and $j$ can form a feasible out-and-back tour without a lay-down then
    Create a column representing the tour
  end if
end for

for all $i, j \in LL$, $i \neq j$ do
  if $i$ and $j$ can form a feasible meet-and-turn then
    Create a column representing both tours in the meet-and-turn
  end if
end for

for all $i, j \in LL$, $i \neq j$ do
  if $i$ and $j$ can form a feasible out-and-back tour with a lay-down then
    Create a column representing the tour
  end if
end for

for all $i \in I = SL$ such that $i$ has not been covered by any tour do
  Create a column representing an out-and-back tour with an outbound dispatch moving $i$ and empty inbound dispatch
end for

for all $i \in I = LL$ such that $i$ has not been covered by any tour do
  Create a column representing a meet-and-turn consisting of $i$ and an empty dispatch in the opposite direction
end for
tour $t$ becomes

$$\sum_{a \in t} \left( c_a - \max_{i \in A(a)} \pi_i \right) + \alpha \left( 11 \cdot n_t - \sum_{a \in t} d_a \right)$$

$$= \alpha \cdot 11 \cdot \left( 1 + \sum_{a \in t \cap LD} 1 \right) + \sum_{a \in t} \left( c_a - \max_{i \in A(a)} \pi_i - \alpha \cdot d_a \right)$$

$$= \alpha \cdot 11 + \sum_{a \in t} \left( c_a - \max_{i \in A(a)} \pi_i - \alpha \cdot d_a + 1_{LD}(a) \cdot \alpha \cdot 11 \right)$$

which is still additive on arcs.

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

<table>
<thead>
<tr>
<th>Initial value</th>
<th>Transportation arc $a$</th>
<th>Waiting arc</th>
<th>Lay-down arc</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Cost</td>
<td>$\alpha \cdot 11$</td>
<td>$+ (c_a - \max_{i \in A(a)} \pi_i) - \alpha \cdot d_a$</td>
<td>unchanged</td>
</tr>
</tbody>
</table>

Table 3: Cost Extension with Penalty

## 6 Computational Study

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

<table>
<thead>
<tr>
<th>Instance</th>
<th>Start date</th>
<th>End date</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>March 01, 2009</td>
<td>March 07, 2009</td>
</tr>
<tr>
<td>W2</td>
<td>March 08, 2009</td>
<td>March 14, 2009</td>
</tr>
<tr>
<td>W3</td>
<td>March 15, 2009</td>
<td>March 21, 2009</td>
</tr>
<tr>
<td>W4</td>
<td>March 22, 2009</td>
<td>March 28, 2009</td>
</tr>
</tbody>
</table>

Table 4: Weeks Used in Our Computational Study

All computational experiments were executed on a Linux system with a 2.66 GHz Intel Xeon processor and 4 GB of RAM, and use CPLEX 11.1 as the optimization engine.
6.1 GRASP Heuristic Parameters

The first experiment is designed to determine the parameters $\lambda$ and $N$ for the GRASP heuristic for loading and matching trailers. For $\lambda = 0.1, 0.2, ..., 0.9, 1.0$, we let Algorithm 1 run for 100 iterations and monitor the progress of the value of the best solution found. Note that when $\lambda = 1$, the algorithm reduces to a greedy heuristic and the behavior of the algorithm is deterministic, so there is no benefit of performing more than one iteration. Each iteration takes approximately 4 minutes to run. Figure 5 and 6 show the progress over time for weeks W1 and W4 (similar behavior was observed for weeks W2 and W3).

![Figure 5: Progress of GRASP Heuristic for Instance W1](image)

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

6.2 Dispatches and Dispatch Windows

Next, we present a few statistics related to the dispatches built in Phase I and the dispatch windows determined in Phase II. (Note that the dispatch windows are determined using a relatively small linear program, and thus are computed in a matter of seconds.)

In Figure 7, we show the number of dispatches occurring at particular times during the day as determined by Phase I. We see that most dispatches occur between 7 p.m. and 6 a.m. This is not unexpected, and in line with what happens in practice, as a significant portion of shipments have
1-day service guarantees, which implies that they have to be moved between 7 p.m. and 8 a.m. If possible, freight with longer service standards is also moved within this time frame so that it may be consolidated with the 1-day freight.

In Figure 8, we show the distribution of the widths of dispatch windows as determined by Phase II. As is evident from the figure, a few dispatches have little or no flexibility and must be dispatched according to a specific schedule to make service; most likely these represent shipments in relatively long corridors with a 1-day service guarantee. At the other end of the spectrum are a few dispatches that have quite a bit of flexibility; most likely these represent shipments on origin-destination pairs that are relatively close, but have a 5-day service guarantee. From an operational perspective, the most relevant information is that most dispatches have some flexibility, and this flexibility can be exploited to build low-cost driver tours.

It is also insightful examine the dispatch windows on a single linehaul leg in more detail. Figure 9 shows all the Markham-Chicago dispatches and their dispatch windows; note that Markham is another location in Illinois, not far from Chicago. A few interesting observations can be made. First, the dispatches occurring at 7 p.m. and 8 p.m., which likely include a substantial portion of the shipments picked up during the day, have little or no flexibility. These dispatches are the used to move shipments with a 1-day service standard. Furthermore, we note that the dispatches between 10 a.m. and 6 p.m. have the most flexibility. These likely contain shipments bound for Chicago or for a further terminal with service levels that can relatively easily be achieved.
6.3 Column Generation and IP Optimization

Next, we consider parameter tuning for the column generation and IP optimization processes at the heart of Phase III. Recall that during the column generation process, if an attempt to build a tour with a negative reduced cost starting with a particular load fails, we exclude that load from consideration for the next $M$ iterations to hopefully avoid wasting computing time on an unsuccessful search. The tradeoff between the computing time and the value of the final LP solution when we vary $M$ is shown in Figure 10; for $M = 50, 100, 1,000, 10,000, \infty$. We see that re-visiting loads provides a small benefit, but it comes at a very high price in terms computing time. Hence, for the remaining computational experiments, we use $M = \infty$, i.e., we will not re-visit a load again once our attempt to build a tour with a negative reduced cost starting from that load fails.

Next, we provide more details about the initial columns generated using structured templates; the templates are summarized in Table 5. Figure 11 shows the composition of the columns in the initial LP solution and the final LP solution in terms of their structure, i.e., the template corresponding to their structure. Of course, in the final LP solution we encounter structures that were not present

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

Table 5: Template Types
in the initial LP solution. These structures are lumped together under the “template” COLGEN. For example, columns representing tours with duties involving more than two dispatches will end up under this template. This includes, for example, triangular duties, i.e., duties of the form A-B-C-A (dispatches AB, BC, and CA), which can be quite effective. Column generation is used precisely to generate such duties if desirable. The figure demonstrates that using these more complicated structures substantially reduces the use of inefficient structures with empty dispatches.

Finally, and most importantly, in Table 6 we report the value of the final LP solution and the value of the IP solution generated using the columns in the final LP solution (where the stopping criterion for the IP solve was an optimality gap of less than 0.1%). Since IP solution values are very close to LP solution values, it appears that it is reasonable not to perform additional column generation within the branch-and-bound process.

<table>
<thead>
<tr>
<th></th>
<th>Value LP solution</th>
<th>Value IP solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>3,247,583</td>
<td>3,249,398</td>
</tr>
<tr>
<td>W2</td>
<td>3,267,249</td>
<td>3,269,375</td>
</tr>
<tr>
<td>W3</td>
<td>3,311,176</td>
<td>3,314,513</td>
</tr>
<tr>
<td>W4</td>
<td>3,350,560</td>
<td>3,353,409</td>
</tr>
</tbody>
</table>

Table 6: Comparison of LP and IP Solutions
6.4 Tour Structures

In this section, we provide more details about the structure of the driver tours generated. In Table 7, we report the number of tours with 1 duty and 2 duties, the number of duties with 1, 2, 3, and 4 dispatches, and the number of duties involving meet-and-turns, and the number of loaded versus empty dispatches. We see that a relatively small percentage of duties involve more than 2 legs. Since counts only provide a partial picture, Figure 12 and 13 present the distribution of the driving and duty time of the duties. We see that the majority of duties have a driving time of more than 7 hours and a duty time of more than 9 hours, which is desirable. However, a non-trivial fraction corresponds to short duties with short driving times.

6.5 Execution Cost Estimates

The primary goal of this research was to develop technology to accurately estimate the operational execution cost of a load plan. To demonstrate that we have achieved our goal, we compare for the four instances, in Figure 14, the actual execution costs incurred by the carrier, the execution cost estimate produced by SuperSpin (the current industry-standard for load plan design), and our execution cost estimate. The actual execution costs are normalized at 100% and the two estimates are given as a percentage of the actual costs. The figure shows that our technology produces remarkably accurate execution cost estimates, within 1.7% of the actual execution costs incurred for each of the four weeks. The figure also shows that SuperSpin tends to under-predict execution costs (about 90% of the actual execution costs incurred), primarily due to an over-estimation of the
consolidation opportunities that likely results from the static network representation. Furthermore, we present in Figure 15 for our cost estimate the breakdown into loaded transportation costs, empty repositioning costs, and lay-down costs.

6.6 Varying Maximum Allowed Number of Dispatches in a Duty

During the construction of tours, we limit the number of dispatches in a duty. This restriction is included for two primary reasons. First, duties with a small number of legs are preferred by both drivers and the carrier. Second, limiting the number of dispatches per duty limits the number of feasible duties and thus simplifies the pricing problem, which reduces the computing time. In the next experiment, we investigate the impact of varying the maximum number of dispatches allowed in a duty. Figure 16 shows the total linehaul cost and the number of column generation iterations versus the maximum allowed number of dispatches in a duty. When a duty is allowed to contain more dispatches, the technology is able to generate more complicated and efficient driver tours, and thus to reduce the total linehaul cost. However, we see that the benefits of allowing more than 4 dispatches in a duty is negligible.

6.7 Short Driving and Duty Times

To this point in our computational study, short driving and duty times were not discouraged. As we observed in Figures 12 and 13, a majority of the duties have a driving and duty time close to their respective limits, but there are a fair number of duties with small driving and duty times.
In Figure 17, we analyze the tradeoff between the quality of the tours, measured in terms of their driving and duty time, and the operational execution costs. We see that an increase in the average driving time of 1.5 hours and an increase in the average duty of 1 hour comes at an increase in operational execution costs of approximately 1%.

7 Future Research

Load plan design technologies must use simplifying assumptions to ensure computational tractability, and thus may substantially under- or over-estimate actual operational execution costs. We designed and implemented technology that accurately estimates the operational execution costs of a given load plan.

One important next challenge is to integrate load planning and execution cost estimation technologies. Building a load planning methodology that explicitly recognizes that driver tours will be used to cover planned dispatches may have substantial promise in further improving LTL load plan design.

References


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

Table 7: Tour Structure


27
Figure 12: Driving Time Histogram


Figure 13: Duty Time Histogram

Figure 14: Estimated Linehaul Cost as a Percentage of Actual Execution Cost
Figure 15: Cost Breakdowns

Figure 16: Impact of Varying the Maximum Number of Allowed Dispatches in a Duty
Figure 17: Impacts of Varying Penalty Factor $\alpha$