Locating Drivers in a Trucking Terminal Network

Alan Erera, Michael Hewitt, Burak Karacık, Martin Savelsbergh

The Supply Chain and Logistics Institute
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Abstract

We consider the problem of determining the home locations, or domiciles, of truck drivers of a less-than-truckload carrier. Domiciling decisions are complex, not in the least because of U.S. DOT regulations and union rules restricting driver schedules, but have a noticeable impact on the operating costs of less-than-truckload carriers. We present an iterative scheme, using driver dispatch technology in each iteration, to allocate drivers to terminals and to determine drivers’ bids so as to satisfy union requirements. Computational experiments demonstrate the value of the iterative scheme and quantify the impact of union rules on the number of drivers required (and thus on carrier operating costs).

*corresponding author; email: alerera@isye.gatech.edu, fax: +1 404 894 2301
1 Introduction

The trucking industry is vitally important to the economy of the United States. Trucking provides an essential service by transporting goods from business to business and from business to consumer. The less-than-truckload (LTL) segment serves businesses that ship quantities ranging from 150 lbs to 10,000 lbs, i.e., less-than-truckload quantities. Large LTL carriers employ thousands of drivers to move loaded and empty trailers between dozens of terminals in their linehaul network. Drivers are expensive resources, and their cost-effective management is complex. This complexity arises from two primary sources: first, drivers have domiciles, or home terminals, to which they return with some frequency, and second, driver dispatching is constrained by regulation and labor union rules that limit how much driving is allowed between rest (sleep) periods and which terminals are feasible to visit.

Thus, major tactical questions faced by LTL carriers include determining at which terminals to domicile drivers and how many drivers to locate at each of these terminals. Most large LTL carriers are unionized, and these unions have negotiated a variety of rules governing driver use, which complicate decisions regarding driver domiciles. Union rules typically require, for example, that a certain percentage of a driver fleet be designated as bid drivers, whose dispatches are restricted. Each bid driver at a domicile has a bid terminal, which defines a region in the linehaul network to which the driver can be dispatched. Union rules also guarantee that bid drivers receive favorable treatment in the dispatch process. For example, bid drivers have priority over other drivers for dispatches destined to a terminal in their region.

In this paper, we focus on three tactical questions related to allocation of drivers within trucking terminal networks:

- How many drivers should be domiciled at each terminal?
- What fraction of these drivers should be bid drivers?
- What should be the bid terminal for each bid driver?

Driver fleet sizing and allocation has a significant impact on LTL system performance. If too many drivers are allocated to a domicile or if the distribution of drivers to various bid terminals is not appropriate, the result may be poor driver utilization and increased cost.
Alternatively, if too few drivers are allocated to a domicile or to specific bid terminals, then the carrier may not be able to serve loads on time resulting in poor customer service.

The primary difficulty of driver allocation is the fact that an individual driver may cover dispatches during the course of a schedule that are neither outbound from nor inbound to his domicile terminal. Some drivers therefore “help” drivers domiciled at other terminals. Driver allocation rules which do not take into account this interaction thus may be overly simplistic and lead to ineffective decisions. On the other hand, predicting the level or location of such interaction may be difficult without a detailed simulation of dispatch decisions.

The methodology we propose to address these questions utilizes a driver scheduling tool as a dispatch simulator. Given a set of drivers with known domiciles and bid information, the scheduling tool determines effective driver schedules and load dispatches that satisfy all rules and restrictions, generating empty movements as necessary. The scheduling tool developed in this paper is an extended version of methodology initially described in Erera et al. (2005) which incorporates new logic that ensures that dispatches meet union regulations.

The primary contributions of this paper can be summarized as follows:

- We develop and describe a driver dispatch methodology to manage drivers in accordance with the basic hours-of-service regulations imposed by the U.S. Department of Transportation (U.S. DOT) and in accordance with the most typical union rules faced by LTL carriers;

- We develop and describe a tactical driver domiciling methodology that can be used to determine the total required driver fleet size, the fraction of the driver fleet to domicile at each terminal, and the bid driver percentage and bid terminals for each domicile location, while maintaining a fixed system-wide total fraction of bid drivers;

- We present a computational study using truck movement data representative of operations at a major U.S. LTL carrier, which demonstrates the effectiveness of our domiciling methodology; and

- We study the impact of typical union rules on the number of drivers required to achieve a certain service level, and show that a decrease in the percentage of bid drivers in the system from 66% (the typical percentage found at unionized LTL carriers) to 50% results in a reduction of 20% in the required number of drivers.
The remainder of this paper is organized as follows. In Section 2, we describe LTL carrier operations. In Section 3, we discuss restrictions on drivers imposed by union regulations and their effect on driver management. In Section 4, we define the driver management problem formally. In Section 5, we introduce a dynamic driver dispatch scheme that takes U.S. DOT and union rules into account. In Section 6, we present a methodology for allocating drivers within a terminal network. In Section 7, we provide computational results based on real-life dispatch data from a large U.S. LTL carrier.

2 LTL Carrier Operations

LTL carriers operate fixed terminal networks to provide service to customers, with three categories of terminals: (1) satellite (or end-of-line) terminals, (2) breakbulk terminals, and (3) relay yards. Satellite terminals are usually the first and last stop for a shipment moving from an origin customer to a destination customer. Customer shipments originating in a small geographic area are collected using pickup and delivery vehicles and brought initially to a satellite terminal where they are sorted and consolidated into full trailer-loads for dispatch into the terminal network. In the United States, LTL carriers use 48-foot or 53-foot vans and combinations of 28-foot trailers, known as pups. In most states, at most two pups can be joined together and moved by a single tractor.

While it is sometimes possible at a satellite to build nearly-full loads with shipments destined for a common destination satellite, usually there is not enough freight to support reasonable service frequencies with this strategy. Dispatches are made therefore to breakbulk terminals, which provide another level of consolidation by receiving loaded trailers from multiple satellites, and sorting and reloading trailers for next destinations. A typical shipment travels 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. Even when a pup does not need to be unloaded at a breakbulk, it may be paired up with a different pup at a breakbulk for the outbound dispatch; this process is known as pup-matching. Breakbulk terminals are also frequently used as relay points for loads that need not be unloaded or pup-matched, but require a driver change due to long travel distance. Sometimes, carriers do not need unload-sort-reload capability in a particular geographic location, but do need the ability to pup-match or exchange drivers.
In such locations, the carrier may operate a relay yard only.

![Diagram of LTL network operations]

**Figure 1: LTL network operations**

The path of a shipment through the terminal network is identified by the sequence of terminals where the shipment is to be handled, where handling refers to unloading and loading of the shipment into some trailer. Most large network carriers operate a fixed load plan which identifies for each origin-destination pair the path a shipment follows and the associated service level, *i.e.*, the maximum time the shipment will spend in the linehaul network.

If enough freight accumulates on a given day, the carrier may be able to build a so-called direct trailer. A direct trailer contains shipments with a common destination, and therefore it need not be handled at intermediate terminals. This reduces handling costs and also saves time. Given a load plan and rules (or decision support technology) for generating direct trailers, LTL carriers build trailers as freight becomes available at terminals. Further, carriers use rules (or decision support technology) to match trailers together when necessary into *loads* for dispatch. Since freight flow is usually not balanced, it is also often required to move trailers empty to reposition them for future loading. Empty pups can be matched together, or with a loaded pup, when repositioning. As noted in Erera et al. (2005), long-run trailer balance is generally maintained in LTL systems if drivers pull a van or two pups, loaded or empty, whenever they are dispatched between terminals.
After trailers are built and matched, they are ready for dispatch; this time is referred to as the \textit{ready time} of the load. Many shipments are included in most loads, and the shipment with the most binding service commitment determines the latest time the load can be dispatched and still “make service,” \textit{i.e.}, arrive at the destination on time. This latest time is referred to as the \textit{cut time} of the load. To ensure that all freight meets its service requirements, each load needs to be dispatched between its ready and cut time.

Given a load plan and a method for forming loads from trailers, the remaining decision of an LTL carrier is to schedule drivers to move the loads. This is a critical component of the operation, given that driver expenses represent a large percentage of total costs.

3 Managing Driver Schedules With Union Regulations

A schedule for a driver consists of a sequence of timed dispatches (loaded or empty), along with interspersed rest (sleep) periods. Work periods between rest periods are known as \textit{duties}. A feasible schedule for a driver must satisfy U.S. DOT hours-of-service regulation, which limit the amount of driving time during and the total duration of each duty, and specify a minimum number of rest hours between consecutive duties.

Union regulations and bid rules also affect the feasibility of driver duties and schedules. In order to describe how, we first introduce some notation. Each satellite terminal in the terminal network is matched with a breakbulk terminal, called the \textit{parent terminal}. Shipments originating from and destined to a satellite are typically sent to its parent terminal, and then forwarded. While some networks may feature satellites with multiple parents that are used for freight headed in different directions, this complexity is ignored here. For convenience, we assume that a breakbulk terminal is parent of itself. The \textit{district} of terminal $T$, denoted by $D(T)$, is the set of terminals for which $T$ is the parent terminal, \textit{i.e.}, $D(T) = \{ t \mid T \text{ is parent of } t \}$.

3.1 Domiciles

As mentioned earlier, each driver has a home terminal called \textit{domicile}, where he returns with some regularity. The large majority of network carrier drivers are domiciled at breakbulk and relay terminals. While satellites typically do not have domiciled drivers, ones that are distant from their parent breakbulks sometimes do.
3.2 Driver types in unionized carriers

Drivers working for unionized carriers are grouped into two types. A bid driver is a prioritized driver, usually with seniority, whose dispatches are restricted by negotiated union rules. On the other hand, an extra-board driver is an unrestricted driver who can typically be dispatched to any terminal in the network at any point in time. Extra-board drivers are usually only domiciled at breakbulk or relay terminals.

The union rules that govern usage of bid drivers ensure regularity in drivers’ day-to-day schedules and improve quality-of-life measures. These rules are typically negotiated locally, and thus may vary from terminal to terminal. There are two types of restrictions: (1) location restrictions, and (2) time restrictions. Location restrictions limit the terminals that a bid driver can visit, based on his bid terminal.

Unions negotiate the driver work “owned” by a domicile terminal. If domicile A is the primary for a lane between two terminals A and B, then drivers domiciled at A have priority over drivers domiciled at B for dispatches between A and B, regardless of the origin of the load. Typically, for a given lane, the terminal that originates more freight running over the lane is designated the primary. Let P(A) be the set of terminals for which A is primary; then, bid drivers at A may select terminals in P(A) as bid terminals. The other bid terminal option is A itself, which is known as a district bid. Drivers which bid on the district are dispatched with loads between A and terminals in D(A). Note that if A is a satellite that domiciles drivers, the only bid terminal option is its parent breakbulk terminal.

Each bid driver, in addition to having a bid terminal, may also be classified by a subtype. The most common subtype is the ABA bid. ABA drivers are dispatched according to the following rule: the first dispatch after rest at the driver domicile A will be to a terminal in the district D(B) of his bid terminal B. Depending only on the driving time between A and B, there are two classes of ABA bid drivers that indicate whether the driver returns to A for rest nightly, or every other night:

1. ABA laydown drivers: If the driving time between A and B is greater than half of the maximum allowable driving time imposed by U.S. DOT, then the driver is an ABA laydown driver. After coming off rest at A, such a driver will end his next duty at a terminal in D(B). His subsequent duty after rest must return him to A for his next rest. Examples of potential duties for an ABA laydown driver can be found in Figure
2. *ABA turn drivers:* If the driving time between $A$ and $B$ is no greater than half of the maximum allowable driving time imposed by U.S. DOT, then the driver is an *ABA turn* driver. In this case, after coming off rest at $A$, the driver may travel to any terminal in $D(B)$, but is required to return to $A$ or any terminal in $D(A)$ for rest. Such a driver may not be dispatched to any terminal other than those in $D(A)$ and $D(B)$.

![Diagram](image)

Figure 2: Potential duties for an ABA laydown driver. $A' \in D(A)$, $B' \in D(B)$.

Note that if the carrier cannot dispatch a bid driver back to his domicile or domicile district with a loaded move to conform to the rules outlined above, the carrier will be forced to introduce an empty equipment dispatch. For *ABA* laydown drivers, this may result in a duty consisting entirely of empty travel.

Union rules may also specify time restrictions on driver usage. For example, a driver may have a *bid start time* indicating a time he is expected to be dispatched each day, or may have a *weekly plan* that limits the days of the week on which he can work. We will only consider location restrictions in this research, since they tend to have the largest impact on driver scheduling.

It should be clear that LTL carriers would prefer a driver fleet composed entirely of extra-board drivers, since such drivers have the flexibility to be dispatched to any terminal in the network, thus potentially maximizing the time they spend moving loads. However, the schedule regularity provided by bids is attractive to drivers. To manage this tradeoff,
unions negotiate a minimum percentage of bid drivers in the driver fleet. This percentage is typically 60–70 per cent for large national LTL carriers.

3.3 Local dispatch rules

Union rules prioritize the drivers that can be assigned to the dispatch of a load. A driver’s domicile, his type, his bid terminal, and the primary on the load lane all impact the priority ranking of available drivers. Given a load to be dispatched from $T_1$ to $T_2$, available drivers at $T_1$ can be grouped according to priority, from highest to lowest, as follows:

1. *In-motion drivers:* In-motion drivers are drivers that have completed a partial duty, and still have sufficient remaining drive and duty hours to execute the load (the driver’s domicile and bid type do not matter).

2. *Must-go drivers:* Must-go drivers are drivers becoming available off rest that must be dispatched to avoid violating work rules or paying extra charges. Determining which drivers are must-go drivers depends on which terminal is primary for the lane from $T_1$ to $T_2$. Must-go drivers are further prioritized in the following order:

   - If $T_1$ primary:
     
     (a) All drivers with domicile $T_2$ (note that these drivers are returning to their domicile).
     
     (b) ABA bid drivers domiciled at $T_1$ with bid terminal $T_2$.

   - If $T_2$ primary:
     
     (a) All ABA bid drivers with domicile $T_2$ (note that these are laydown drivers returning to domicile).
     
     (b) All extra-board drivers with domicile $T_2$ (note that these drivers are returning to their domicile).

3. *Domiciled extra-board drivers:* Domiciled extra-board drivers becoming available off rest are given priority over extra-board drivers not domiciled at $T_1$ or $T_2$.

4. *Non-domiciled extra-board drivers:* Extra-board drivers that are neither domiciled at $T_1$ nor $T_2$ are next in priority, and should be used before the final priority class.
5. **Should-not-go drivers:** Should-not-go drivers are those that should be reserved for
dispatch on different lanes. This class includes all bid drivers at domicile $T_1$ with a
bid terminal other than $T_2$. Although such drivers typically travel on their primary
lane, they may be dispatched as a last resort on other lanes.

4 **Driver Scheduling and Load Dispatching Problem**

Driver scheduling and load dispatching involves determining a set of driver schedules for
a given planning period and a given set of loads with minimum total operating cost, where
the set of schedules complies with U.S. DOT regulations and satisfies union rules.

Each load is characterized by an origin terminal, a destination terminal, a ready time, a
cut time, a set of preceding loads which must arrive at the origin terminal before a dispatch
can take place, and a set of succeeding loads which can depart from the destination terminal
only after the load has arrived there. In this research, it is assumed that the origin and
destination terminals of each load are connected by a lane that can be feasibly operated by
a driver within a single duty period; loaded trailers that travel further distances are usually
relayed through the terminal network on such lanes. We model separate loads along each
lane in such cases, with appropriately defined predecessor and successor relationships.

A set of drivers is available to move the loads. Each driver has a domicile, type informa-
tion that indicates his bid type or whether he is extra-board, a bid terminal if applicable,
a current location at the beginning of the planning period, and a ready time at the current
location.

To simplify the remaining presentation, consider the following notation:

- $\mathcal{T}$ The set of all terminals.
- $\mathcal{C}$ The set of all connections.
- $\mathcal{L}$ The set of all loads.
- $\mathcal{D}$ The set of all drivers.

The following parameters are used to model U.S. DOT driver regulations:

- $\tau_V$ The maximum number of driving hours in a duty between rests.
- $\tau_U$ The maximum duration (hours) of a duty (including waiting time) between rests.
- $\tau_R$ The minimum duration (hours) of a rest.

In this paper, we use current U.S. DOT driver regulations: $\tau_V = 11$, $\tau_U = 14$, and
\( \tau_R = 10. \)

Each load \( \ell \in \mathcal{L} \) has the following attributes:

\begin{itemize}
  \item \texttt{orig}(\ell) \quad \text{The origin terminal.}
  \item \texttt{dest}(\ell) \quad \text{The destination terminal.}
  \item \texttt{ready}(\ell) \quad \text{Earliest dispatch time.}
  \item \texttt{cut}(\ell) \quad \text{Latest dispatch time.}
  \item \texttt{prec}(\ell) \quad \text{The set of immediate predecessor loads.}
  \item \texttt{suc}(\ell) \quad \text{The set of immediate successor loads.}
\end{itemize}

Each driver \( d \in \mathcal{D} \) has the following attributes, representing his state at the beginning of the planning period:

\begin{itemize}
  \item \texttt{domicile}(d) \quad \text{The domicile.}
  \item \texttt{type}(d) \quad \text{Type of the driver.}
  \item \texttt{bidterm}(d) \quad \text{Bid terminal for bid drivers.}
  \item \texttt{location}(d) \quad \text{The current location.}
  \item \texttt{ready}(d) \quad \text{Earliest dispatch time at the current location.}
  \item \texttt{maxdrive}(d) \quad \text{Remaining driving time.}
  \item \texttt{maxduty}(d) \quad \text{Remaining duty time.}
\end{itemize}

Note that driver \( d \) may be enroute to \texttt{location}(d) at the beginning of the planning period. In this case, and in the case where a driver is waiting (not resting) at the current location, \texttt{maxdrive}(d) and \texttt{maxduty}(d) initially may be less than \( \tau_V \) and \( \tau_U \) respectively; otherwise, \texttt{maxdrive}(d) = \( \tau_V \) and \texttt{maxduty}(d) = \( \tau_U \). The parameter \texttt{type}(d) indicates if the driver \( d \) is a bid driver or an extra-board driver; in this research, we assume that all bid drivers are the \( \text{ABA} \) type. For bid drivers, \texttt{bidterm}(d) indicates the bid terminal.

Each connection in \( \mathcal{C} \) links two terminals \( A, B \in \mathcal{T} \) and has the following attributes:

\begin{itemize}
  \item \texttt{time}(A, B) \quad \text{The driving time (in hours) between terminals } A \text{ and } B.
  \item \texttt{rtime}(A, B) \quad \text{The run time (in hours) between terminals } A \text{ and } B.
  \item \texttt{primary}(A, B) \quad \text{The primary terminal (if applicable) on the lane from } A \text{ to } B.
\end{itemize}

The set of connections defines the linehaul network. As mentioned earlier, if terminals \( A \) and \( B \) are connected, \( \text{time}(A, B) \leq \tau_V \). If this driving time is long, a driver may need to take short breaks along the way. These short breaks are included in \texttt{rtime}(A, B) and
count towards duty duration. Note that $\text{rtime}(A, B) \leq \tau_U$. Not every terminal-to-terminal connection defines a lane with loaded moves, and further not all lanes have an associated primary terminal. When applicable, $\text{primary}(A, B)$ represents the primary terminal for the lane and is equal to either $A$ or $B$.

As mentioned at the beginning of this section, the objective is to minimize total operating costs over a period of time. Our approach, to be described next, aims to accomplish that objective indirectly by attempting simultaneously to minimize the number of drivers dispatched, to maximize the utilization of dispatched drivers, to minimize empty travel, to minimize the number of foreign beds, and to return the drivers to their domiciles as often as possible. Getting drivers home regularly improves their quality of life and reduces costly driver turnover, which can be substantial at an LTL carrier.

5 Solving Driver Scheduling and Load Dispatching Problems

Before we describe our approach, we briefly review the literature on transportation operator scheduling. For systems with fixed task dispatch times, such as airlines, transit, and passenger rail, large-scale mixed-integer programming approaches have been developed. Barnhart et al. (2003) provide a thorough overview of tactical airline crew scheduling problems and solution approaches. Such approaches typically require much computational effort; recent advances reported in Elhallaoui et al. (2005) still indicate computation times greater than an hour for scheduling problems with more than 1,500 tasks. The stream of research by Powell and collaborators (see, e.g., Powell (2003) and Powell and Topaloglu (2003)) is primarily concerned with approaches for managing uncertainty in such problems. However, Powell et al. (2002) address a deterministic driver management problem quite similar to the one considered herein, and apply an approximate dynamic programming solution approach. While the method yields good results for similarly-constrained problems with computation times of about an hour, it uses a discrete representation of time and computational effort depends on the fineness of the discretization.

In this paper, we extend an iterative procedure for generating driver schedules that comply with U.S. DOT hours-of-service regulations, initially presented in Erera et al. (2005). In this section, we briefly outline how the scheme functions and then describe how we have extended it to properly and efficiently handle the added complexity of union rules.
A high level description of the driver scheduling and load dispatching scheme is given in Algorithm 1.

**Algorithm 1 High level overview of the driver scheduling and load dispatching scheme**

```
while there exist undispached loads do
    Select a load \( \ell \) to be dispatched next.
    Generate a set of duties that include \( \ell \) as the first loaded move of the duty
    Generate a set of feasible trips by matching drivers with duties
    Evaluate the quality of each trip with respect to a set of objectives
    Select and dispatch a high-quality trip
    Update system information
end while
```

Next, we briefly discuss the key steps of the scheme.

*Load selection*

The scheme iterates over the loads in order of increasing cut times. That is, it always selects the undispached load \( \ell \) with minimum \( \text{cut}(\ell) \), breaking ties arbitrarily, and attempts to dispatch it.

*Duty generation*

Given selected load \( \ell \), the scheme generates a set of duties that includes \( \ell \) as the first loaded move. Each such duty consists of one or more loaded moves and potentially one or more empty moves, where each load \( \ell' \) in the duty is dispatched no earlier than \( \text{ready}(\ell') \) and no later than \( \text{cut}(\ell) \). Further, the total driving time of each duty is no greater than \( \tau_V \) and its total duration no greater than \( \tau_U \).

Duties are generated using a truncated depth-first recursive procedure. Each recursive step starts with a base duty, initially just the load \( \ell \). Using a limit \( W \) on the allowable waiting time between loaded movements, loads \( \ell \) originating at the final terminal visited by the base duty are identified with \( \text{ready}(\ell) \) no greater than the arrival time at this terminal plus \( W \). For each such load, it is checked whether the duty consisting of the base duty and the load \( \ell \) remains feasible with respect to \( \tau_V \) and \( \tau_U \). Dispatch times for each move in the duty are set such that the duty requires minimum duration, and is dispatched as early as possible. If so, the duty is added to the list of generated duties and a new recursion is
invoked where the base duty is the newly generated duty, \textit{i.e.}, load $\ell$ is the last load in the new base duty. The recursion is truncated if an upper limit on the number of generated duties is reached.

\textit{Trip generation}

Once a set of duties is generated, the set of all feasible \textit{trips} is generated by matching drivers available at the origin terminal of $\ell$ (and thus the origin terminal of all the duties) with duties. Driver $d$ can be matched to a duty if $\text{LOCATION}(d)$ is $\text{ORIG}(\ell)$, $\text{READY}(d)$ is no later than the dispatch time of $\ell$, and $\text{MAXDRIVE}(d)$ and $\text{MAXDUTY}(d)$ are compatible with the duty.

If no feasible trips exist using drivers at $\text{ORIG}(\ell)$, trips are created instead for drivers at other terminals. To do so, an empty move is prepended to the duty from $\text{LOCATION}(d)$ to $\text{ORIG}(\ell)$, but only if the resulting extended duty is still time-feasible. To limit empty repositioning costs, this process is conducted in two phases. First, all drivers at a predefined set of \textit{nearby} terminals are considered. Second, if no trips can be generated using nearby terminals, trips are generated for drivers at more distant terminals with historical empty moves to this terminal; this process considers terminals one at a time in order of increasing distance from $\text{ORIG}(\ell)$.

Lastly, since one of the objectives is to minimize foreign bed cost, additional trips are generated that append an empty move to $\text{DOMICILE}(d)$ as the final leg to return driver $d$ to domicile when such a move is time-feasible.

\textit{Trip scoring and selection}

Each generated trip is evaluated with respect to the various problem objectives. Once the best trip is identified, it is scheduled for dispatch, \textit{i.e.}, the driver and duty are dispatched to cover the initial load (and all other loads included in the duty).

\textit{Updating system information}

Once a trip is selected for dispatch, system information is updated to reflect this selection. For dispatched driver $d$, $\text{LOCATION}(d)$ is updated to the destination of the trip, and $\text{READY}(d)$ is updated to the arrival time of the trip at the destination plus the mandatory rest time $\tau_R$. In addition, given the dispatch times for loads included in the duty, $\text{READY}(\hat{\ell})$ is updated for each load $\hat{\ell}$ that is a successor to a dispatched load (or a successor to a
successor, etc.). Similarly, \( \text{cut}(\ell) \) is updated for each load \( \ell \) that is a predecessor to a dispatched load (or a predecessor to a predecessor, etc.).

To ensure that driver schedules satisfy union rules, the scheduling scheme presented above is now extended so that the location restrictions and local dispatch rules outlined in Section 3 are respected.

In-motion drivers have the highest priority. The driver scheduling scheme always generates a complete duty for a driver, i.e., the driver rests at the end of the assigned duty. During each iteration, no drivers in the system are in motion except the driver assigned to the duty in the currently selected trip. Thus, the duty extension mechanism and the complete duty property guarantee that the approach also gives highest priority to in-motion drivers.

Given a selected load \( \ell \) for dispatch, drivers can be grouped into the prioritization hierarchy implied by the dispatch rules: must-go drivers, domiciled extra-board drivers, non-domiciled extra-board drivers, and should-not-go drivers. To properly account for this prioritization, the scheduling scheme is modified to generate duties and trips for each group in sequence, and to dispatch the best generated trip within a priority group. If no trips are generated for a group, the scheme moves to the next group. Since the drivers within each group may face location restrictions, the duty generation mechanism is also enhanced to respect these rules.

A high level overview of the modified driver scheduling scheme is given in Algorithm 2.

Generating duties based on driver priority groups

To generate feasible duties for all driver priority groups, we allow the duty generation mechanism to limit both the (intermediate) terminals a driver can visit and the terminal at which the duty terminates. We denote these two sets by \( \mathcal{I} \) and \( \mathcal{E} \) respectively. In the case where \( \mathcal{I} = \mathcal{E} = \mathcal{T} \), no location restrictions are present, and the duty generator functions as previously described.

We make two modifications to enforce these restrictions:

1. Each candidate load \( \ell \) for expanding a base duty must be such that \( \text{dest}(\ell) \in \mathcal{I} \); and
2. If \( \text{dest}(\ell) \in \mathcal{E} \) then the duty is marked saved, otherwise it is marked temporary.

Only saved duties are feasible, and are added to the set of generated duties. Temporary
Algorithm 2 High level overview of the driver scheduling scheme with union rules.

while there exist undispached loads do

    Select a load \( \ell \) to be dispatched next

    for each driver priority subgroup in the order described in Section 3.3 do

        Generate a set of duties that include \( \ell \) as the first loaded move of the trip
        Generate a set of trips by matching drivers with duties
        if any trip generated then
            Assign a score to each trip indicating its desirability
            Select a trip with the highest score
            Break for loop
        end if
    end for

    Update system information
    if dispatched driver \( d \) is type ABA laydown then
        Create a return trip for driver \( d \)
    end if
end while

duties are created since such a duty may become feasible by appending additional load(s) in later steps of the recursion.

Expanding duties in this way may lead the recursive search to explore part of the network where it is not possible to return to any one of the terminals in \( \mathcal{E} \), which results in wasted computation time. To avoid such a situation, we check if a base duty can still be extended (possibly with an empty move) to return to one of the terminals in \( \mathcal{E} \) within the remaining driving time. If such a return is not possible, we end the expansion of the duty immediately.

Location restrictions for different driver priority groups

The first class of drivers that is eligible for dispatch is the set of must-go drivers. Among the must-go drivers, foreign drivers \( d \) such that \( \text{domicile}(d) \) is \( \text{dest}(\ell) \) have the highest priority. If \( \text{orig}(\ell) \) is \( \text{primary}(\text{orig}(\ell), \text{dest}(\ell)) \), the next priority group is composed of ABA bid drivers with \( \text{domicile}(d) \) equal to \( \text{orig}(\ell) \) and \( \text{bidterm}(d) \) equal to \( \text{dest}(\ell) \). If these drivers are ABA laydown drivers, then duties are generated by setting \( \mathcal{I} = \mathcal{T} \) and \( \mathcal{E} = D(\text{dest}(\ell)) \). Alternatively, if these drivers are ABA turn drivers, then \( \mathcal{I} = \mathcal{E} = \)
\[ D(\text{orig}(\ell)) \cup D(\text{dest}(\ell)) \text{ and } \mathcal{E} = D(\text{orig}(\ell)). \]

The next two groups of drivers eligible for dispatch are the set of domiciled extra-board drivers at \(\text{orig}(\ell)\) and then foreign extra-board drivers. Drivers from both groups can be dispatched to any terminal, and thus duties are generated by setting \(\mathcal{I} = \mathcal{T}\) and \(\mathcal{E} = \mathcal{T}\).

The lowest priority group of drivers is the set of should-not-go bid drivers \(d\) with \(\text{domicile}(d)\) equal to \(\text{orig}(\ell)\) but \(\text{bidterm}(d) = T' \neq \text{dest}(\ell)\). For these drivers, duties are generated by setting \(\mathcal{I} = \mathcal{T}\) and \(\mathcal{E} = D(T')\).

**Expanding short duties**

If we consider only loaded moves for duty generation for bid driver groups, it is likely that the resulting duties will include few loads. More importantly, few if any feasible duties may be generated in some cases. This is especially true for ABA turn drivers. In the duty generation step, therefore, if a duty cannot be expanded with a load and the total driving time of the duty is below a threshold, then an empty move to each one of terminals in \(\mathcal{E}\) is appended to the duty if time feasible. Duty expansion continues after the empty move. Since the destination of the added empty move is in \(\mathcal{E}\), we are guaranteed not to violate bid requirements of the drivers and the resulting duty is feasible. If we can further extend the duty with loaded moves, then the utilization of the driver also increases.

**Limiting the number of generated duties**

When generated duties for bid drivers, we continue to use a truncation limit to improve computational performance. In this case, however, we truncate the generation when the sum of the number of temporary and saved duties reaches the limit. To increase the number of saved duties generated prior to truncation, and to improve the quality of the duties generated, we prioritize loads \(\ell\) considered during each expansion step. Loads \(\ell\) with \(\text{dest}(\ell)\) in \(\mathcal{E}\) are considered before other loads at each level in the search tree.

**Trip generation**

Trip generation only differs from the original scheme when empty moves are prepended to duties. Again, if no trips are generated for drivers at \(\text{orig}(\ell)\) in any group, trips are created for drivers located at other terminals with appropriate empty moves prepended when time feasible. However, only extra-board drivers at these other terminals are eligible for such trips.
Return trip for ABA laydown driver

If an ABA laydown driver $d$ is selected for dispatch from his domicile terminal $\text{orig}(\ell)$, such a driver always rests at a terminal in $D(\text{bidterm}(d))$. Since the driver must return to domicile in his subsequent duty, the scheme immediately generates the return trip for $d$.

We consider a hierarchy of possibilities for the return trip. First, assume that $d$ rests at his bid terminal. An attempt is made to generate a set of feasible duties using the first available load $\ell'$ with $\text{dest}(\ell') = \text{domicile}(d)$. If no such duties can be generated, the scheme next attempts to generate duties that involve an intermediate stop on the way to the driver's domicile. If still no duties are generated, then the driver is sent on an empty trip directly to $\text{domicile}(d)$. Now assume that $d$ is resting at a terminal in $D(\text{bidterm}(d))$. In addition to the above logic, we also consider duties that begin with an empty move to $\text{bidterm}(d)$, since it is more likely that loaded dispatches are available from that breakbulk back to his domicile. Again, if no duties are generated, the driver is sent home empty.

6 Driver Fleet Sizing and Allocation Methodology

The driver fleet sizing and allocation methodology we propose is an iterative scheme that requires a fixed terminal network with a subset of terminals $\mathcal{M} \subseteq \mathcal{T}$ that can be used as domiciles, a set of loads $\mathcal{L}$ to be dispatched, and a minimum ratio $\rho$ of bid drivers to all drivers in the system. At its conclusion, the outputs are a suggested driver fleet size, an allocation of the driver fleet to domiciles, and for each domicile the fraction of bid drivers as well as the fraction of drivers assigned to each bid terminal.

The methodology is not designed to be completely automated, and requires user intervention at two distinct decision points. The scheme uses two phases: in the first phase, a fleet size and allocation of bid drivers is determined, and in the second phase, an allocation of extra-board drivers is determined for a fleet size no greater than a limit imposed by ratio $\rho$. During each phase, an initial allocation of some relatively small set of drivers that is certain not to be capable of serving all loaded dispatches is determined via a simple allocation rule. Then, the driver scheduling tool is used to plan best schedules for the drivers in this set. Using the loads that remain unserved, the simple allocation rule is used again to allocate (and re-allocate) a small set of $m_{\text{new}}$ new drivers along with drivers that were not dispatched during the previous scheduling run. This iterative process of scheduling and
allocating is repeated in each of the two phases until the user decides that enough drivers have been added, or until an upper bound on the number of drivers (for example, the one implied by $\rho$ during the extra-board phase) has been reached.

Algorithm 3 gives a high-level description of the scheme. Phase I begins with $m_{\text{init}}$ bid drivers. Since bid drivers have priority, considering them separately and first guarantees that the approach is not biased by the presence of extra-board drivers who might be dispatched with “bid work” by the scheduling heuristic; such an effect is especially likely in the initial iterations when the bid driver fleet is still small relative to the number of available loads. Phase I iterations continue until the user decides, based on performance statistics, that an appropriate bid fleet size has been attained. The primary performance statistics to consider are the percentage of loaded movements that are served, and driver performance statistics such as average driving time, average duty time, and average number of trips per driver per week. In phase II, extra-board drivers are added to the system iteratively, using only the loads that remain unserved by bid drivers after phase I. The phase II iterative allocation process concludes when the ratio of bid drivers to all drivers reaches $\rho$. 
Algorithm 3 High level description of the driver allocation algorithm.

\[ m_{bid} \leftarrow m_{init}, \quad m_{eb} \leftarrow 0 \]

\{Start Phase I\}

\textbf{repeat}

\hspace{1em} Run Algorithm 2

\hspace{1em} Calculate remaining bid work for all bid lanes and terminals

\hspace{1em} \( \mathcal{D}_a = m_{new} \) new bid drivers \( \cup \) unassigned bid drivers in previous iteration

\hspace{1em} Allocate \( \mathcal{D}_a \) to domiciles and to bid terminals

\hspace{1em} \( m_{bid} \leftarrow m_{bid} + m_{new} \)

\hspace{1em} Reset all load dispatches

\textbf{until} (No significant increase in marginal percentage load coverage or degradation in driver performance statistics)

User selects appropriate bid driver fleet size \( m_{bid} \) and corresponding driver allocation from discrete choices generated in Phase I

\{Start Phase II\}

Remove all bid drivers and loads covered by them.

\textbf{repeat}

\hspace{1em} Run Algorithm 2

\hspace{1em} Calculate remaining work for all terminals

\hspace{1em} \( \mathcal{D}_a = m_{new} \) new extra-board drivers \( \cup \) unassigned extra-board drivers in previous iteration

\hspace{1em} Allocate \( \mathcal{D}_a \) to domiciles.

\hspace{1em} \( m_{eb} \leftarrow m_{eb} + m_{new} \)

\hspace{1em} Reset all load dispatches

\textbf{until} \( (m_{eb} \geq \frac{1-\rho}{\rho} m_{bid}) \)

\textit{Calculating remaining work}

We define \textit{work} to be the total driving time required by a set of loads. Remaining work refers to driving time required by as of yet undispatched loads. First, we consider calculation of remaining work for bid drivers. Bid drivers domiciled at terminal \( T_d \) are responsible for covering loads between \( T_d \) and each terminal in \( P(T_d) \), as well between \( T_d \) and its district terminals \( D(T_d) \) if applicable. Note that this includes loads moving in both directions.
on these lanes. Of the drivers domiciled at $T_d$, those that bid on $T_b \in P(T_d) \cup \{T_d\}$ are responsible for covering all loads moving back and forth between $T_d$ and any terminal in $D(T_b)$. Let $n_u(T_1, T_2)$ denote the number of unassigned loads from terminal $T_1$ to $T_2$ and let $W_{bid}(T_d, T_b)$ denote the remaining amount of bid work that a driver domiciled at $T_d$ with bid terminal $T_b$ should cover, i.e.,

$$W_{bid}(T_d, T_b) = \sum_{T' \in D(T_b)} (n_u(T_d, T') \times \text{TIME}(T_d, T')) + \sum_{T' \in D(T_b)} (n_u(T', T_d) \times \text{TIME}(T', T_d))$$

The total unassigned bid work for a domicile $T_d \in \mathcal{M}$ is

$$W_{bid}(T_d) = \sum_{T' \in P(T_d)} W_{bid}(T_d, T'),$$

and the total unassigned work is

$$W_{bid} = \sum_{T_d \in \mathcal{M}} W_{bid}(T_d).$$

The logic for calculating the remaining work in phase II is similar to phase I. However, the definition of what constitutes work for a extra-board driver is slightly different.

The primary work responsibility of an extra-board driver domiciled at $T_d$ is comprised of all outbound loads that are not covered by bid drivers. A reasonable approach then is to locate extra-board drivers in phase II based on the total outbound driving time required by unassigned loads. Let $N(T_d)$ be the set of destination terminals for which there are unassigned loads originating at terminal $T_d$. First, we calculate the amount of unassigned work from $T_d$ to each $T_n \in N(T_d)$:

$$W_{eb}(T_d, T_n) = n_u(T_d, T_n) \times \text{TIME}(T_d, T_n).$$

The total unassigned work for a domicile $T_d \in \mathcal{M}$ is then

$$W_{eb}(T_d) = \sum_{T' \in N(T_d)} W_{eb}(T_d, T'),$$

and the total unassigned work is

$$W_{eb} = \sum_{T_d \in \mathcal{M}} W_{eb}(T_d).$$

*Allocating drivers to domiciles and bid terminals*
Denote the set of drivers that is added at each iteration by $D_a$. Note that $D_a$ includes both the set of new drivers added to the system, as well as drivers that were not dispatched in the previous iteration.

In each iteration, the set of driver $D_a$ is allocated to domicile terminals and bid terminals using a simple proportional allocation rule based on unassigned work. The number of new bid drivers that a terminal receives in Phase I is given by

$$m_{\text{new bid}}(T_d) = \left\lfloor \frac{W_{\text{bid}}(T_d)}{W_{\text{bid}}} \times |D_a| \right\rfloor .$$

Next, we determine the bid terminal for each bid driver allocated to $T_d$. The number of drivers with bid terminal $T_b \in P(T_d)$ will be proportional to the unassigned work between $T_d$ and $T_b$:

$$m_{\text{new bid}}(T_d, T_b) = \left\lfloor \frac{W_{\text{bid}}(T_d, T_b)}{W_{\text{bid}}(T_d)} \times m_{\text{new bid}}(T_d) \right\rfloor .$$

In Phase II, the allocation of extra-board drivers is performed similarly:

$$m_{\text{new eb}}(T_d) = \left\lfloor \frac{W_{\text{eb}}(T_d)}{W_{\text{eb}}} \times |D_a| \right\rfloor .$$

In Phase II, we do not need to decide on bid terminals because all the new drivers are extra-board and thus do not have a bid terminal.

**Driver allocation scheme initialization**

Phase I is initialized with $m_{\text{init}}$ bid drivers to save computing time. We decided to use an initial number of drivers that is approximately equal to half the number of bid drivers we expect the system to require. Initial allocation of drivers is performed by considering all loads as undispatched, and then applying the proportional allocation scheme described previously.

**Combining results from different load set inputs**

The driver fleet sizing and allocation scheme described above is designed to use a single set of loads over a fixed planning period to determine an allocation. Since LTL operations are naturally divided into week-long periods, it is appropriate to use a load set that consists of a week’s worth of loads. However, since a week is a relatively short period from a tactical planning perspective and since the loads that must be served will likely vary from week to week, an effective allocation scheme should use multiple load sets to avoid any bias from a
particular week’s worth of loads. When doing so, it is good practice to use planning periods of equal length.

The method we employed is to determine a fleet size $m_{bid}$ (and thus $m_{eb}$ as $m_{eb} \approx \frac{1}{\rho} m_{bid}$) by analyzing the results of the Phase I iterations over several different load sets, and selecting a number of bid drivers $m_{bid}$ that leads to satisfactory system performance statistics for each of them. Given a value $m_{bid}$ and the resultant allocation of drivers to domiciles and bid terminals for each load set, Phase II is executed for each of the load sets.

The result of this process will be an allocation of $m_{bid}$ bid drivers to domiciles and bid terminals and an allocation of $m_{eb}$ extra-board drivers to domiciles that likely differs slightly for each of the load sets. In order to determine a final allocation, therefore, we combine the results for each of the load sets using regression models with the objective of minimizing the total absolute deviation of the final allocated number of drivers from the actual allocated numbers for each of the load sets.

For bid drivers, let

- $m_{bid}^i(T_d, T_b)$ be the number of bid drivers assigned to terminal $T_d$ with bid terminal $T_b$ for load set $i$, and

- $\tilde{m}_{bid}(T_d, T_b)$ be the decision variable for the final number of drivers at $T_d$ with bid terminal $T_b$.

We then solve the following optimization problem to determine the final bid driver allocation:

$$\min_{T_d \in \mathcal{M}, T_b \in \mathcal{P}(T_d)} \sum_i \sum_{T_d \in \mathcal{M}, T_b \in \mathcal{P}(T_d)} |\tilde{m}_{bid}(T_d, T_b) - m_{bid}^i(T_d, T_b)|$$

subject to

$$\sum_{T_d \in \mathcal{M}, T_b \in \mathcal{P}(T_d)} \tilde{m}_{bid}(T_d, T_b) = m_{bid}$$

For extra board drivers let

- $m_{eb}^i(T_d)$ be the number of extra board drivers assigned to domicile $T_d$ for load set $i$, and

- $\tilde{m}_{eb}(T_d)$ be the decision variable for the final number of extra board drivers domiciled $T_d$. 

23
Similarly, we solve the following problem to determine the final extra-board driver allocation:

$$\min \sum_{T_d \in \mathcal{M}} \sum_i |\tilde{m}_{eb}(T_d) - m^i_{eb}(T_d)|$$

subject to

$$\sum_{T_d \in \mathcal{M}} \tilde{m}_{eb}(T_d) = m_{eb}$$

7 Computational Study

We test the driver fleet sizing and allocation scheme using load sets representing three consecutive weeks of real dispatch data for a major national LTL company with a linehaul network consisting of 455 terminals (satellite, breakbulk, and relay terminals) and 12,215 connections (links between terminals). Each week begins on a Monday at 12:00 a.m., and concludes on a Sunday at 11:59 p.m. We assume that drivers may be domiciled at any terminal. Again, if drivers are domiciled at a satellite, they must be bid drivers with the parent breakbulk as the bid terminal. Table 1 gives the total number of loads to be dispatched in each of the three weeks.

<table>
<thead>
<tr>
<th>Week</th>
<th># Loads</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>23,289</td>
</tr>
<tr>
<td>W2</td>
<td>23,037</td>
</tr>
<tr>
<td>W3</td>
<td>22,234</td>
</tr>
</tbody>
</table>

Table 1: Number of loads to be dispatched during each test week for the computational study.

We conduct three experiments. The first experiment is designed to determine the value of the iterative allocation scheme described in Algorithm 3 when compared to a more naive scheme that allocates drivers proportionally based on a single assessment of the amount of work to be performed at terminals. The results demonstrate that the iterative scheme substantially outperforms simpler strategies. The second experiment investigates the system performance impacts of varying the number of drivers while keeping $\rho$ (the percentage of bid drivers) constant. Such an experiment demonstrates the potential for the technology to aid
an LTL carrier in making decisions regarding both the driver fleet size and the allocation of this fleet to domiciles and bid terminals. In the final experiment, we use the scheme to determine fleet sizes and allocations in the case where $\rho = 0$ and $\rho = \frac{1}{2}$, for comparison to the base case where $\rho = \frac{2}{3}$. Such an experiment allows the assessment of the impact of union regulations on the operations of an LTL carrier, i.e., the additional costs incurred. Such information may be extremely valuable during negotiations with the unions.

7.1 Experiment 1: Comparison of Different Driver Allocation Algorithms

In this computational experiment, we compare two schemes for allocating 3,000 drivers with $\rho = \frac{2}{3}$. In the base allocation scheme, we allocate the entire driver fleet in two steps, both using proportional allocation rules. To do so, we first combine all of the loads from the three separate instances into a single large load set. Next, we calculate the total bid work per lane, per terminal, and for the total system using the formulae in Section 6, and use the bid proportional allocation formulae with $|D_a| = 2,000$ to determine domiciles and bid terminals for the bid drivers. Finally, we determine a proportional allocation of the 1,000 extra-board drivers to domiciles by calculating the total outbound work for each terminal using the set of all loads, and then using the extra-board proportional allocation formulae with $|D_a| = 1,000$.

The second allocation scheme uses the approach summarized in Algorithm 3 for a fixed fleet size of $m_{bid} = 2,000$ and $m_{eb} = 1,000$ to determine allocations for instances W1, W2, and W3. Then, the optimization formulations given in Section 6 are used to generate a final best allocation.

Given the final allocations from each scheme, the scheduling tool from Section 5 is used to create driver schedules for each of the three load set instances W1, W2, and W3 to generate the results for comparison. Table 2 summarizes and compares the primary driver performance statistics under each of the allocation schemes. The third column reports the number of drivers assigned for that week. The fourth column reports the percentage of loads that are dispatched. The fifth column reports the number of empty driver miles as a percentage of the number of total driver miles. The sixth column reports the number of foreign beds as a percentage of the total number of rests. The seventh and eighth columns report the average driving time and duty time of all assigned driver trips, in hours. Finally, the last column reports the number of tours executed during the week per dispatched driver.
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Date</th>
<th># Drivers Assigned</th>
<th>% Loads Dispatched</th>
<th>% Empty</th>
<th>% Foreign Beds</th>
<th>Average Drive Time</th>
<th>Average Duty Time</th>
<th>Tours per driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Allocation</td>
<td>W1</td>
<td>2118</td>
<td>82.99</td>
<td>7.45</td>
<td>43.47</td>
<td>9.46</td>
<td>10.66</td>
<td>4.19</td>
</tr>
<tr>
<td></td>
<td>W2</td>
<td>2144</td>
<td>80.51</td>
<td>8.21</td>
<td>45.08</td>
<td>9.38</td>
<td>10.55</td>
<td>4.01</td>
</tr>
<tr>
<td></td>
<td>W3</td>
<td>2117</td>
<td>83.03</td>
<td>7.57</td>
<td>43.82</td>
<td>9.42</td>
<td>10.58</td>
<td>4.08</td>
</tr>
<tr>
<td>Algorithm 3</td>
<td>W1</td>
<td>2621</td>
<td>98.52</td>
<td>9.15</td>
<td>55.23</td>
<td>9.37</td>
<td>10.62</td>
<td>4.28</td>
</tr>
<tr>
<td></td>
<td>W2</td>
<td>2581</td>
<td>97.05</td>
<td>9.35</td>
<td>56.53</td>
<td>9.21</td>
<td>10.46</td>
<td>4.20</td>
</tr>
<tr>
<td></td>
<td>W3</td>
<td>2539</td>
<td>98.66</td>
<td>8.82</td>
<td>52.40</td>
<td>9.39</td>
<td>10.64</td>
<td>4.12</td>
</tr>
</tbody>
</table>

Table 2: Comparison of two different driver allocation schemes

The difference in performance is surprisingly large. With the driver allocation that results from the base allocation scheme, only about 82% of loads are dispatched and only a little over $\frac{2}{3}$ of the drivers are used. On the other hand, with the allocation that results from the iterative driver allocation scheme, about 98% of loads are dispatched using slightly less than $\frac{9}{10}$ of the drivers. These results demonstrate that capturing the complex interactions between available loads and the drivers moving through the terminal network is important when determining an allocation. The iterative allocation scheme captures these interactions by simulating and evaluating the system after adding each set of new drivers, and adjusting the allocations based on the information obtained.

It is important to note that the driver scheduling tool does not allow loads to be delivered late. Therefore, if a load cannot be dispatched between its ready and cut time, contrary to practice, it is not dispatched at all. As such, the driver scheduling tool focuses on on-time performance. Major LTL carriers typically have around a 97% on-time performance. Therefore, a 97% or larger dispatch rate for our driver scheduling tool is acceptable. (The driver scheduling tool can be easily modified to allow loads to be delivered late by dynamically modifying the cut time, when loads cannot be dispatched after a number of tries.)

The effects of iterating and adapting are best illustrated by examining the differences between the driver allocations produced by the two algorithms. Table 3, for example, presents an analysis of those breakbulk terminals where the number of district drivers allocated by two algorithms differs by more than ten drivers. The maximum and average driving time columns calculate these statistics over the lanes from the breakbulk $T$ to each terminal in the district $D(T)$. It is clear that the base allocation scheme allocates too
few district drivers to terminals with a high average district driving time, and too many drivers to terminals with a low average district driving time. This is an indication that the simple scheme used to determine the amount of work that needs to be performed by drivers domiciled at a terminal is unable to capture the intricate interactions between loads. Allocation schemes that rely entirely on the total driving time required by loads that need to be covered by drivers domiciled at a terminal may be insufficient. U.S. DOT regulations impose a driving time limit $\tau_V$ of eleven hours. In a district where the average district move is close to or greater than half the maximum driving time in a day, a driver may only be able to cover one load before resting. On the other hand, when the average district move is short, the driver may be able to cover multiple loads in a duty before rest. Algorithm 3 is not as sensitive to this phenomenon, because its iterative nature allows it to react to the fact that Algorithm 2 builds duties with drivers covering multiple loads. Algorithm 3 receives feedback at each iteration that shows where the work calculation erred in its prediction of the load coverage of a driver.

Table 3: District driver allocation vs. driving time

<table>
<thead>
<tr>
<th>Terminal</th>
<th># Drivers (Algorithm 3)</th>
<th># Drivers (Base Allocation)</th>
<th>Difference</th>
<th>Max. Driving Time District</th>
<th>Avg. Driving Time District</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47</td>
<td>33</td>
<td>14</td>
<td>9.4</td>
<td>5.30</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>9</td>
<td>12</td>
<td>10.6</td>
<td>6.74</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>17</td>
<td>-10</td>
<td>4.3</td>
<td>2.19</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>22</td>
<td>-11</td>
<td>5.9</td>
<td>2.99</td>
</tr>
<tr>
<td>5</td>
<td>57</td>
<td>68</td>
<td>-11</td>
<td>5</td>
<td>2.87</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>23</td>
<td>-13</td>
<td>5</td>
<td>2.57</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>23</td>
<td>-16</td>
<td>5.1</td>
<td>3.24</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>27</td>
<td>-16</td>
<td>5.6</td>
<td>2.92</td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>41</td>
<td>-19</td>
<td>4.2</td>
<td>1.96</td>
</tr>
</tbody>
</table>

Table 4: Terminals with more extra board drivers under the base allocation scheme

<table>
<thead>
<tr>
<th>Terminal</th>
<th># Extra-board (Algorithm 3)</th>
<th># Extra-board (Base Allocation)</th>
<th>Difference</th>
<th>Outbound Work Rank</th>
<th>% Outbound Work Terminal is Primary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>45</td>
<td>-41</td>
<td>8</td>
<td>58.77</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>47</td>
<td>-40</td>
<td>7</td>
<td>74.79</td>
</tr>
<tr>
<td>3</td>
<td>105</td>
<td>141</td>
<td>-36</td>
<td>13</td>
<td>73.17</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>40</td>
<td>-26</td>
<td>10</td>
<td>61.96</td>
</tr>
<tr>
<td>5</td>
<td>29</td>
<td>53</td>
<td>-24</td>
<td>2</td>
<td>55.44</td>
</tr>
</tbody>
</table>
The next two tables illustrate the advantages gained by the fact that Algorithm 3 consists of two separate phases. Table 4 depicts the five terminals in which the difference between the number of extra-board drivers allocated by Algorithm 3 and the base allocation scheme is the most negative. Although these terminals rank highly among all terminals in terms of total outbound work, 55% to 75% of this work is on bid lanes. Thus, by first allocating bid drivers, Algorithm 3 determines that fewer extra-board drivers will be needed since bid drivers will be covering a large amount of the work.

<table>
<thead>
<tr>
<th>Terminal</th>
<th># Extra-board (Algorithm 3)</th>
<th># Extra-board (Base Allocation)</th>
<th>Difference</th>
<th>Outbound Work Rank</th>
<th>% Outbound Work on Reverse primary or Non-Bid Lane</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>102</td>
<td>48</td>
<td>54</td>
<td>5</td>
<td>92.09</td>
</tr>
<tr>
<td>2</td>
<td>65</td>
<td>33</td>
<td>32</td>
<td>15</td>
<td>84.00</td>
</tr>
<tr>
<td>3</td>
<td>59</td>
<td>24</td>
<td>35</td>
<td>25</td>
<td>95.58</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>16</td>
<td>34</td>
<td>21</td>
<td>68.56</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>15</td>
<td>20</td>
<td>24</td>
<td>82.48</td>
</tr>
</tbody>
</table>

Table 5: Terminals with most extra board drivers allocated using Algorithm 3

Alternatively, Table 5 shows that terminals with very little bid work may receive too few extra-board drivers under the base allocation scheme. Although these terminals have relatively little outbound work and thus receive only a small share of the drivers under the base allocation scheme, most of this outbound work is either on lanes without defined primary terminals, or in reverse primary direction (i.e., on lane \((A, B)\) where terminal \(B\) is primary). Since Algorithm 3 first assigns bid drivers, it determines a more accurate estimate of the remaining bid and non-bid work that remains for extra-board drivers, and can thus detect the need for extra-board drivers more accurately.

7.2 Experiment 2: Varying the Number of Drivers with Constant \(\rho\)

As mentioned in the introduction, one of the key tactical decisions an LTL carrier must make is the number of drivers to hire and where to locate them in the linehaul network. This is a difficult decision for many reasons. Freight flows are seasonal, so driver needs are not constant over time. Furthermore, assessing the amount of freight that can be handled for a given number of drivers and a given driver allocation is nontrivial. In this computational experiment, we analyze the impact of varying the number of drivers in the system while keeping \(\rho\) roughly constant. For each fleet size setting, we use the iterative
driver allocation scheme in Algorithm 3 to decide where to locate drivers for each of the weekly load instances W1, W2, and W3. The optimization formulations from Section 6 are again used to determine a final best allocation from these three weekly runs, and the driver scheduling technology is used to create the final schedule and the performance results. Table 6 presents these results for cases with 3,300, 3,000 and 2,500 total drivers.

<table>
<thead>
<tr>
<th>Driver Allocation</th>
<th>Date</th>
<th># Drivers Assigned</th>
<th>% Loads Dispatched</th>
<th>% Empty</th>
<th>% Foreign Beds</th>
<th>Average Drive Time</th>
<th>Average Duty Time</th>
<th>Tours per driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>2200 Bid &amp;</td>
<td>W1</td>
<td>2772</td>
<td>98.69</td>
<td>9.11</td>
<td>53.17</td>
<td>9.34</td>
<td>10.61</td>
<td>4.05</td>
</tr>
<tr>
<td>1100 Extra-</td>
<td>W2</td>
<td>2790</td>
<td>97.83</td>
<td>9.26</td>
<td>54.43</td>
<td>9.16</td>
<td>10.43</td>
<td>3.99</td>
</tr>
<tr>
<td>board</td>
<td>W3</td>
<td>2726</td>
<td>98.66</td>
<td>9.09</td>
<td>52.40</td>
<td>9.34</td>
<td>10.58</td>
<td>3.99</td>
</tr>
<tr>
<td>2000 Bid &amp;</td>
<td>W1</td>
<td>2621</td>
<td>98.52</td>
<td>9.15</td>
<td>55.23</td>
<td>9.37</td>
<td>10.62</td>
<td>4.28</td>
</tr>
<tr>
<td>1000 Extra-</td>
<td>W2</td>
<td>2581</td>
<td>97.05</td>
<td>9.35</td>
<td>56.53</td>
<td>9.21</td>
<td>10.46</td>
<td>4.20</td>
</tr>
<tr>
<td>board</td>
<td>W3</td>
<td>2589</td>
<td>98.66</td>
<td>8.82</td>
<td>52.40</td>
<td>9.39</td>
<td>10.64</td>
<td>4.12</td>
</tr>
<tr>
<td>1700 Bid &amp;</td>
<td>W1</td>
<td>2244</td>
<td>94.70</td>
<td>9.51</td>
<td>56.72</td>
<td>9.46</td>
<td>10.66</td>
<td>4.75</td>
</tr>
<tr>
<td>800 Extra-</td>
<td>W2</td>
<td>2209</td>
<td>93.95</td>
<td>9.91</td>
<td>58.75</td>
<td>9.33</td>
<td>10.54</td>
<td>4.62</td>
</tr>
<tr>
<td>board</td>
<td>W3</td>
<td>2249</td>
<td>95.41</td>
<td>9.21</td>
<td>55.88</td>
<td>9.46</td>
<td>10.67</td>
<td>4.61</td>
</tr>
</tbody>
</table>

Table 6: Varying the driver fleet size with a constant bid driver proportion of $\frac{2}{3}$

Observe that only slightly higher dispatch rates can be achieved with 3,300 compared to 3,000 drivers. However, with a driver fleet size of 2,500, dispatch rates reduce to about 94-95%. Such results suggest that the appropriate number of drivers for this system might be close to 3,000 drivers, located near-optimally using the iterative allocation scheme. We also observe that in all cases, even with 2,500 drivers, not all drivers are being used. This is primarily the result of using a two-phase driver allocation scheme, where bid drivers are allocated independently from extra-board drivers. When both types are included simultaneously in the final scheduling run it turns out that even fewer drivers are needed, since the extra-board drivers can perform bid work when no bid drivers are available. While such an observation may point to developing a methodology that allocates bid drivers and extra-board drivers simultaneously, this is actually very difficult to do due to the “poaching” bias that we described earlier in Section 6.

7.3 Experiment 3: Varying $\rho$ with nearly constant load coverage

LTL carriers realize that union rules add costs to their operations, but quantifying these additional costs is difficult. In this final experiment, we use the driver management
technology and the driver allocation scheme to develop initial insights regarding the impact of union rules on the number of drivers required in the system.

To do so, we first use the Phase II driver allocation scheme of Algorithm 3 to domicile 2,000 extra-board drivers for load set instances W1, W2, and W3. Then, we develop a final best allocation using the optimization models given in Section 6. Finally, we develop driver schedules for each of the load set instances using the driver scheduling technology to generate the final performance statistics for operating a system with 2,000 well-located extra-board drivers. Table 7 summarizes the results.

<table>
<thead>
<tr>
<th>Date</th>
<th>#Drivers Assigned</th>
<th>%Loads Dispatched</th>
<th>%Empty</th>
<th>%Foreign Beds</th>
<th>Average Drive Time</th>
<th>Average Duty Time</th>
<th>Tours per driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>2000</td>
<td>97.67</td>
<td>5.96</td>
<td>75.74</td>
<td>9.28</td>
<td>10.54</td>
<td>5.36</td>
</tr>
<tr>
<td>W2</td>
<td>1994</td>
<td>96.11</td>
<td>6.70</td>
<td>76.98</td>
<td>9.14</td>
<td>10.43</td>
<td>5.27</td>
</tr>
<tr>
<td>W3</td>
<td>1996</td>
<td>97.73</td>
<td>6.13</td>
<td>75.65</td>
<td>9.29</td>
<td>10.56</td>
<td>5.21</td>
</tr>
</tbody>
</table>

Table 7: Results with only extra-board drivers in the system

Next, we use Algorithm 3 to generate a driver allocation with 2,400 well-located drivers and \( \rho = \frac{1}{2} \) of them designated as bid drivers. We again develop driver schedules for each of the load set instances to generate the final performance measure results. Table 8 summarizes these results.

<table>
<thead>
<tr>
<th>Date</th>
<th>#Drivers Assigned</th>
<th>%Loads Dispatched</th>
<th>%Empty</th>
<th>%Foreign Beds</th>
<th>Average Drive Time</th>
<th>Average Duty Time</th>
<th>Tours per driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>2285</td>
<td>97.88</td>
<td>8.19</td>
<td>61.15</td>
<td>9.41</td>
<td>10.65</td>
<td>4.74</td>
</tr>
<tr>
<td>W2</td>
<td>2285</td>
<td>96.43</td>
<td>9.09</td>
<td>62.46</td>
<td>9.23</td>
<td>10.47</td>
<td>4.73</td>
</tr>
<tr>
<td>W3</td>
<td>2280</td>
<td>97.82</td>
<td>8.42</td>
<td>63.82</td>
<td>9.43</td>
<td>10.67</td>
<td>4.63</td>
</tr>
</tbody>
</table>

Table 8: Results with \( \rho = \frac{1}{2} \)

We observe that whereas a 97% dispatch rate required 3,000 well-located drivers when \( \rho = \frac{2}{3} \), only 2,400 are needed if \( \rho = \frac{1}{2} \). Similarly, when only extra-board drivers are in the system, 2,000 well-located drivers were sufficient to achieve a similar dispatch rate. It is evident that with service constant, an increase in \( \rho \) requires an increase in the driver pool. In fact, the required increase can be steep: when increasing \( \rho \) from 0 to \( \frac{2}{3} \), 50% more drivers are required to achieve the same level of service. Even a much smaller increase in \( \rho \), from \( \frac{1}{2} \) to \( \frac{2}{3} \), requires 20% more drivers. Observe too that, not surprisingly, there is a significant increase
(20\%) in the number of foreign beds when there are only extra-board drivers (compared to $\rho = \frac{2}{3}$). While such an increase in foreign bed cost may also be significant for carriers, this cost increase is likely to be small relative to the savings generated by maintaining a much smaller driver fleet.

8 Future Work

The driver fleet sizing and allocation technology and the driver scheduling and load dispatching technology proposed in this research can be used to analyze various other tactical decisions, such as the effect of changing primary terminals on lanes and the effect of changing load plans. We plan to conduct such analyses for a major LTL carrier in the near future.

Acknowledgements

The work described in this paper was supported, in part, by a grant from the Sloan Trucking Industry Program.

References


