Improved Load Plan Design
Through Integer Programming Based Local Search

Alan Erera, Michael Hewitt, Martin Savelsbergh, Yang Zhang
The Supply Chain and Logistics Institute
School of Industrial and Systems Engineering
Georgia Institute of Technology
765 Ferst Drive, NW
Atlanta, GA 30332-0205

Abstract

We present advances both in modeling and algorithm design for a service network design problem faced by less-than-truckload (LTL) freight transportation carriers. The developed models more accurately capture key operations of today’s carriers: decisions for loaded and empty trailer movements are considered simultaneously, and a time discretization is used that can appropriately model the timing of freight consolidation opportunities. In addition to providing decision support for traditional service network plans used by LTL carriers, the models also enable the development of plans that allow more flexibility, such as allowing certain freight routes to vary by weekday. Given the additional detail within the proposed models, very large problem instances result when they are applied to large-scale LTL networks. We therefore propose a local search solution heuristic that searches a large neighborhood each iteration using an integer program. Computational experiments using data from a large U.S. carrier demonstrate that the proposed modeling and solution approach has the potential to generate significant cost savings.

1 Introduction

National less-than-truckload (LTL) carriers run high-volume freight transportation operations, often spending millions of dollars in transportation and handling costs each week. A LTL motor carrier transports shipments that typically occupy only 5-10% of trailer capacity. Hence, transporting each customer shipment directly from origin to destination is not economically viable. As a result, LTL carriers collect and consolidate freight from multiple shippers to increase trailer utilization, referred to as the “load factor,” and route freight through a network of consolidation terminals between origin and destination. The savings generated by increasing trailer load factors through consolidation is partially offset by other costs; transferring freight between trailers generates a handling cost, and terminal-to-terminal routing increases the total time and distance each shipment requires to reach its destination.

This paper focuses on methods for planning how freight should be routed from origin to destination through the terminal network, specifying where along the way it is transferred.
from one trailer to another. In the LTL industry, this service network design problem is known as constructing a *load plan*. Customer service and pricing pressures have made effective load plans more critical than ever to a carrier’s success. National LTL carriers now compete with both so-called super-regional carriers (resulting from mergers of regional LTL carriers), and also LTL service offerings from traditional package express companies like UPS and FedEx; this has led to pressure on pricing, as well as increased competition on customer service including transit time and on-time reliability.

LTL shipments are quoted a service standard from origin to destination in business days. Historically these standards were long enough (often 5 or more business days) that service only loosely constrained freight routing decisions. Today, service standards of 1, 2, and 3 days are much more common (Figure 1 presents a freight profile by service standard for a national carrier), and these tighter standards must be enforced when planning shipment paths. Shorter service standards reduce opportunities for consolidation (since consolidation introduces handling time and circuitry time penalties), especially if rigid operating rules are also enforced (such as a shipment transferred at most one breakbulk per day). Modern freight routes are more complex, and a shipment may visit multiple breakbulks in a single operating day. As a result, carriers need methods for designing load plans that accurately model how short service standards constrain shipment paths and the consolidation opportunities that still exist.

**Figure 1: Number of Shipments by Service Standard for a Major US LTL Carrier**

LTL load plans are typically constructed using a number of self-imposed simplifying rules that are later relaxed during actual operations. For example, a traditional load plan assumes that the same freight routing decisions are executed every day. On a given operating day, terminal managers then often override the plan in an attempt to reduce linehaul costs given daily freight volume fluctuations; a simple example is to build a trailer that skips a planned next handling terminal. By relaxing some of these simplifying rules when planning, it may be possible to build improved load plans that generate cost savings and require less manual replanning. For example, one such relaxation would be to plan for predictable daily volume fluctuations by allowing for different freight routing decisions on different weekdays.

In this paper, we discuss the design and implementation of load planning technology that is more accurate than existing solutions, because it explicitly and accurately models
time, and more flexible than existing solutions, because it accommodates relaxations to
the simplifying rules of traditional load plans. In our approach, we model freight routing
decisions on a time-space network. To accurately capture consolidation opportunities given
short service standards, we map a single terminal on a single day to multiple nodes. The
resulting network is very large with up to 6,000 nodes and 500,000 arcs. To generate plans
that allow different freight routing decisions by weekday, we model freight originating at a
terminal on a given day and destined for another terminal on another day as a commodity,
resulting in instances with more than 60,000 commodities. To effectively solve such large
instances, we develop an approach that combines exact optimization with heuristic search.
In particular, we implement a neighborhood search approach in which neighborhoods are
defined by the feasible solutions to an integer program, and searching for an improving
solution in the neighborhood is conducted by solving the integer program.

Our load planning technology differs in one other important aspect from existing solution
approaches. Due to the difficult nature of load plan design, existing approaches simplify
the problem by separating the freight routing decisions from empty trailer repositioning
decisions (repositioning is necessary due to freight demand imbalance). By sequentially
deciding first how to route freight, and then how to move trailer equipment to eliminate
imbalance, traditional approaches lead to freight routing decisions that ignore natural
“backhaul lanes” where empty trailers are likely to be moving. Carriers correctly understand
that it is desirable to route freight when possible along such lanes, and often must modify
traditional load planning outputs to do so. Our load planning technology makes freight
routing and empty repositioning decisions simultaneously.

The research presented in this paper makes contributions both in the context of load
plan design and heuristic search. Specifically, we present methodology that

- designs traditional load plans that offer significant cost-savings; approximately $200,000
  weekly for a national carrier,

- is the first to model time using a discretization appropriate for the tight service stan-
dards a carrier must offer to remain competitive,

- is the first to integrate empty trailer repositioning trailers within freight routing de-
cisions,

- designs day-differentiated load plans that account for predictable daily freight volume
  fluctuations that yield even greater cost-savings; approximately $350,000 weekly for
  a national carrier, and

- illustrates a successful application of using exact optimization within heuristic search
  for a large-scale optimization problem.

The remainder of the paper is organized as follows. Section 2 provides background in-
formation on the LTL industry and load plan design in particular, and Section 3 presents a
brief review of relevant literature. Section 4 details how our model of LTL operations rep-
resents a significant improvement over existing models in the literature. Section 5 outlines
how we model freight routing, and Section 6 presents an integer programming formulation
of the load plan design problem and some of its variants. Section 7 details the solution
heuristic, including integer programming techniques to improve its performance. Finally,
Section 8 presents the results of an extensive computational study conducted using data
from a national LTL carrier.
LTL networks include two types of terminals: satellite or end-of-line (EOL) terminals that serve only as origin or destination terminals for freight, and breakbulk (BB) terminals that additionally may serve as transfer points for shipments. The set of EOL and BB terminals is known as the linehaul network, and typically has a hub-and-spoke structure with an EOL loading to at most two or three BB terminals. A separate operation from each terminal, known often as the city operation, is responsible for pickup and delivery at customer locations.

Shipments are consolidated at two levels of the linehaul network. During the day, city operation drivers deliver and collect shipments from customers within a small geographic area. Collected shipments are then brought to the terminal serving as the sorting center and loading facility for outbound and inbound freight for the geographic area. This is the first level of consolidation.

When these origin terminals do not have enough freight to build a full trailer to a destination terminal, they instead route freight to an intermediate BB terminal; this is the second level of consolidation. For example, outbound freight from an EOL is typically loaded first to a BB terminal where it is consolidated with freight from other EOLs and BBs into outbound trailers. The outbound trailers from the BB may be built direct to certain destinations, or may be built to other downstream BBs for another round of consolidation. The consolidation operation at BBs involves unloading and reloading of trailers, and thus involves both time and cost (commonly referred to as handling time and cost).

An originating shipment is typically delivered by the city operation to the origin terminal by 7 or 8 pm, and must be moved to the destination terminal by 8 or 9 am on the day of delivery specified by the service standard (all times local). For an example drawn from a national LTL carrier, a shipment originating in Atlanta, GA on Monday with a destination of Cincinnati, OH and a service standard of 1 business day will be available at the Atlanta terminal on Monday by 7 pm and must be moved to the Cincinnati terminal by no later than 8 am Tuesday morning.

An important concept in load planning is that of a direct. A direct specifies where handling of freight occurs, and is a fundamental building block for a load plan. If a shipment uses the direct \(A \rightarrow B\), it is loaded into a trailer at \(A\) which is not opened (and hence the freight not handled) again until it reaches \(B\). Note that a direct trailer may be moved by multiple drivers (and possibly, multiple modes) en route from \(A\) to \(B\). For example, a trailer on the direct Atlanta, GA \(\rightarrow\) Seattle, WA may actually travel Atlanta, GA \(\rightarrow\) Nashville, TN \(\rightarrow\) Kansas City, MO \(\rightarrow\) Spokane, WA \(\rightarrow\) Seattle, WA; in this example, the trailer is moved by rail from Kansas City to Spokane.

Consider a path from origin to destination consisting of a sequence of directs. Then, a load plan specifies a path for each shipment and thus prescribes how all freight should be routed through the linehaul network. A typical traditional load plan will specify a unique path for each (origin terminal, destination terminal) combination. Furthermore, such plans also have a special structure where the set of all paths terminating at a specific destination terminal \(d\) form a directed in-tree on the network of potential directs. Thus, all shipments that pass through intermediate terminal \(i\) on a path to \(d\) use the outbound direct \((i, j)\). For example, the load plan may give the following instruction: “all freight in Jackson, TN destined for Atlanta, GA loads direct to Nashville, TN.”

Of course, freight cannot move without a trailer to carry it, hence an LTL carrier usually needs to move trailers empty to correct trailer imbalances caused by underlying
freight volume directional imbalances. Empty trailer repositioning movements incur costs. Furthermore, load plans are typically adjusted in practice to attempt to route freight when possible along natural backhaul lanes. While such adjustments may reduce the load factor of loaded trailers (the fraction of available space in trailers that is occupied by shipments), they can reduce total system costs.

The load plan design problem is to determine how freight is routed through the network of potential directs. Load plans are designed to minimize total linehaul costs which are broken down into two categories:

1. Transportation costs associated with moving loaded and empty trailers
2. Handling costs associated with transferring freight between trailers at a terminal.

A traditional load plan specifies freight paths that do not vary during the week. However, the freight volumes seen by carriers often exhibit predictable daily variation, either in the freight volume or the direction of the freight. For example, a carrier may serve a customer whose distribution center is in Lexington, KY. On Monday, outbound freight for that customer may be destined for terminals in the Southeast, while on Tuesday it may be destined for terminals in the Northwest. Additionally, since service standards to customers are quoted in business days, shipments that originate near the end of the business week often can be routed over many more time-feasible paths because of the flexibility offered by the weekend.

Hence, we introduce in this paper the concept of a day-differentiated load plan. A day-differentiated plan will specify a unique path for all freight from a specific origin to a specific destination that originated on the same weekday. To do so, we extend the in-tree structure of a traditional load plan such that the outbound direct \((i, j)\) for freight at \(i\) destined for \(d\) also depends on the day of the week. For example, a day-differentiated load plan could specify: “on Monday all freight in Jackson, TN destined for Atlanta, GA loads direct to Nashville, TN, but on all other days all freight in Jackson, TN destined for Atlanta, GA loads direct to Birmingham, AL.”

3 Literature Review

Load plan design is similar to the classic network design problem, which has been extensively studied. A common theme in much of the work on network design is that exact optimization is often impractical for realistically-sized instances, and therefore heuristic solution techniques are needed in practice. Metaheuristics have been developed that find good primal solutions to instances of the capacitated fixed charge network design problem. A tabu search algorithm using pivot-like moves in the space of path-flow variables is proposed in Crainic et al. (2000). The scheme is then parallelized in Crainic and Gendreau (2002). A tabu search algorithm using cycles that allow the re-routing of multiple commodities is presented in Ghamlouch et al. (2003). This cycle-based neighborhood is incorporated within a path-relinking algorithm in Ghamlouch et al. (2004). Each of these heuristics allows the flow for a specific commodity (defined by an \((\text{origin}, \text{destination})\) pair) to be split among multiple paths. Furthermore, none considers underlying equipment moves (including empty repositioning movements).

Load plan design can be seen as a special case of service network design; this problem class has also received a great deal of attention (see Crainic (2000) or Wieberneit (2008) for a review of research in this area). The need to consider equipment management decisions
in service network design problems is recognized in Pederson et al. (2009), which presents both a model and a metaheuristic for the problem. However, the instance sizes considered are significantly smaller than those typical for load planning for a large LTL carrier, and it is not clear how effective the proposed solution approach would be if adapted to the load plan design problem.

Relatively little research has focused specifically on the load plan design problem for LTL carriers. Early research focused on problem models developed using flat (static) networks that do not explicitly capture service standard constraints or the timing of consolidation opportunities. A local improvement heuristic for such a model is presented in Powell (1986); related work includes Powell and Sheffi (1983), Powell and Sheffi (1989), and Powell and Koskosidis (1992). Recognizing the limitations of the static network models, Powell and Farvolden (1994) present a dynamic model that can more accurately model consolidation timing. The paper presents an alternative heuristic that relies on determining service network subgradients by solving large-scale multi-commodity network flow problems. However, this approach allows origin-destination shipments to split onto multiple paths and does not model empty equipment balancing decisions. Most recently, Jarrah et al. (2009) present a model that more accurately captures consolidation timing using a model that is structurally similar to ours. However, time is modeled much more coarsely with a single node per day per terminal, and loaded and empty routing decisions are considered separately and sequentially. To the best of our knowledge, no work has allowed freight routing decisions to vary by day.

The neighborhood search solution heuristic that we develop in this paper solves carefully chosen integer programs to improve an existing solution. Thus, like very-large-scale neighborhood search heuristics (VLSN) it uses exponential-sized neighborhoods (Ahuja et al. (2002)). However, in contrast to VLSN, no polynomial-time algorithm exists for searching these neighborhoods. An similar approach is developed for the classic network design problem in Hewitt et al. (2009). Other examples of combining exact and heuristic search techniques can be found in De Franceschi et al. (2006), Schmid et al. (2008), Archetti et al. (2008), and Savelsbergh and Song (2008).

4 Enhanced Load Planning

To improve model-based approaches for LTL load planning, we focus on three key modeling features that differ significantly from what is found in existing load plan research: we model time explicitly at an appropriate level of detail, we consider empty and loaded trailer routing decisions simultaneously, and we enable creation of plans that allow additional routing flexibility by relaxing one of the primary assumptions of traditional load plans. Regarding the third key feature, the primary relaxation we consider is to allow generation of day-differentiated load plans.

- **Modeling Time:** Approximately 80% of the freight volume for the carrier that we studied has a service standard of either 1 or 2 business days. The need to build consolidated shipments in conjunction with such short service standards necessitates a model that uses a time discretization with a finer level of detail than typically considered. According to our carrier’s existing load plan, nearly 20% of overnight (one-day service standard) shipments were transferred from one trailer to another at least once. For example, consider the following existing path for freight originating in Evansville, IN at 7 pm destined for Atlanta, GA at 8 am the next day:
– Travel for 3 hours from Evansville, IN to Nashville, TN.
– Handling for 2 hours in Nashville, TN.
– Travel for 5 hours from Nashville, TN to Atlanta, GA.

In addition, consider the path for freight originating in Louisville, KY at 8 pm also destined overnight to Atlanta:

– Travel for 2.5 hours from Louisville, KY to Nashville, TN.
– Handling for 2 hours in Nashville, TN.
– Travel for 5 hours from Nashville, TN to Atlanta, GA.

Note that a simple time-space model that discretizes time with a single node for each terminal each day and arcs that only move forward in time would conclude that each of these paths is infeasible, and thus a potential real-world consolidation opportunity would be lost. However, by modeling time at a finer level of detail, it is possible to determine dispatch times for each of these paths such that the freight may be consolidated at Nashville into a common trailer (or trailers) outbound to Atlanta.

Freight is only handled and consolidated at EOL terminals when it initially enters the linehaul system, and is deconsolidated at those terminals only when it exits linehaul for delivery. In our models, we assume that all freight enters the linehaul network (sorted and ready for dispatch) at 7 pm and must arrive at the destination terminal by 8 am. Therefore, we only include nodes for these two time epochs for EOLs in our model. Freight is handled at consolidated at BB terminals primarily during the overnight hours, therefore we divide a day into the windows \{1 to 3 am, 3 to 5 am, 5 to 8 am, 8 am to 2 pm, 2 to 7 pm, 7 to 9 pm, 9 to 11 pm, 11 to 1 am\}, with nodes specified at each time breakpoint.

**Empty Integration:** Since solving realistically-sized instances of models for the load plan design problem is beyond the capability of even the most sophisticated integer programming solvers available today, heuristic solution approaches are used. A common feature in all approaches is to first route the freight (hence determining requirements for loaded trailers) assuming trailers are always available when needed. Then, a linear program is solved to resolve trailer imbalances (specifically, a transportation problem).

Such sequential heuristics, however, may result in freight routing decisions that carriers would never implement. For example, consider the small static network presented in Figure 2 where the numbers above each arc represent the dispatch cost of a trailer. Suppose that we have the following freight, where volume is measured in fractional trailerloads:

– One originating at \(C\) and destined for \(A\),
– One originating at \(C\) and destined for \(B\),
– One half originating at \(A\) and destined for \(C\), and
– One half originating at \(B\) and destined for \(C\).

Given full trailerloads traveling \(C \to A\) and \(C \to B\), there is no need to consolidate that freight. If we ignore empty repositioning and only minimize the costs of moving
loaded trailers to determine freight routing, we obtain the solution in Figure 3, where
the numbers above arcs represent the number of trailers dispatched, and the numbers
beside each node represent the balance of trailers at that node. The loaded trailer
cost is 11.5, but resolving trailer imbalances at B and C costs 2.5 for a total linehaul
cost of 14. If on the other hand, we simultaneously optimize freight routing and trailer
repositioning, we obtain the solution in Figure 4, which has a higher loaded trailer
cost (13), but requires no repositioning of empty trailers and hence has a lower total
linehaul cost. In practice, a carrier would recognize that A → C and B → C are
backhaul lanes and hence would be wary of solutions like those in Figure 3 that move
freight away from those lanes.

- **Day-Differentiation**: The first two proposed improvements address the accuracy
of how traditional LTL carrier operations are modeled. Alternatively, enabling day
differentiation of load plans allows us to evaluate the potential cost savings of relaxing
the rules that govern how a carrier routes freight. Consider again our example network
in Figure 2, but now assume the following freight volumes:

  - **On Monday**:
    * One originating at A and destined for C,
    * One half originating at B and destined for C.
  - **On Tuesday, Wednesday, Thursday, Friday**:
    * One originating at A and destined for C,
Figure 4: Freight routing decisions made in conjunction with empty repositioning decisions yield higher loaded costs, but lower total costs.

Figure 5: Load plan on Monday does not consolidate freight.

* One originating at $B$ and destined for $C$.

Given a full trailer traveling $A \rightarrow C$ on Monday, there is no need for consolidation, and the load plan depicted in Figure 5 (the numbers above each arc are the fractional trailerloads assigned to each arc) is optimal. For the other days of the week, however, it is less costly to route the $A \rightarrow C$ freight through $B$ and consolidate. Thus, it is better to execute the load plan depicted in Figure 6 on Tuesday through Friday. In fact, most carriers already route freight in a day-differentiated manner, deviating from the prescribed load plan when opportunities arise. On Monday, the terminal manager at $A$ will recognize that there is enough freight going $A \rightarrow C$ to build a

Figure 6: Load plan on Tuesday-Friday consolidates freight through BB terminal $b$. 
full trailer and send it on the opportunistic direct $A \rightarrow C$ even though the load plan may specify that the freight should next be routed to $B$. By enabling the creation of day-differentiated plans, we create the ability to capture some of the daily freight routing decisions made by terminal managers, and potentially find improvements. In practice, terminal managers typically base their decisions about when to introduce an opportunistic direct solely on information available locally at their terminal. Thus, the decision to use an opportunistic direct could actually lead to increased downstream costs. By incorporating day-differentiation into our model, we can capture the global impact of these decisions.

5 Modeling Freight Routing

To design a load plan, we must determine the directs used to route freight as well as when and where to hold freight to improve consolidation and reduce system costs. To make these decisions, we model terminals and potential directs on a time-space network. Let $LN = (U, L)$ denote the carrier’s linehaul network, where $U$ is the set of terminals in the carrier’s network and $L$ is the set of potential directs connecting terminals. Associated with each direct $l = (u_1, u_2) \in L$ is a transit time that reflects how long it takes a carrier to route a trailer from terminal $u_1$ to terminal $u_2$. For a given time discretization of a planning horizon, we define the time-space linehaul network $TS - LN = (N, A)$, where $N$ denotes the set of nodes and $A$ denotes the set of arcs. Each node $n = (u, t), u \in U, t \in T$ represents a terminal at a particular point in time. Each arc $a = ((u_1, t_1), (u_2, t_2))$ with $u_1$ and $u_2 \in U$ and $u_1 \neq u_2$ represents a potential dispatch from $u_1$ at time $t_1$ on direct $(u_1, u_2)$ arriving in $u_2$ no later than time $t_2$. We create such arcs for each direct $l = (u_1, u_2) \in L$ and each timed copy $((u_1, t_1))$ of the origin node $u_1$. The destination node $(u_2, t_2)$ is then chosen to be the earliest timed copy of the node $u_2$ such that $t_2 - t_1$ is no less than the transit time of the underlying direct $l$. We also create arcs $a = ((u_1, t_1), (u_1, t_2))$ to connect subsequent timed copies of each node $u_1$. These allow us to model holding a trailer or shipment at terminal $u_1$.

Given networks $LN$ and $TS - LN$, let $\delta^+(u) \subseteq L$ denote the set of potential outbound directs from terminal $u \in U$, $l(a)$ denote the direct $l \in L$ corresponding to the arc $a \in A$, $\delta^+(n) \subseteq A$ denote the set of outbound arcs from node $n \in N$, $\delta^-(n) \subseteq A$ denote the set of inbound arcs to node $n \in N$, and $c_a$ denote the per-trailer travel cost along arc $a \in A$.

We model freight which enters the linehaul network at terminal $u_1$ on day $d_1$ as entering the time-space network at node $n_1 = (u_1, t_1)$ where $t_1 = d_1 \@ 7$ pm. Freight which must reach terminal $u_2$ by day $d_2$ is given the destination node $n_2 = (u_2, t_2)$ where $t_2 = d_2 \@ 8$ am. All freight shipments with a common $(n_1, n_2)$ pair are considered a single commodity. Carriers typically quote a single service standard for freight originating at terminal $u_1$ and destined for terminal $u_2$. Although we use this assumption, accommodating multiple classes of service (say “regular” and “express”) between $u_1$ and $u_2$ is also enabled by this modeling framework.

Let $K$ denote the set of commodities. For each commodity $k \in K$, let $o(k)$ denote the origin terminal, $d(k)$ denote the destination terminal, $on(k) \in N$ denote the origin node, $dn(k) \in N$ denote the destination node, $w_k$ denote the weight in pounds, and $q_k$ denote its size measured in fractional trailers (note, $q_k$ need not be less than one). Let $K(d) \subseteq K, d \in U$ denote the set of commodities with destination terminal $d$ and let $K(o, d) \subseteq K, o \in U, d \in U$ denote the set of commodities with origin terminal $o$ and destination terminal $d$.  


Our approach will use a path-based optimization model on the time-space network. Therefore, let \( P(k) \) be a set of possible freight paths for commodity \( k \in K \), where a freight path \( p \) is a sequence of arcs, i.e., \( p = (a_1, \ldots, a_n) \). Each path \( p = (a_1, \ldots, a_n) \in P(k) \) is constructed such that \( a_1 \in \delta^+(on(k)) \) and \( a_n \in \delta^-(dn(k)) \). How commodity \( k \) is routed then simply becomes a question of choosing a path \( p \in P(k) \). By using a path-based model, certain constraints can easily be enforced, e.g., freight is handled at most two times. Associated with a path \( p = (a_1, \ldots, a_n) \) is an underlying path \( p' \) of directs \( p = (l(a_1), \ldots, l(a_n)) \). For convenience, we sometimes represent a path \( p \) of directs as a sequence of terminals, e.g., we may refer to the path of directs \( ((u_1, u_2), (u_2, u_3)) \) as \((u_1, u_2, u_3)\). Note that given a path \( p \), we can calculate its total handling cost \( h_p \) per pound by summing the costs for the intermediate terminals visited.

To construct a set of paths \( P(k) \) for commodity \( k \), we begin by computing up to \( m \) minimum cost paths in the linehaul network \( LN \) with respect to total travel and handling cost (for some given value of \( m \)). Since we ignore time in this computation, we next determine which of these paths on \( LN \) can be mapped to service feasible paths on \( TS-LN \). To do so, we first determine the minimum execution duration of each path by mapping the sequence of directs to a feasible set of arcs in \( A \), determined by the transit time of the directs and required handling time between directs at intermediate terminals. For commodities that represent 1-day (overnight) services, 30 minutes of handling time is assumed while for all other service standards, we assume two hours.

To insert handling time into a path, we assume that the slack created by mapping transit times to our time discretization is available for handling. Consider a path of directs \((u_1, u_2, u_3)\) for a commodity with 2 day service originating on Monday at \( u_1 \) and destined for \( u_3 \). Since initial dispatches occur at 7 pm, if the transit time from \( u_1 \) to \( u_2 \) is 11 hours then the arc in \( A \) arrives at \( u_2 \) on Tuesday morning at 6 am. Since the next timed copy of \( u_2 \) in \( N \) occurs at 8 am, we assume handling occurs between 6 am and 8 am, and this commodity is ready for dispatch to \( u_3 \) at 8 am on arc \( a = ((u_2,Tuesday @ 8am),(u_3,t_3)) \). If, however, the transit time from \( u_1 \) to \( u_2 \) is 12 hours, then only one hour of handling time is available prior to 8 am. Therefore, we next determine whether there is slack available in the mapping of \((u_2, u_3)\) to \( TS-LN \). For exposition, suppose that \( u_3 \) is a BB terminal, and therefore its next timed copy is Tuesday at 2 pm. Then, if the transit time from \( u_2 \) to \( u_3 \) is \( \leq 5 \) hours, we assume that the remaining hour of handling occurs before the dispatch and map the direct \((u_2, u_3)\) to the arc \( a = ((u_2,Tuesday @ 8am),(u_3,t_3)) \).

We recognize that given this mapping methodology, for different paths for different commodities, the arc \((u_2,Tuesday @ 8am),(u_3,t_3))\) may assume a dispatch at slightly different points in time. Since we will assume that any freight traveling on the same arc \( a \in A \) can be loaded into the same trailers, these assumptions might seem to overestimate consolidation opportunities. However, it should be noted that the handling time estimates are not hard lower bounds, and carriers can prioritize handling to reduce these times when needed.

For each path of directs that is service feasible, we include not only the minimum duration path \( p \) into \( P(k) \), but potentially also other versions that add holding arcs of the form \((u_1,t_1),(u_1,t_2)\) if they are also feasible. Adding such timed copies models the ability to hold freight at intermediate terminals to improve the plan. We construct a limited set of such paths by only holding freight until specific events occur. First, we allow freight to be held at a terminal until the time that new freight originates at that terminal; thus, freight arriving at a breakbulk during the day can be consolidated with that evening’s originating outbound freight. Second, we allow freight to be held at a terminal until its cut time,
i.e., the latest time at which the freight can be dispatched and still arrive on time to its destination. In this way, freight destined for common destinations may be consolidated.

Consider the two examples in Figure 7. In the first, commodity \( k \) originates at and is destined for an end-of-line terminal. The network in the figure depicts the locations and time points where we model consolidation opportunities for this commodity, and all possible paths between the origin node and destination node of the commodity in the network would be added to \( P(k) \). In the second example, commodity \( k \) originates at and is destined for a breakbulk terminal, and thus there are many more opportunities for this commodity to be consolidated with other freight.

![Figure 7: Holding Freight for Consolidation](image)

We have introduced two networks, \( LN \) and \( TS - LN \), each of which plays a role in load plan design. Recall that a traditional load plan specifies the unique direct a shipment should take given its current terminal location and its ultimate terminal destination. Choosing the unique outbound direct for freight at terminal \( u_1 \) and destined for terminal \( d \) (regardless of its origin or service standard) corresponds to choosing a single arc in \( LN \) from \( \delta^+(u_1) \) for freight destined to node \( d \in U \). Hence, the structure of a traditional load plan requires that the directs chosen for freight destined for terminal \( d \) must form a directed in-tree on \( LN \) rooted at the node \( d \) (as depicted on a small example in Figure 8). Note this tree structure also implies that freight at terminal \( u_1 \) and destined for terminal \( d \) follows a unique path in \( LN \).

In our path-based approach, we choose for each commodity \( k \in K \) on \( TS - LN \) a path of arcs, where a component arc \( a \) is a timed copy of a direct. Therefore, when constructing load plans, we must ensure that the set of paths chosen for all commodities are such that there is appropriate consistency of the paths selected for commodities in \( K(d) \) for each given destination \( d \). For traditional load plans, we have chosen only to ensure that the outbound direct \( l \in L \) is the same for all such commodities, but the timing of the movements may change. Continuing the example from the previous paragraph, Figure 9 illustrates a simplified time-space network where each direct requires one time period for transit and handling. Furthermore, the depicted network only includes directs consistent with the load plan for commodities \( k \) with \( \delta n(k) = (d, t_i) \) given in Figure 8. Suppose freight is destined for \( d \) originating at \( u_2 \) at both \( t_0 \) and \( t_1 \), with a service standard of two periods. Thus, in our model there are two commodities, \( k_0 = ((u_2, t_0), (d, t_2)) \) and \( k_1 = ((u_2, t_1), (d, t_3)) \). For \( k_0 \), the model might choose the path \( p_0 = (a_0, a_1) \), which holds the freight at the origin for one period, and the path \( p_1 = (a_1, a_2) \) for \( k_1 \) which does not hold the freight. Such decisions consolidate the freight on arc \( a_1 \).
Figure 8: LN for a four terminal network, and load plan as a directed in-tree into \( d \)

Figure 9: Time-space network depicting routing choices into \( d \) given a load plan.
6 Load Plan Design Integer Program

We next present an integer programming formulation of the traditional load plan design problem (TLD-IP). TLD-IP has three sets of decision variables. First, $x$ variables indicate whether commodity $k$ uses path $p$, i.e., $x^k_p \in \{0, 1\}$ $\forall k \in K$, $\forall p \in P(k)$. Second, $y$ variables enforce consistency between paths for commodities heading to common destinations by indicating whether direct $l \in \delta^+(u)$ is chosen for all commodities destined for terminal $d$ routed through terminal $u$, i.e., $y^d_l \in \{0, 1\}$ $\forall d \in U$, $\forall l \in \delta^+(u)$, $u \in U$. Finally, $\tau$ variables count the number of trailers (empty or loaded) that move on arc $a$, i.e., $\tau_a \in \mathbb{Z}^+$ $\forall a \in A$.

The formulation is to then minimize

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

subject to

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

$$\sum_{l \in \delta^+(u)} y^d_l \leq 1 \quad \forall u \in U, \forall d \in U \quad (2)$$

$$\sum_{p \in P(k) : a \in p} x^k_p \leq y^d_l(d) \quad \forall k \in K, \forall a \in A \quad (3)$$

$$\sum_{k \in K} \sum_{p \in P(k) : a \in p} q_k x^k_p \leq \tau_a \quad \forall a \in A \quad (4)$$

$$\sum_{a \in \delta^+(v)} \tau_a - \sum_{a \in \delta^-(v)} \tau_a = 0 \quad \forall v \in V \quad (5)$$

The objective is to minimize total transportation and handling costs. Constraints (1) ensure that a path is chosen for each commodity. Constraints (2) ensure that a single outbound direct is selected for each terminal $u$ and destined for terminal $d$. Constraints (3) ensure that a path can only be chosen for commodity $k$ when all of its component directs are chosen. Constraints (4) ensure that there are enough trailers moved along an arc to carry the freight assigned to the arc via the paths chosen. Finally, constraints (5) ensure flow balance of trailers at every node in the time-space network, and thus ensure proper repositioning of trailers.

For reasonably sized instances, there will be prohibitively many constraints (3). However, a valid formulation with fewer constraints, but a weaker linear programming relaxation, can be obtained. Let $P(k)$ denote a set of paths of directs for commodity $k \in K$, and for a destination $d \in U$, let $K_l(d) \subseteq K$ be the set of commodities $k \in K(d)$ such that $\exists p \in P(k)$ containing the direct $l$. Then, by aggregating over the set $K_l(d(k))$, the following constraints are equivalent to (3):

$$\sum_{k \in K_l(d(k))} \sum_{p \in P(k) : a \in p} x^k_p \leq |K_l(d(k))| y^d_l(d(k)) \quad \forall a \in A.$$
6.1 Variations on Traditional Load Plan Design

Without requiring the directs in a load plan to form an in-tree, the problem faced by carriers is a special case of network design. In fact, constraints (1) and (4) represent a variable upper-bound network design problem where the upper bounds represent trailer capacities. The in-tree requirement is enforced to simplify terminal operations, since it allows a terminal worker to only examine the destination of a shipment to determine the appropriate outbound trailer for loading. However, advances in information technology and the introduction of handheld scanners into terminal operations largely render the in-tree requirement unnecessary for LTL operations.

To understand the cost savings possible by changing the traditional load plan structure, we consider three relaxations: the day-differentiated load plan, the same-path load plan, and the unrestricted load plan.

- **Day-Differentiated Load Plan Design:** Whereas a traditional load plan ensures that a single outbound direct is chosen for freight at terminal \( u \) destined for terminal \( d \) for the entire week, a day-differentiated load plan only ensures a single outbound direct each day. Day-differentiation can be accommodated in TLD-IP by redefining the variables that indicate what directs are chosen and by slightly modifying the constraints (2) and (3). We will refer to the formulation for the day-differentiated load plan design problem as DDLD-IP. Let the set of days in the planning horizon be denoted by \( DAYS \), and let \( m(a) \) denote the day \( m \in DAYS \) corresponding to the tail node of arc \( a \in A \).

We introduce variables \( y_{l}^{m,d} \) to indicate whether direct \( l \in \delta^{+}(u) \) is chosen for freight destined for terminal \( d \) at terminal \( u \) on day \( m \), i.e.,

\[
y_{l}^{m,d} \in \{0, 1\} \quad \forall d \in U, \quad \forall m \in DAYS, \quad \forall l \in \delta^{+}(u), u \in U.
\]

The day-differentiated analog of the TLD-IP constraints (2) ensures that a single outbound direct is chosen for freight ultimately destined for terminal \( d \) at terminal \( u \) on day \( m \), i.e.,

\[
\sum_{l \in \delta^{+}(u)} y_{l}^{m,d} \leq 1 \quad \forall u \in U, \quad \forall m \in DAYS, \quad \forall d \in U \tag{6}
\]

The day-differentiated analog of the TLD-IP constraints (3) ensures that a path can only be chosen for commodity \( k \) when all the underlying day-indexed directs are chosen, i.e.,

\[
\sum_{p \in P(k):a \in p} x_{p}^{k} \leq y_{l}^{m(a),d(k)} \quad \forall k \in K, \forall a \in A \tag{7}
\]

The same comments regarding aggregating the coupling constraints (3) of TLD-IP applies to the constraints (7).

- **Same-Path Load Plan Design:** The same-path load plan drops the in-tree requirement completely but keeps the restriction that the freight between two terminals follows the same sequence of directs every day. We can remove the in-tree restriction from the TLD-IP by removing the variables \( y_{l}^{d} \) and the constraints (2) and (3).
Given that we require a unique path between terminals in the linehaul network $LN$, all commodities with the same origin and destination must use the same sequence of directs. Let $P(o, d)$ denote the set of paths of directs between terminals $o, d \in U$. Furthermore, let a set of paths of arcs for a path of directs $p$ and commodity $k$ be denoted by $P(p, k)$. Then, the following same-path inequalities need to be satisfied:

$$\sum_{p \in P(p, k_1)} x^k_p = \sum_{p \in P(p, k_2)} x^k_p \quad \forall o, d \in U, \; \forall k_1, k_2 \in K(o, d), \; \forall p \in P(o, d) \quad (8)$$

We will refer to the formulation for the same path load plan design problem as SPLD-IP.

- **Unrestricted Load Plan Design:** The unrestricted load plan drops both the in-tree and same path requirements. We can accommodate this relaxation in the TLD-IP by removing the variables $y_i^d$ and the constraints (2) and (3), leaving a variable upper bound flow path network design problem with the trailer balance constraints 5). We will refer to the formulation for the unrestricted load plan design problem as ULD-IP.

Finally, we note that in practice it may not be practical for a carrier to implement one of these relaxations for its entire terminal network, since the costs of automating all terminal operations with handheld scanners may not be justified by corresponding benefits. It should be clear, however, that it is possible to formulate blended models where only some terminals relax the constraints that enforce single outbound directs for each destination $d$.

### 7 In-tree Reoptimization Heuristic

Realistically-sized instances of TLD-IP (and its variants) cannot be solved directly by commercial integer programming solvers. Therefore, we develop a local search procedure with neighborhoods defined by carefully chosen restricted versions of TLD-IP; see Algorithm 1 for a general outline of the procedure.

**Algorithm 1** Integer Programming Based Neighborhood Search

**Require:** a feasible solution to TLD-IP

while the search time has not exceeded a prespecified limit $T$ do

Choose a subset of variables $V$

Solve TLD-IP with all variables not in $V$ fixed at their current value

if an improved solution is found then

Update the best known feasible solution

end if

end while

We use a subset of variables $V$ that is motivated by the structural property that the directs selected into a destination terminal $d$ must form a directed in-tree, and we refer to the associated integer program as an In-tree IP into $d$, or $IIP_d$. The purpose of $IIP_d$ is to improve the current solution by optimally choosing the directs used for $d$-bound freight, and by optimally choosing when and where $d$-bound freight is held. The $IIP_d$ problem is to determine a set of paths for all commodities in $K(d)$; note that this problem is then to determine a new directed in-tree to $d$. More formally, given a current feasible solution $(\bar{y}, \bar{x}, \bar{\tau})$, $IIP_d$ is defined by holding fixed the variables.
• \( y_u^t = \bar{y}_d^t \) \( \forall u \in U \) such that \( u \neq d \)

• \( x_p^k = \bar{x}_p^k \) \( \forall k \in K \setminus K(d) \).

A specialized version of Algorithm 1 is presented in Algorithm 2. Note that Algorithm

\begin{algorithm}[H]
\caption{IIP Neighborhood Search}
\begin{algorithmic}
\Require an initial load plan \((\bar{y}, \bar{x}, \bar{\tau})\)
\For {each terminal \( d \)}
\State Set \( F_d = \sum_{k \in K(d)} q_k \), the total amount of freight destined for \( d \)
\EndFor
\State Set \( TERMS = \) array of top 25% of terminals with respect to \( F_d \)
\State Set \( N = |TERMS| \)
\State Sort \( TERMS \) in descending order of \( F_d \)
\State Set \( iter = 0 \)
\While {the search time has not exceeded a prespecified limit \( T \)}
\State Choose destination terminal \( d = TERMS[iter \mod N] \)
\State Solve In-tree IP \( IIP_d \)
\If {Solution to \( IIP_d \) gives lower total load plan cost}
\State Update \((\bar{y}, \bar{x}, \bar{\tau})\)
\EndIf
\State Set \( iter = iter + 1 \)
\EndWhile
\end{algorithmic}
\end{algorithm}

2 can be applied to the traditional or any of its variants by simply modifying the choice of In-tree IP (TLD-IP, DDLD-IP, SPLD-IP, ULD-IP) at each iteration. Note that we never fix the trailer variables \( \tau_a \) to a certain value. Thus, at each iteration of Algorithm 2 empty trailer repositioning decisions are explicitly considered. Since our approach improves the load plan by re-routing freight destined for a specific terminal, we do not want to spend time solving In-tree IPs for terminals for which little freight is destined. Thus, we only consider the top 25% of terminals for which freight is destined. The algorithm we present iterates through this subset of terminals in a round-robin manner.

The success of Algorithm 2 depends on the time needed at each iteration to solve In-tree IPs. To reduce these solution times, we use two sets of valid inequalities derived from the structure of a load plan and the fact that some variables are fixed when solving an In-tree IP. We also utilize a preprocessing rule based on the knowledge we have of where loaded trailers may flow to prune arcs from \( TS - LN \).

Path-continuation inequalities. Path-continuation inequalities are derived from the in-tree structure of a load plan. Suppose that freight originating in Athens, GA and destined for Columbus, OH uses the path of directs \((Athens, Atlanta, Cincinnati, Columbus)\). Then freight originating in Atlanta, GA and destined for Columbus, OH must use the path of directs \((Atlanta, Cincinnati, Columbus)\). Let the first and second in a path \( p \) of directs be denoted by \( f(p) \) and \( s(p) \) respectively. Then, we have the following valid path-continuation inequalities

\[
\sum_{k' \in K(o(k), d(k))} \sum_{p \in P(p, k')} x_{p'}^{k'} \leq \frac{|K(o(k), d(k))|}{|K(s(p), d(k))|} \sum_{k' \in K(s(p), d(k))} \sum_{p \in P(p, j(p, k'))} x_{p}^{k'} \quad \forall k \in K, \forall p \in P(k).
\]  

(9)
Note that while we have presented the path-continuation inequalities in the context of solving an In-tree IP, they are also valid for the TLD-IP. They can also easily be extended to DDLD-IP.

Trailer Disaggregate Inequalities. Trailer disaggregate inequalities are derived from the fact that the paths for some commodities are fixed when solving an In-tree IP. Let \( f_a \) denote the amount of freight that is fixed on arc \( a \) due to fixed commodity paths. For \( IIP_d \), we have \( f_a = \sum_{k \in K \setminus K(d)} \sum_{p \in P(k): a \in p} q_k \bar{x}_{p}^{k} \) and constraints (4) become

\[
\sum_{k \in K(d)} \sum_{p \in P(k): a \in p} q_k \bar{x}_{p}^{k} + f_a \leq \tau_a \quad \forall a \in A.
\] (10)

We first observe that when solving \( IIP_d \), we can bound the variable \( \tau_a \) from below by recognizing that we must have enough trailers to carry the freight that is fixed on the arc \( a \), i.e., \( \tau_a \geq \lceil f_a \rceil \). By also considering the freight that we are trying to route over the arc, we can strengthen the inequality. This gives the following trailer disaggregate inequalities

\[
\tau_a \geq \lceil f_a \rceil + (\lceil f_a + q_k \rceil - \lceil f_a \rceil) \sum_{p \in P(k): a \in p} x_{p}^{k} \quad \forall a \in A, \quad \forall k \in K.
\] (11)

Since we potentially have a trailer disaggregate inequality for each arc and each commodity, it is clearly impractical to add all of them to \( IIP_d \). Hence, we need to determine which might be most effective. Note the derivation of the trailer disaggregate inequality is conducted by replacing a sum over all commodities with a single commodity. Therefore, the inequality will be the strongest on arcs carrying only freight for the chosen commodity \( k \). When we consider any outbound arc from a breakbulk terminal, it is likely that the arc carries freight associated with many commodities. On the other hand, when we consider an outbound arc from an end-of-line terminal, it is likely that the arc carries freight associated with relatively few commodities. Since it is more likely that trailer disaggregate inequalities will be effective on such outbound arcs, we add them to \( IIP_d \) only for outbound arcs from end-of-line terminals.

Preprocessing. Since the size of the time-space network can get large quickly for practical instances, reducing the network size, i.e., eliminating arcs (or flows on arcs), may significantly enhance our ability to solve instances. Certain arcs in the network can only be used for empty repositioning, and often there exist alternate times at which this repositioning can take place. Recognizing this, we can restrict the number of repositioning options. This not only reduces the size of the instances that need to be solved, but also eliminates some of the symmetry embedded in the instances. More specifically, if there is an arc \( a = ((u_1, t_1), (u_2, t_2)) \in A \) where node \((u_2, t_2)\) does not appear in any path for any commodity, then there is no reason to use \( a \) for repositioning as there always exists an alternate repositioning option with the same cost.

8 Computational Results

The algorithm was developed in C++ with CPLEX 11 as the Mixed Integer Program (MIP) solver, interfaced via ILOG Concert Technology. When solving MIPs with CPLEX, we set the MIPEmphasis parameter to integer feasibility and all other parameters to their defaults. When solving instances of the In-tree IP, we use an optimality tolerance of 0.1%; this
typically represents approximately $5,000 of weekly linehaul cost for the carrier we study. All experiments were run on a Debian Linux computer with 32 GB of RAM and 8 Intel Xeon 2.6 GHz processors. We report all times in seconds.

Using a planning horizon of a week for load planning is very typical. Our test set consists of seven instances based on historical data, each consisting of the freight originating for a single week. The instances represent actual freight volumes transported by a super-regional LTL carrier in the U.S., and include three weeks in April of 2008 (Apr08-W1, Apr08-W2, Apr08-W3) and four weeks in March of 2009 (Mar09-W1, Mar09-W2, Mar09-W3, Mar09-W4). When modeling freight which originates in one week, but is due in the subsequent week, we use a wrapped version of the time-space network where arcs connect later time periods in the week to time periods in the beginning.

### 8.1 Solving In-tree IPs

We first focus on configuring and tuning the process of solving In-tree IPs. Specifically, we investigate the benefits derived from the three classes of valid inequalities: trailer disaggregate inequalities (TD), path-continuation inequalities (PC), and same-path inequalities (SP). Note that the same-path inequalities may improve solution times since they are valid, even though any load plan satisfying the in-tree structure requirement automatically satisfies them. We measure the effectiveness of classes of valid inequalities by the decrease in optimality gap at the root node of the search tree, the number of integer programs that can be solved in a fixed amount of time, and the savings obtained in a fixed amount of time. For this analysis, we use instance Apr08-W1.

Recall that our neighborhood search approach solves In-tree IPs for the top 25% of destinations in terms of inbound freight, in this case 40 terminals. In the results presented below, the values are averaged over the instances defined by these 40 terminals. Since we have the load plan that was used by the carrier for the test week, we have an initial solution for each $IIP_d$ and hence an upper bound $ub_d$ on the optimal value of $IIP_d$. The root relaxation value reported by CPLEX provides a lower bound $lb_d$ on the optimal value of $IIP_d$ and thus an optimality gap $gap_{init} = (ub_d - lb_d)/ub_d$. When we add classes of valid inequalities to $IIP_d$, we obtain an improved lower bound at the root and thus an improved optimality gap. In Table 1, we report the number of rows, the improvement in optimality gap, and the time to solve the root relaxation when using certain classes of valid inequalities.

<table>
<thead>
<tr>
<th># Rows</th>
<th>Gap Reduction</th>
<th>Root Solve Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Inequalities</td>
<td>37,252</td>
<td>-</td>
</tr>
<tr>
<td>TD</td>
<td>37,438</td>
<td>1.02</td>
</tr>
<tr>
<td>TD + PC</td>
<td>39,400</td>
<td>4.71</td>
</tr>
<tr>
<td>TD + PC + SP</td>
<td>44,587</td>
<td>12.88</td>
</tr>
</tbody>
</table>

We see that as we add classes of valid inequalities, the optimality gap gets smaller, but with an increase in the time required to solve the root relaxation that is likely due to the increased number of rows.

While smaller optimality gaps are desirable, we are primarily interested in reducing the time required to solve $IIP_d$ instances, since this will allow us to solve more of them in a fixed amount of time. Thus, we next report in Table 2 how many $IIP_d$ instances can be solved to within 0.1% of optimality in three minutes. The “Number Solved” column reports how many of the 40 $IIP_d$ instances could be solved. The “Avg. Time” column
reports the average time it took to solve the instances that were solved. Finally, the “Avg. Time*” column reports the average time it took to solve the 27 instances that could be solved without adding any of the valid inequalities.

<table>
<thead>
<tr>
<th>No Inequalities</th>
<th>Number Solved</th>
<th>Avg. Time</th>
<th>Avg. Time*</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD</td>
<td>27</td>
<td>46.41</td>
<td>46.41</td>
</tr>
<tr>
<td>TD + PC</td>
<td>29</td>
<td>38.97</td>
<td>35.44</td>
</tr>
<tr>
<td>TD + PC + SP</td>
<td>30</td>
<td>48.10</td>
<td>40.89</td>
</tr>
</tbody>
</table>

We see that for each configuration of classes of valid inequalities, the average time required to solve the 27 instances that could be solved without adding any valid inequalities reduces. Furthermore, we see that adding path-continuation plus trailer disaggregate inequalities increases the number of instances that can be solved and reduces the average time required to solve the 27 instances that could be solved without adding any valid inequalities by nearly 25%. Although additionally including the same-path inequalities does enable the solution of one more instance, it increases the average time required to solve the 27 instances. This is likely due to the increase in solve time of the root relaxation.

The ultimate goal of adding classes of valid inequalities is to solve more \( IIP_d \) instances in a fixed amount of time, which hopefully leads to greater savings. Hence, we next report in Table 3 the savings that are obtained when running the neighborhood search for 30 minutes, where savings are defined as: \( \frac{\text{cost of the initial load plan minus the cost of the load plan designed by the algorithm}}{\text{cost of the initial load plan}} \).

<table>
<thead>
<tr>
<th>No Inequalities</th>
<th>TD</th>
<th>TD + PC</th>
<th>TD + PC + SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.54</td>
<td>1.81</td>
<td>2.01</td>
<td>1.82</td>
</tr>
</tbody>
</table>

We see that again the path-continuation and trailer disaggregate inequalities yield the best performance; enabling nearly 0.5% more savings to be found than when no cuts were added.

Based on the results of these experiments, we chose to explicitly add the trailer disaggregate and path-continuation inequalities to the \( IIP_d \) formulation. Alternatively, the same-path inequalities were included as CPLEX user cuts in all further experiments.

The integer program \( IIP_d \) represents a special type of network design problem, a class of optimization problems which is notoriously difficult to solve due to weak dual bounds. Yet, we are able to solve instances with more than 20,000 constraints and more than 300,000 variables (after CPLEX preprocessing) in less than one minute. Much of this success can be attributed to the freight that is fixed on arcs and effective preprocessing. In fact, even without adding any valid inequalities the average optimality gap is only 0.52% for the instances used to produce the results reported in Table 1. However, the root relaxation value used to compute the optimality gap is significantly higher than the value of the linear programming relaxation (LPR) of the original formulation. For the instances used to produce the results reported in Table 1, the average optimality gap when computed using the optimal value of LPR is 54.74%. Network design problems often exhibit large optimality gaps due to the fact that a solution to LPR only pays for exactly the capacity that is used. By recognizing (in preprocessing) that the number of trailers required on arc
a to carry the fixed freight $f_a$ is $\lceil f_a \rceil$, the bound produced by LPR can be strengthened significantly. Adding these bounds reduces the average gap from 54.74% to 0.65%. Hence, we see that even though the approach relies on repeatedly solving instances of the fixed charge capacitated network design problem, we avoid much of the difficulty associated with solving this type of integer program because much of the fixed charge is already paid.

### 8.2 Traditional Load Plan Improvements

The primary goal of our research is to develop technology that can produce more cost-effective load plans. In Table 4, we report the savings ("ΔCost") and the increase in pounds per trailer ("ΔPPT") when we run the neighborhood search on the April 2008 and March 2009 instances for 6 hours. Again, these improvements are measured in percentages relative to the initial load plan provided by the carrier for the relevant week. For these experiments, we provided CPLEX a time limit of 90 seconds to solve each $IIP_d$ instance. Since a 1% savings represents about $60,000 per week for the carrier, these suggested changes indicate a substantial improvement to the carrier’s bottom line. The increase in pounds per trailer indicates that the neighborhood search does increase consolidation and thus finds significant savings for each of the weeks. While the approach finds savings in both data sets, they are greater in the April 2008 weeks.

<table>
<thead>
<tr>
<th>Apr08-W1</th>
<th>Apr08-W2</th>
<th>Apr08-W3</th>
<th>Mar09-W1</th>
<th>Mar09-W2</th>
<th>Mar09-W3</th>
<th>Mar09-W4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ΔCost</td>
<td>4.49</td>
<td>3.94</td>
<td>4.77</td>
<td>3.73</td>
<td>3.29</td>
<td>3.69</td>
</tr>
<tr>
<td>ΔPPT</td>
<td>2.17</td>
<td>1.37</td>
<td>3.13</td>
<td>2.37</td>
<td>1.68</td>
<td>1.66</td>
</tr>
</tbody>
</table>

One of the important features of our approach is that it integrates the planning of loaded and empty trailer movements. To analyze the value of such joint planning, we compare the load plans resulting from our approach with those constructed via a sequential approach, i.e., an approach in which freight is routed first (hence determining flows of loaded trailers) and trailer balance is restored second via repositioning. To implement such a sequential approach, we first run our neighborhood search with optimization problems that do not include constraints (5). The result is a solution $(x^1, y^1, \tau^1)$ which is not necessarily feasible for TLD-IP. We next solve TLD-IP, but fixing variables $x = x^1$ and $y = y^1$ to re-position trailers and create a feasible solution to TLD-IP. Note that with these variable fixed, TLD-IP reduces to a minimum cost network flow problem. The results of this experiment are provided in Table 5, where we report the cost savings achieved by our integrated approach ("Integrated") and the sequential approach ("Sequential").

<table>
<thead>
<tr>
<th>Apr08-W1</th>
<th>Apr08-W2</th>
<th>Apr08-W3</th>
<th>Mar09-W1</th>
<th>Mar09-W2</th>
<th>Mar09-W3</th>
<th>Mar09-W4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated</td>
<td>4.49</td>
<td>3.93</td>
<td>4.77</td>
<td>3.73</td>
<td>3.29</td>
<td>3.69</td>
</tr>
<tr>
<td>Sequential</td>
<td>2.49</td>
<td>2.49</td>
<td>2.10</td>
<td>3.21</td>
<td>2.82</td>
<td>3.08</td>
</tr>
</tbody>
</table>

Clearly, the integrated approach leads to significant increases in savings when compared to an idealized sequential approach.

In Table 6, we compare the individual cost components (loaded trailer costs, empty trailer costs, and handling costs) of the load plans produced by each approach relative to the carrier’s load plan. Specifically, we define $c_{comp}^{\text{init}}$ to be the cost for a component in the
carrier’s load plan and $c^\text{comp}_{\text{Alg}}$ to be the cost for a component of the load plan produced algorithmically, and report in the columns values of $\frac{c^\text{Component}_{\text{Alg}}}{c^\text{Component}_{\text{Init}}} \times 100\%$. While the loaded trailer costs and the handling costs are about the same for both approaches, routing freight and empty trailers sequentially results in an increase in empty trailer costs compared to the carrier’s initial load plan.

Finally, we illustrate in Figure 10 the savings found by the neighborhood search for the Apr08-W1 instance over the course of its execution. We see that although we allow the heuristic to run for 6 hours, the vast majority of savings are found in the first 3 hours.

![Figure 10: Savings Found by Algorithm over Time](image)

8.3 Variations on the Traditional Load Plan

Having established that integer programming based neighborhood search can produce traditional load plans with the potential for significant savings, we next turn to the question of what savings can be achieved by relaxing different constraints of the traditional load plan. Again, savings are computed relative to the cost of the carrier’s load plan for the relevant week. Not surprisingly, we see in Table 7 that unrestricted load plans lead to the greatest savings. Note that while day-differentiated load plans lead to significantly greater savings than traditional load plans, same-path load plans do not.

We next compare the individual cost components of each load plan variant in Table
Table 7: Cost Savings For Load Plan Variants

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>Day-Differentiated</th>
<th>Same-Path</th>
<th>Unrestricted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr08-W1</td>
<td>4.49</td>
<td>6.08</td>
<td>4.93</td>
<td>7.41</td>
</tr>
<tr>
<td>Apr08-W2</td>
<td>3.94</td>
<td>6.11</td>
<td>4.90</td>
<td>7.62</td>
</tr>
<tr>
<td>Apr08-W3</td>
<td>4.77</td>
<td>6.93</td>
<td>5.14</td>
<td>8.05</td>
</tr>
<tr>
<td>Mar09-W1</td>
<td>3.73</td>
<td>6.51</td>
<td>4.01</td>
<td>7.60</td>
</tr>
<tr>
<td>Mar09-W2</td>
<td>3.29</td>
<td>6.02</td>
<td>3.44</td>
<td>7.22</td>
</tr>
<tr>
<td>Mar09-W3</td>
<td>3.69</td>
<td>6.51</td>
<td>3.85</td>
<td>7.60</td>
</tr>
<tr>
<td>Mar09-W4</td>
<td>3.77</td>
<td>6.76</td>
<td>3.83</td>
<td>7.89</td>
</tr>
</tbody>
</table>

8 where we use the following abbreviations: Traditional (T), Day-Differentiated (DD), Same-Path (SP), and Unrestricted (U). We see that the opportunity to vary freight routing decisions by day (which is present in both the Day-Differentiated and Unrestricted load plans) allows for a significant decrease in empty trailer repositioning costs.

Table 8: Load Plan Variants Cost Component Comparison

<table>
<thead>
<tr>
<th></th>
<th>Loaded Trailers</th>
<th>Empty Trailers</th>
<th>Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>DD</td>
<td>SP</td>
</tr>
<tr>
<td>Apr08-W1</td>
<td>96.74</td>
<td>96.26</td>
<td>96.27</td>
</tr>
<tr>
<td>Apr08-W2</td>
<td>97.02</td>
<td>95.88</td>
<td>96.07</td>
</tr>
<tr>
<td>Apr08-W3</td>
<td>96.11</td>
<td>95.08</td>
<td>96.19</td>
</tr>
<tr>
<td>Mar09-W1</td>
<td>97.69</td>
<td>95.55</td>
<td>97.61</td>
</tr>
<tr>
<td>Mar09-W2</td>
<td>98.33</td>
<td>96.35</td>
<td>98.23</td>
</tr>
<tr>
<td>Mar09-W3</td>
<td>98.31</td>
<td>96.03</td>
<td>98.18</td>
</tr>
<tr>
<td>Mar09-W4</td>
<td>98.46</td>
<td>97.99</td>
<td>99.37</td>
</tr>
</tbody>
</table>

In both a Day-Differentiated and Unrestricted load plan, we relax the restriction that freight between two terminals must follow the same sequence of directs regardless of the day of origin. Therefore, we present in Table 9 how often different paths are used during the week. Specifically, Table 9 reports what percentage of origin-destination pairs whose path could have been changed during the execution of the heuristic sends freight along 1, 2, 3, 4, or 5 paths during the week. For the Day-Differentiated load plan, we see that the majority of origin-destination pairs still use a single path for each weekday. Roughly 19.5% of origin-destination pairs sends freight along two paths during the week. Finally, only few origin-destination pairs send freight on three or more paths. From an implementation perspective, this “path profile” is desirable since the Day-Differentiated load plan does not represent a significant shift from how freight would be routed under a traditional load plan.

Table 9: Allowing Different Paths on Different Days

<table>
<thead>
<tr>
<th># Diff.</th>
<th>Apr08-W1</th>
<th>Apr08-W2</th>
<th>Apr08-W3</th>
<th>Mar09-W1</th>
<th>Mar09-W2</th>
<th>Mar09-W3</th>
<th>Mar09-W4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paths</td>
<td>DD</td>
<td>U</td>
<td>DD</td>
<td>U</td>
<td>DD</td>
<td>U</td>
<td>DD</td>
</tr>
<tr>
<td>1</td>
<td>79.57</td>
<td>75.35</td>
<td>80.40</td>
<td>75.76</td>
<td>77.78</td>
<td>74.95</td>
<td>75.16</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.53</td>
<td>0.21</td>
<td>0.34</td>
<td>0.42</td>
<td>0.70</td>
<td>0.73</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.02</td>
<td>0.02</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>
For the Unrestricted load plan, we see that while slightly fewer origin-destination pairs send freight along a single path, compared to the Day-Differentiated load plan, the vast majority still do.

In contrast, the Same-Path load plan retains the constraint that freight between two terminals must follow the same sequence of directs each day, but relaxes the in-tree structure of a traditional load plan. Specifically, these variants allow two commodities with the same destination to take different outbound directs from a terminal. Day-Differentiated plans also allow this, but only on different weekdays. In Tables 10 and 11, we report how many terminals use different outbound directs at some point during the week to route freight to the same destination. Note for both tables we again only report the origin-destination pairs whose path could have been changed during the execution of the heuristic. We see that relaxing the tree structure while enforcing the same path constraints generates load plans that are very similar to traditional load plans; i.e., they rarely violate the single outbound direct constraint. However, the Day-Differentiated and Unrestricted load plans that we generate are such that the vast majority of the single outbound direct constraints are satisfied. Even when multiple outbound directs are used, rarely are more than two used. This is not completely surprising, since using a single outbound direct for freight destined for a specific terminal is a form of consolidation.

Table 10: % Terminals That Load a Single Destination on Multiple Directs - Apr08

<table>
<thead>
<tr>
<th># Outbound Directs</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SP</td>
<td>DD</td>
<td>U</td>
</tr>
<tr>
<td>1</td>
<td>97.54</td>
<td>78.84</td>
<td>74.64</td>
</tr>
<tr>
<td>2</td>
<td>2.39</td>
<td>17.92</td>
<td>20.71</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>3.00</td>
<td>3.91</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.21</td>
<td>0.69</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>6+</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 11: % Terminals That Load a Single Destination on Multiple Directs - Mar09

<table>
<thead>
<tr>
<th># Outbound Directs</th>
<th>W1</th>
<th>W2</th>
<th>W3</th>
<th>W4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SP</td>
<td>DD</td>
<td>U</td>
<td>SP</td>
</tr>
<tr>
<td>1</td>
<td>98.45</td>
<td>75.81</td>
<td>66.69</td>
<td>98.71</td>
</tr>
<tr>
<td>2</td>
<td>1.55</td>
<td>19.32</td>
<td>24.59</td>
<td>1.25</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>4.22</td>
<td>7.18</td>
<td>0.04</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.62</td>
<td>1.32</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.04</td>
<td>0.18</td>
<td>0.00</td>
</tr>
<tr>
<td>6+</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
</tr>
</tbody>
</table>

9 Future Research

We believe that there is potential to improve our solution heuristic for load plan design in a few ways; namely by solving a different restriction of the TLD-IP (potentially motivated by the hub-and-spoke structure of LTL networks), and by developing more sophisticated approaches for determining the In-tree IP to solve at each iteration. We also believe that the heuristic approach we have developed could be extended to include facility location and/or role decisions, where the terminal role refers to whether or not it should be used as a breakbulk transfer.
Finally, there is the question of real-time load planning, the problem of determining how to alter plans as operations unfold. By deploying wireless devices in pickup and delivery operations, carriers have relatively early visibility of the freight that will enter the linehaul network on a given day. Hence, while load plan design is typically treated as a planning problem and is based on projected or nominal freight volumes, one can easily envision the savings potential of designing a load plan to be executed each night given an accurate estimate of that day’s freight volumes.

References


