

Improved Load Plan Design Through Integer Programming Based Local Search

Alan Erera¹, Michael Hewitt ^{*2}, Martin Savelsbergh³, and Yang Zhang¹

¹School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta,
GA 30332-0205

²Department of Industrial and Systems Engineering, Rochester Institute of Technology,
Rochester, NY 14623

³School of Mathematical and Physical Sciences, University of Newcastle, Callaghan, NSW
2308, Australia

Abstract

We present integer programming models of the service network design problem faced by less-than-truckload (LTL) freight transportation carriers, and a solution approach for the large-scale instances that result in practical applications. To accurately represent freight consolidation opportunities, the models use a fine discretization of time. Furthermore, the models simultaneously route freight and empty trailers, and thus explicitly recognize the efficiencies presented by backhaul lanes. The solution approach can generate the traditional service network designs commonly used by LTL carriers, but also enables the construction of designs that allow more flexibility, e.g., that allow freight routes to vary by day of week. An iterative improvement scheme is employed which 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. An LTL 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. LTL carriers collect and consolidate freight from multiple shippers to increase trailer utilization, and route freight through a network of consolidation terminals between origins and destinations. The savings generated by increasing trailer utilization is partially offset by other costs; terminal-to-terminal routing increases the total time and distance each shipment requires to reach its destination, and transferring freight between trailers generates handling costs at terminals.

The service network design problem considered in this paper is to decide how freight should be routed through a terminal network, specifying sequences of terminals where trailer-to-trailer transfer takes place for each shipment moved. In the LTL industry, this problem is known as determining a *load plan*. The load plan determines which shipments are packed together into the same trailers; thus, a load plan is a freight consolidation plan.

*corresponding author; email: mrheie@rit.edu, phone: +1 585 475 5571, fax: +1 585 475 2520

Constructing effective load plans is more critical than ever to the success of large LTL carriers. National carriers now compete with both so-called super-regional carriers (resulting from mergers of regional operators), and LTL services offered by traditional package express companies like UPS and FedEx. This has led to increased competition and pressure on pricing, and the development of new services with shorter quoted transit times often with guarantees.

In this paper, we discuss the design and implementation of load planning technology that is significantly more effective than existing solutions. The technology uses a detailed model of LTL operations that explicitly and accurately models the movement of loaded and empty trailers over time, and thus better captures true freight consolidation opportunities. Furthermore, the technology is flexible and allows load planning in cases where traditional simplifying operating rules are relaxed to enable larger cost savings. 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.

We formulate the load planning problem using a type of integer multi-commodity flow model on a time-space network with commodities representing origin-destination demands. For a large U.S. carrier, the number of commodities and the size of the time-space network are very large (tens of thousands of commodities, thousands of nodes, and hundreds of thousands of arcs) resulting in enormous integer programs. To effectively solve these formulations, we use an approach that combines exact optimization with heuristic search. The approach is a local search heuristic where a neighborhood contains all feasible solutions to a smaller integer program defined given the current feasible solution, and the search for an improving solution is performed by solving that integer program.

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 4% weekly for a national carrier,
- models time using a discretization appropriate for the tight service standards a carrier must offer to remain competitive,
- integrates empty trailer repositioning with freight routing decisions,
- is the first to design day-differentiated load plans that account for predictable daily freight volume fluctuations, which yields even greater cost-savings; approximately 6.5% 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 describes the LTL industry and load plan design in particular, and discusses how we model key current challenges in load plan design. Section 3 presents a brief review of relevant literature. Section 4 then presents our approach in detail, including how we model routing and holding freight, an integer programming formulation of the load plan design problem and some of its variants, and a solution heuristic. Finally, Section 5 presents the results of an extensive computational study conducted using data from a national LTL carrier.

2 LTL Operations and Load Planning

LTL networks use two terminal types: *satellite* or *end-of-line* (EOL) terminals that serve only as origin or destination terminals, and *breakbulk* (BB) terminals that additionally serve as transfer points for shipments. The set of EOL and BB terminals is known as the *linehaul* network. A separate operation at 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 served by a single terminal. Collected shipments are then brought back to the terminal, and sorted by destination. This is the first level of consolidation. If there is not enough freight to justify loading dedicated (direct) trailers to a destination terminal, shipments to that destination may be instead loaded on a trailer to an intermediate BB terminal; this is the second level of consolidation. One example model of operations is for each EOL terminal to load all freight into trailers destined to one or two nearby BB terminals. Outbound freight from each breakbulk for some destinations may be loaded into direct trailers, while for others may be loaded to another downstream breakbulk for another round of consolidation. The freight transfer operation at breakbulks requires unloading and reloading of trailers, and thus generates both *handling time* and *handling cost*.

An originating shipment is typically delivered by the city operation to the origin terminal no later than 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 example, 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.

2.1 Load plans

The fundamental building block in load planning is a direct trailer, or a *direct*. If a shipment is assigned to direct $A \rightarrow B$, it is loaded into a trailer at terminal A which is not unloaded (and hence the freight not handled) 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 . A *load path* for a particular shipment is a sequence of directs connecting its origin terminal to its destination terminal. A *load plan* specifies a load path for each shipment, and thus prescribes where all freight is handled.

Traditional load plans have a specific structure, which we denote as a *uniform in-tree structure*. First, a unique load path is used for all shipments moving between origin terminal o and destination terminal d . Second, the set of all load paths inbound to d from all origins forms a directed in-tree on the network of directs. Therefore, each shipment with destination d that includes terminal i in its load path must use a unique outbound direct (i, j) . Traditional load plans prescribe that the load path from o to d is uniform for each weekday; in practice, carriers alter the load plan during operations to reduce costs or improve service measures when possible. A common example is loading a direct trailer that skips one or more planned terminals in the load plan path to reduce handling costs when enough freight is available. Conversely, a carrier may also cancel a direct trailer, moving shipments through an intermediate BB to avoid sending a lightly-loaded trailer.

Load plans are designed to enable shipments to meet service standards at minimum total linehaul cost, consisting of both:

1. Transportation cost associated with moving loaded and empty trailers, and
2. Handling cost associated with transferring freight between trailers.

2.2 Enhanced load planning

In this paper, we develop an enhanced procedure for constructing load plans that includes three key features that are essential for modern load planning: (1) detailed time-space modeling of shipment and trailer dispatching; (2) integrated consideration of the movement of loaded and empty trailers; and (3) modeling of flexible load plans that do not require a uniform in-tree structure throughout the week.

- **Detailed Time-space Modeling:**

We model potential dispatches of trailers between terminals at a relatively fine time discretization, with dispatches from breakbulk terminals occurring at 8 different times during each weekday. Such a model can accurately capture consolidation opportunities in time. One driving force necessitating a detailed model are the shorter service standards offered by LTL carriers today, often 1, 2, or 3 days. Figure 1 depicts the fraction of shipments with different service standards served by a large U.S. carrier.

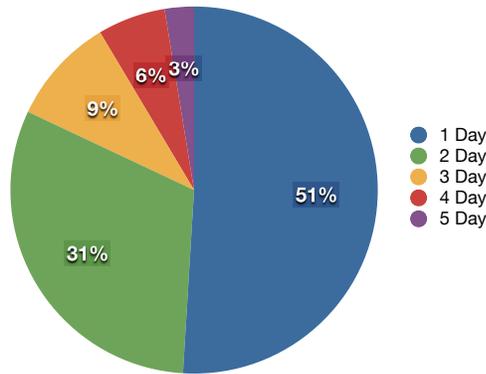


Figure 1: Number of shipments by service standard

Since overnight shipments may still be transferred at more than one breakbulk en route, it is very important to model accurately when such transfers will occur during the overnight hours. Consider the following simple example. Suppose Memphis to Atlanta freight with one-day service needs to depart Memphis no later than 1 a.m. to arrive in Atlanta in time for next day delivery. On the other hand, there may be Kansas City to Miami two-day service that transfers in Memphis and then Atlanta, but this freight does not arrive in Memphis until 3 a.m. Thus, Kansas City to Miami freight cannot feasibly travel to Atlanta in trailers with Memphis to Atlanta direct freight, since it arrives too late. Furthermore, it must depart too early from Memphis to be consolidated with the next day's Memphis to Atlanta freight, since Memphis to Miami through Atlanta requires 20 hours assuming a 2-hour transfer in Atlanta.

- **Loaded and Empty Trailer Movement Integration:**

Our procedure is not the first to use models that capture the costs of moving both loaded and empty trailers in service network design, but it is important to understand why it is important to do so especially when using a detailed model of shipment and trailer dispatching.

To illustrate this idea, consider the small static network presented in Figure 2, where the numbers above each arc represent the dispatch cost per trailer.

Suppose the network serves the following freight, with volume is measured in trailerloads:

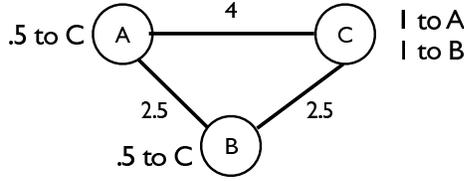


Figure 2: Example Network

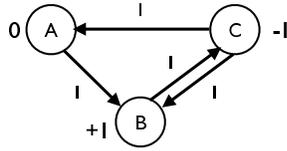


Figure 3: Trailer routing minimizing total loaded trailer costs.

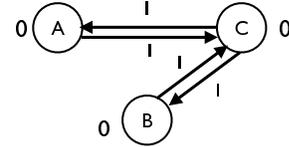


Figure 4: Integrated routing decisions.

- 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 .

Minimizing only the costs of moving loaded trailers yields the solution in Figure 3. Freight from C is routed directly to A and B , while freight from A to C is consolidated with the B to C freight at B . Note that the loaded trailer movements depicted in the figure are not balanced at B or C , so an additional empty trailer must be moved from B to C . The loaded trailer cost in this solution is 11.5, and adding the empty cost yields a total cost of 14. Moving all of the freight direct yields the solution in Figure 4, which has larger loaded trailer cost (13), but requires no empty trailer costs and thus lower total cost.

Carriers refer to lanes like (A, C) and (B, C) as *backhaul* lanes, and build load plans in practice that route freight along such lanes to reduce total costs. Algorithmic approaches should also do so.

• **Flexible Load Plans:**

Traditional load plans with the uniform in-tree structure have a few drawbacks in practice. First, LTL carriers typically do not pick up or deliver freight on Saturday or Sunday in the U.S., but linehaul dispatches and freight sorting are possible on those days. Thus, there is an opportunity to route freight with the same service standard between the same origin terminal and destination terminal differently if the weekend separates the pickup day from the delivery day. Second, there are predictable variations in some terminal-to-terminal freight demands by day of week, due to the influence of larger shippers. Since this variation is predictable, there is again an opportunity to route freight differently by weekday. And, the enabling technology to support the resulting daily changes in terminal operations is now widely available and reasonably priced.

Table 1 illustrates this second point, depicting the fraction of origin-destination terminal pairs with positive freight volume for a certain number of days a week for a large U.S. carrier. Note that for half of the terminal pairs, freight is only shipped once or twice per week.

Table 1: % O-D freight by Frequency

% O-D pairs	Frequency
33.42	1
20.84	2
14.19	3
12.78	4
18.74	5

Thus, in this paper we introduce a more flexible load plan that we denote a *day-differentiated load plan*. A day-differentiated plan specifies a unique load path for freight originating on the same weekday at origin terminal o and bound for destination d . To do so, we relax the uniform in-tree structure of a traditional load plan such that the outbound direct (i, j_k) for freight at i destined for d can depend on the weekday k . 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.”

Day-differentiated load plans may lead to substantial cost savings. To illustrate, consider again the example network in Figure 2, but now with the following freight volumes (depicted in Figure 5):

- **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 half originating at A and destined for C ,
 - * One half 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 6 (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 7 on Tuesday through Friday.

3 Literature Review

Models for load plan design decisions are typically similar to the classic network design problem, which has been extensively studied. Exact optimization is often impractical for network design, and therefore heuristic solution techniques are needed. Metaheuristics have been developed that find good feasible 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 to be split among multiple paths. Furthermore, none considers underlying equipment moves (including empty repositioning movements).

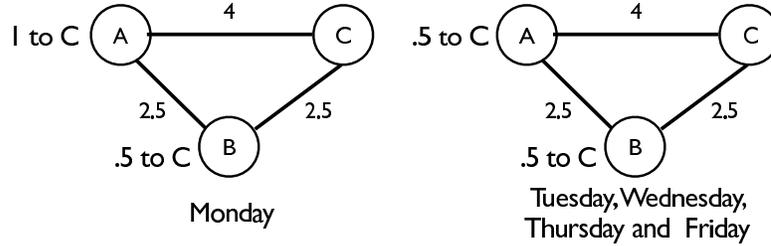


Figure 5: Example Network



Figure 6: Monday load plan does not consolidate Figure 7: Tuesday-Friday load plan consolidates freight through BB terminal B .

Load plan design can also be viewed as a special case of service network design; this problem class has also received much attention (see [Crainic, 2000] or [Wieberneit, 2008] for a review). 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 accommodate the in-tree structure of a load plan.

Relatively little research has focused specifically on LTL load plan design. 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 static network models, [Powell and Farvolden, 1994] present a dynamic network model where freight flows on paths through time and space. The paper presents an alternative heuristic that relies on determining service network arc 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 different dynamic network model that more accurately captures consolidation timing and is structurally similar to ours. However, the model measures consolidation (trailer utilization) by assuming all dispatches from a terminal on a specific day occur at the same time; this assumption may substantially overestimate consolidation opportunities. While loaded and empty routing decisions are modeled jointly, the solution approach considers them separately and sequentially with a feedback loop mechanism. To the best of our knowledge, no authors have developed models for creating day-differentiated load plans.

The neighborhood search 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. A similar

approach is developed for the classic network design problem in [Hewitt et al., 2010]. 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], [Savelsbergh and Song, 2008], and [Espinoza et al., 2008].

4 Finding Improved Load Plans

Load planning is primarily a problem of freight routing: given shipment volumes between origin and destination terminals, a load plan determines how freight is routed across the terminal network in consolidated trailerloads. Our load planning model uses a detailed time-space network to represent the dispatch of trailers and the temporary holding of freight at terminals within each weekday for a planning horizon of one week. We then define an integer program (IP) to select a set of freight load paths in this detailed time-space network, one for each origin-destination terminal pair for each weekday, such that the set of paths conforms to either the uniform or day-differentiated in-tree structure. Due to the size of linehaul networks in practice and the required fidelity of the time discretization, instances of this IP cannot be solved directly by commercial solvers.

Therefore, a customized solution approach has been developed. Recently, the idea of solving restricted IPs to obtain high-quality solutions has been shown to be quite successful, both for generic IPs, e.g., [Danna et al., 2005], as well as for specific IPs, e.g., [Hewitt et al., 2010]. This idea forms the basis for our customized approach: a local search procedure with neighborhoods defined by carefully chosen restricted load planning IPs. See Algorithm 1 for a high-level description.

Algorithm 1 IP-based Neighborhood Search

Require: a load plan

```

while the search time has not exceeded a prespecified limit  $T$  do
  Choose a subset of variables  $V$ 
  Solve 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

```

More specifically, at each iteration, the algorithm chooses a subset of variables V involving the routing of freight destined for a single terminal. The choice to re-optimize the routing and holding of freight destined for a single terminal at each iteration is motivated partially by the structure of traditional load plans and partially by the intuition that freight traveling to a single destination can likely travel in the same trailer.

Next, we discuss the construction of the time-space network, the way freight routing is modeled, and the different opportunities considered for holding freight. After that, we present the integer program that forms the heart of our solution approach when traditional load plans are considered and discuss the solution approach in more detail. Lastly, we discuss how the solution approach can be applied to obtain day-differentiated load plans and other load plan variants.

4.1 Modeling Freight Routing

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 the planning horizon, we define

the time-space linehaul network $TS-LN = (N, A)$, where N denotes the set of nodes and A 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 greater than or equal to the transit time of the underlying direct l . We also create arcs $a = ((u_1, t_1), (u_1, t_2))$ to connect consecutive timed copies of each node u_1 . These arcs allow us to model holding a trailer or shipment at terminal u_1 .

We divide a day into a set of time 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} and the time discretization is defined by the breakpoints. (See Appendix A for a discussion on the challenges associated with choosing an appropriate discretization of time.)

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 $t_1 = d_1$ at 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 $t_2 = d_2$ at 8 am. All freight shipments with a common (n_1, n_2) pair are considered a single *commodity* k . Carriers typically quote a single service standard for freight originating at terminal u_1 and destined for terminal u_2 . Although we make this assumption, accommodating multiple classes of service (say “regular” and “express”) between u_1 and u_2 can easily be accommodated in this modeling framework.

We model potential routes for freight from its origin to destination terminal with paths on $TS-LN$. Therefore, let $P(k)$ be a set of possible freight paths for commodity k , where a freight path p is a sequence of arcs, i.e., $p = (a_1, \dots, a_{n_p})$, and each $a_i \in A$. Each path $p = (a_1, \dots, a_{n_p}) \in P(k)$ is constructed such that a_1 departs the commodity’s origin node and a_{n_p} arrives at its destination node. 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 $p = (a_1, \dots, a_{n_p})$ is an underlying path \mathbf{p} of directs $\mathbf{p} = (l(a_1), \dots, l(a_{n_p}))$. Note that given a path p , we can calculate its total per-unit handling cost h_p 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) using Dijkstra’s algorithm ([Dijkstra, 1959]). Because 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 $a = ((u_2, \text{Tuesday @ 8am}), (u_3, t_3))$. 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 ≤ 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 $a = ((u_2, \text{Tuesday @ 8am}), (u_3, t_3))$.

We recognize that given this mapping methodology, for different paths for different commodities, the arc $((u_2, \text{Tuesday @ 8am}), (u_3, t_3))$ may reflect 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 can overestimate consolidation opportunities. However, it should be noted that the handling time are only estimates and that 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 load 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 8. 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.

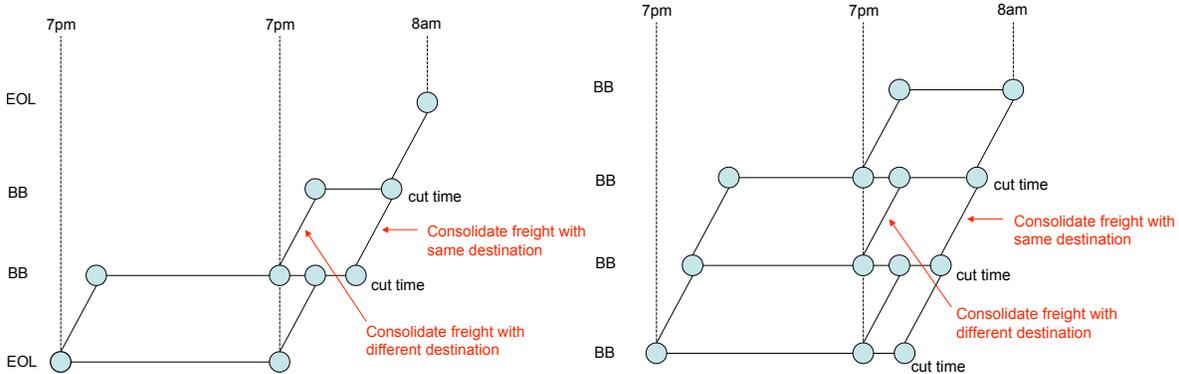


Figure 8: Holding Freight for Consolidation

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 destination *terminal*. 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 direct in LN departing 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 9). 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 on $TS - LN$ a path of arcs, where

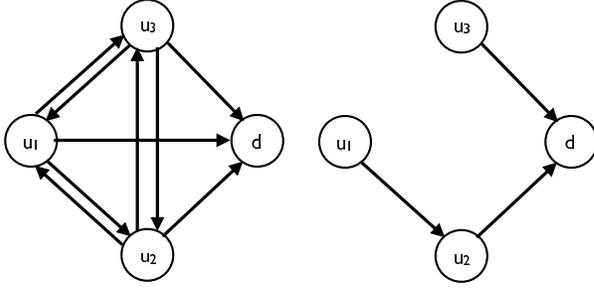


Figure 9: LN for a four terminal network, and load plan as a directed in-tree into d

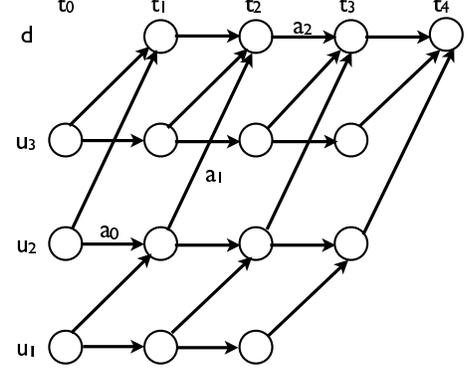


Figure 10: Time-space network depicting routing choices into d given a load plan.

an arc 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 destined for terminal 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 10 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 $dn(k) = (d, t_i)$ given in Figure 9. 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 .

4.2 Load Plan Design Integer Program

We next present an integer programming formulation of the traditional load plan design problem considered in this paper. While our solution approach will not solve this formulation directly, it is the foundation of the integer program that we solve in our local search heuristic. We must first define some further notation.

Given networks $LN = (U, L)$ and $TS - LN = (N, A)$, we let $\Delta^+(u) \subseteq L$ denote the set of potential outbound directs from terminal $u \in U$, let $l(a)$ denote the direct $l \in L$ corresponding to the arc $a \in A$, let $\delta^+(n) \subseteq A$ denote the set of outbound arcs from node $n \in N$, let $\delta^-(n) \subseteq A$ denote the set of inbound arcs to node $n \in N$, and let c_a denote the per-trailer travel cost along arc $a \in A$. We let K denote the set of commodities. For each commodity $k \in K$, let $o(k) \in U$ denote the origin terminal, let $d(k) \in U$ denote the destination terminal, let w_k denote the weight in pounds, and let q_k denote its size measured in fractional trailers (note, q_k need not be less than one).

The integer program has three sets of decision variables. First, x variables indicate whether commodity k uses path p , *i.e.*, $x_p^k \in \{0, 1\} \forall k \in K, \forall p \in P(k)$. Second, 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.*, $y_l^d \in \{0, 1\} \forall d \in U, \forall l \in \Delta^+(u), u \in U$. Finally, τ variables count the number of trailers

(empty or loaded) that move on arc a , i.e., $\tau_a \in \mathbb{Z}_+ \quad \forall a \in A$.

The formulation is to then minimize

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

subject to

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

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

$$\sum_{p \in P(k): a \in p} x_p^k \leq y_{l(a)}^{d(k)} \quad \forall k \in K, \quad \forall a \in A, \quad (3)$$

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

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

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 freight 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 by aggregating commodities with a common destination. Let $P(k)$ denote the set of paths of directs for commodity $k \in K$, and let $K_l(d) \subseteq K$, for a destination $d \in U$, denote the set of commodities $k \in K(d)$ such that there exists a path $p \in P(k)$ containing the direct l . Then, by aggregating over the set $K_{l(a)}(d)$, the following constraints are equivalent to (3):

$$\sum_{k \in K_{l(a)}(d)} \sum_{p \in P(k): a \in p} x_p^k \leq |K_{l(a)}(d)| y_l^d \quad \forall a \in A \quad \forall d \in U. \quad (6)$$

4.3 In-tree Reoptimization Heuristic

We next specify how we implement Algorithm 1. 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. The purpose of restricted integer program 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. That is, the restricted integer program determines a new directed in-tree into d . More formally, given a current feasible solution $(\bar{y}, \bar{x}, \bar{\tau})$ and a destination terminal d , the restricted integer program, IIP_d , is defined by holding fixed the variables

- $y_l^u = \bar{y}_l^u \quad \forall u \in U$ such that $u \neq d$
- $x_p^k = \bar{x}_p^k, \forall k \in K \setminus K(d)$.

Algorithm 2 In-tree Neighborhood Search

Require: an initial load plan $(\bar{y}, \bar{x}, \bar{\tau})$

for each terminal d **do**

 Set $F_d = \sum_{k \in K(d)} q_k$, the total amount of freight destined for d

end for

Set $TERMS =$ array of top 25% of terminals with respect to F_d

Set $N = |TERMS|$

Sort $TERMS$ in descending order of F_d

Set $iter = 0$

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

 Choose destination terminal $d = TERMS[iter \bmod N]$

 Solve IIP_d

if Solution gives lower total load plan cost **then**

 Update $(\bar{y}, \bar{x}, \bar{\tau})$

end if

 Set $iter = iter + 1$

end while

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

Note that we never fix the trailer variables τ_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 success of Algorithm 2 depends on the time needed at each iteration to solve in-tree IPs. We discuss techniques for reducing those solution times, i.e., preprocessing and cut generation, and study their effectiveness computationally in Appendix B.

4.4 Variations on Traditional Load Plan Design

The in-tree requirement of load plans is enforced to simplify terminal operations, as 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. We next present integer programs for each of these variants and how they form the foundation of the integer program solved by Algorithm 2. One of the strengths of a solution approach such as Algorithm 2 is that considering these relaxations requires only changing the integer program solved at each iteration.

- **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 the IP by redefining the variables that indicate what directs are chosen and by slightly modifying the constraints (2) and (3). 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 $a \in A$ (i.e., the day of dispatch).

We replace the variables y_l^d in the in-tree IP with variables y_l^{md} that indicate whether direct $l \in \Delta^+(u)$ is chosen for freight destined for terminal d at terminal u on day m , *i.e.*,

$$y_l^{md} \in \{0, 1\} \quad \forall d \in U, \quad \forall m \in \text{Days}, \quad \forall l \in \Delta^+(u), u \in U.$$

The day-differentiated analog of 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.*,

$$\sum_{l \in \Delta^+(u)} y_l^{md} \leq 1 \quad \forall u \in U, \quad \forall m \in \text{Days}, \quad \forall d \in U \quad (7)$$

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

$$\sum_{p \in P(k): a \in p} x_p^k \leq y_{l(a)}^{m(a)d(k)} \quad \forall k \in K, \forall a \in A \quad (8)$$

The same comments regarding aggregating the coupling constraints (3) applies to the constraints (8).

When solving restrictions of the IP to re-route freight to terminal d in the context of executing Algorithm 2, we fix variables $y_l^{mu} = \bar{y}_l^{mu} \quad \forall u \in U$ such that $u \neq d$ and $x_p^k = \bar{x}_p^k, \forall k \in K \setminus K(d)$.

- **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 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 $K(o, d) \subseteq K, o \in U, d \in U$ denote the set of commodities with origin terminal o and destination terminal d . Let $P(o, d)$ denote the set of paths of directs between terminals $o, d \in U$. Furthermore, let $P(p, k)$ denote the set of timed-copies of a path of directs p for commodity k . Then, the following *same-path inequalities* need to be satisfied:

$$\sum_{p \in P(p, k_1)} x_p^{k_1} = \sum_{p \in P(p, k_2)} x_p^{k_2} \quad \forall o, d \in U, \quad \forall k_1, k_2 \in K(o, d), \quad \forall p \in P(o, d) \quad (9)$$

When solving restrictions of the IP to re-route freight to terminal d in the context of executing Algorithm 2, we fix variables $x_p^k = \bar{x}_p^k, \forall k \in K \setminus K(d)$.

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

When solving restrictions of the IP to re-route freight to terminal d in the context of executing Algorithm 2, we fix variables $x_p^k = \bar{x}_p^k, \forall k \in K \setminus K(d)$.

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 .

5 Computational Results

The algorithm was developed in C++ with CPLEX 11 as the Mixed Integer Program (MIP) solver with which we 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 .1%. We include the aggregated constraints that ensure the proper directs are chosen for the paths chosen for each commodity and include the disaggregated constraints as CPLEX *user cuts*. All experiments were run on a Debian Linux computer with 32 GB of RAM and 8 Intel Xeon 2.6 GHz processors and times are always reported in seconds.

Using a planning horizon of a week for load planning is 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). The linehaul network of the carrier has about 60 BBs, 100 EOLs, and 16,000 directs. We note that the structure of the carrier’s network is slightly different than that of a traditional carrier, in that over a third of the terminals are BBs. However, many of these BBs are so-called “transfer points” with limited capacity for handling and transferring freight; mostly carried out by drivers. 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. When creating these instances, we allowed at most 50 paths in LN for each (origin terminal, destination terminal) pair. After creating timed copies of these paths to represent holding freight at a terminal, we can have over 100 paths for each commodity. The time-space networks have about 5,000 nodes, 500,000 arcs, 40,000 commodities, and 3,000,000 paths. Clearly, the resulting integer programs cannot be solved directly.

Our initial computational experiments were geared towards configuring the process of solving the restricted IPs. Based on the results of these experiments, we found that incorporating two classes of valid inequalities directly into the formulation, adding one class of valid inequalities to the formulation on the fly, and limiting the solution time to 90 seconds provided the best balance between quality and efficiency. More details can be found in Appendix B.

The primary goal of our research is to develop technology that can produce more cost-effective load plans. In Table 2, we report the savings (“ $\Delta Cost$ ”) and the increase in pounds per trailer (“ ΔPPT ”) when we run our IP-based neighborhood search algorithm on the April 2008 and March 2009 instances for six hours, in percentages relative to the initial load plan provided by the carrier.

Table 2: Percentage Savings

	Apr08-W1	Apr08-W2	Apr08-W3	Mar09-W1	Mar09-W2	Mar09-W3	Mar09-W4
$\Delta Cost$	4.49	3.94	4.77	3.73	3.29	3.69	3.77
ΔPPT	2.17	1.37	3.13	2.37	1.68	1.66	1.27

Even if only portion of these savings can be realized in practice, it will have a noticeable impact on 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. We believe that this can be attributed to the carrier’s load plan yielding a higher pounds per trailer in March 2009 than in April 2008. A more detailed analysis reveals that most of the savings can be attributed

to reduced empty trailer repositioning costs.

Having established that IP-based neighborhood search is effective in producing high-quality traditional load plans, we next explore the benefits of relaxing the requirements of traditional load plans that there is a single outbound direct to each destination, and that this single outbound direct for each destination is the same every day of the week. Thus, the variants that we consider drop either one or both of these restrictions, i.e., a day-differentiated load plan ensures a single outbound direct to each destination, but the outbound direct can vary by day of the week, a same-path load plan allows more than one outbound direct to a destination, depending on the origin of the freight, but ensures that same outbound directs are used every day of the week, and an unrestricted load plan, allows more than one outbound direct to a destination, depending on the origin of the freight, and allows these outbound directs to vary by day of week. We run the different versions of our IP-based neighborhood search algorithm for six hours on each of the seven instances to study the impact. The results are reported in Table 3, where savings are again computed relative to the cost of the carrier’s load plan for the relevant week.

Table 3: Percentage Savings For Variants

		Traditional	Day-Differentiated	Same-Path	Unrestricted
Apr08-W1	$\Delta Cost$	4.49	6.08	4.93	7.41
	ΔPPT	2.17	3.25	2.50	3.47
Apr08-W2	$\Delta Cost$	3.94	6.11	4.90	7.62
	ΔPPT	1.37	2.77	2.20	4.60
Apr08-W3	$\Delta Cost$	4.77	6.93	5.14	8.05
	ΔPPT	3.13	4.77	3.40	5.81
Mar09-W1	$\Delta Cost$	3.73	6.51	4.01	7.60
	ΔPPT	2.37	4.88	2.60	5.78
Mar09-W2	$\Delta Cost$	3.29	6.02	3.44	7.22
	ΔPPT	1.68	3.90	1.91	5.06
Mar09-W3	$\Delta Cost$	3.69	6.51	3.85	7.60
	ΔPPT	1.66	4.24	1.79	5.02
Mar09-W4	$\Delta Cost$	3.77	6.76	3.83	7.89
	ΔPPT	1.27	4.07	1.59	5.18

Not surprisingly, the unrestricted load plans lead to the greatest savings. Note that while day-differentiated load plans lead to significantly greater savings than traditional load plans, and that same-path load plans do not. A more detailed analysis again reveals that most of the savings can be attributed to reduced empty trailer repositioning costs and that the difference in performance among the different variants can also be attributed to reduced empty trailer repositioning costs, i.e., day-differentiated and unrestricted load plans have substantially smaller empty trailer repositioning costs than traditional and same-path load plans.

In day-differentiated and unrestricted load plans, the origin-destination path for a commodity is allowed to vary by day of week. Therefore, we present in Table 4 how many (different) origin-destination paths are used for an origin, destination pair during a week. More specifically, we report what percentage of origin, destination pairs use 1, 2, 3, 4, or 5 paths during a week.

We see that for the day-differentiated load plans, the majority of origin, destination pairs, about 75%, still use the same path each day of the week, while about 19% of origin, destination pairs send freight along two path during the week. As a result, only very few origin, destination pairs send freight along three or more paths. This is encouraging since it implies that a day-differentiated load plan does not represent a significant shift from how freight is routed compared to a traditional load plan. Even for unrestricted load plans, the vast majority of origin, destination pairs still use only one or two paths during the week.

A slightly different perspective is to consider outbound directs at a terminal. In day-differentiated

Table 4: Percentage Different Paths on Different Days

# Paths	Apr08-W1		Apr08-W2		Apr08-W3	
	DD	U	DD	U	DD	U
1	79.57	75.35	80.40	75.76	77.78	74.95
2	17.06	20.31	16.94	19.86	18.47	20.08
3	3.12	3.79	2.43	3.99	3.28	4.24
4	0.25	0.53	0.21	0.34	0.42	0.70
5	0.00	0.02	0.02	0.04	0.04	0.04

# Paths	Mar09-W1		Mar09-W2		Mar09-W3		Mar09-W4	
	DD	U	DD	U	DD	U	DD	U
1	75.16	66.11	73.94	65.65	73.37	65.52	71.36	63.33
2	19.69	25.13	20.59	25.68	21.07	26.36	22.09	27.03
3	4.39	7.38	4.85	7.32	4.86	6.86	5.70	7.98
4	0.73	1.30	0.60	1.19	0.65	1.20	0.76	1.57
5	0.04	0.09	0.02	0.16	0.04	0.05	0.09	0.09

and unrestricted load plans, the outbound direct for a destination is allowed to vary by day of week. As this impacts terminal operations, it is relevant to study the impact of this flexibility in more detail. Therefore, we present in Table 5 how many (different) outbound directs are used for a terminal, destination pair during a week. More specifically, we report what percentage of terminal, destination pairs use 1, 2, 3, 4, 5, or 6 or more outbound directs during a week.

Table 5: Percentage Different Directs on Different Days

# Outbound Directs	Apr08-W1		Apr08-W3		Apr08-W3	
	DD	U	DD	U	DD	U
1	78.84	74.64	80.06	75.43	77.38	74.50
2	17.92	20.71	17.22	19.72	19.03	20.01
3	3.00	3.91	2.57	4.27	3.17	4.67
4	0.21	0.69	0.13	0.49	0.38	0.73
5	0.04	0.02	0.02	0.06	0.04	0.08
6+	0.00	0.02	0.00	0.02	0.00	0.00

# Outbound Directs	Mar09-W1		Mar09-W2		Mar09-W3		Mar09-W4	
	DD	U	DD	U	DD	U	DD	U
1	75.81	66.69	74.30	65.61	73.86	65.48	77.38	74.50
2	19.32	24.59	20.80	25.85	20.96	26.48	19.03	20.01
3	4.22	7.18	4.41	6.99	4.50	6.86	3.17	4.67
4	0.62	1.32	0.47	1.39	0.65	1.07	0.38	0.73
5	0.04	0.18	0.02	0.16	0.04	0.11	0.04	0.08
6+	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00

We see that the day-differentiated and unrestricted load plans are such that for the vast majority of terminal, destination pairs the same outbound direct is used during the week and if more than one outbound directs is used during the week, it is rarely are more than two. This is not completely surprising, since using a single outbound direct for freight destined for a specific terminal is a form of consolidation. More importantly, however, from an implementation perspective, this “profile” implies that the impact on terminal operations is relatively small, which is desirable.

6 Conclusions and Future Research

We have developed integer programming based load plan construction technology that is specifically designed to handle the tight service standards that LTL carriers have to offer to remain competitive in today’s environment. Computational experiments indicate that the technology is

effective and has the potential to deliver substantial savings to LTL carriers. We have also investigated variations of traditional load plan designs, which are, or will soon be, implementable in practice due to advances in enabling technology, for example day-differentiated load plans. Our computational experiments show that significant may result from adopting different load plan designs. Our efforts show that careful and detailed modeling of a complex environment and that developing customized solution approaches can deliver substantial benefits to an industry.

We believe that there is still potential to improve our technology for load plan design. For example by investigating different neighborhoods, i.e., different restrictions of the basic integer program. The current neighborhood has a system-wide perspective when seeking improvements for freight into a single destination. Initial experiments with a neighborhood that has a local perspective but seeks improvements for freight into many destinations, by attracting to or diverting freight from a particular direct are promising.

We are also exploring “real-time” load planning, in which a load plan is altered dynamically during the day more accurate information about the freight in the system becomes available. By deploying wireless communication devices in pickup and delivery operations, carriers have relatively early visibility of the freight that will enter the linehaul network. 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

- [Ahuja et al., 2002] Ahuja, R., Ergun, O., Orlin, J., and Punnen, A. (2002). A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics*, 123:75–102.
- [Archetti et al., 2008] Archetti, C., Speranza, M. G., and Savelsbergh, M. W. P. (2008). An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42:22–31.
- [Crainic, 2000] Crainic, T. (2000). Service network design in freight transportation. *European Journal of Operations Research*, 122(2):272–288.
- [Crainic and Gendreau, 2002] Crainic, T. and Gendreau, M. (2002). Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics*, 8:601–627.
- [Crainic et al., 2000] Crainic, T., Gendreau, M., and Farvolden, J. (2000). A simplex-based tabu search method for capacitated network design. *INFORMS J. Comput.*, 12:223–236.
- [Danna et al., 2005] Danna, E., Rothberg, E., and Le Pape, C. (2005). Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102:71–90.
- [De Franceschi et al., 2006] De Franceschi, R., Fischetti, M., and Toth, P. (2006). A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming B*, 105:471–499.
- [Dijkstra, 1959] Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.
- [Espinoza et al., 2008] Espinoza, D., Garcia, R., Goycoolea, M., Nemhauser, G., and Savelsbergh, M. (2008). Per-seat, on-demand air transportation part II: Parallel local search. *Transportation Science*, 42(3):279–291.

- [Ghamlouch et al., 2003] Ghamlouch, I., Crainic, T., and Gendreau, M. (2003). Cycle-based neighborhoods for fixed charge capacitated multicommodity network design. *Operations Research*, 51:655–667.
- [Ghamlouch et al., 2004] Ghamlouch, I., Crainic, T., and Gendreau, M. (2004). Path relinking, cycle-based neighborhoods and capacitated multicommodity network design. *Annals of Operations Research*, 131:109–133.
- [Hewitt et al., 2010] Hewitt, M., Nemhauser, G., and Savelsbergh, M. (2010). Combining Exact and Heuristic Approaches for the Capacitated Fixed-Charge Network Flow Problem. *INFORMS Journal on Computing*, 22(2):314–325.
- [Jarrah et al., 2009] Jarrah, A., Johnson, E., and Neubert, L. (2009). Large-scale, less-than-truckload service network design. *Operations Research*, 57(3).
- [Pederson et al., 2009] Pederson, M. B., Crainic, T. G., and Madsen, O. B. (2009). Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43(2).
- [Powell and Farvolden, 1994] Powell, W. and Farvolden, J. (1994). Subgradient methods for the service network design problem. *Transportation Science*, 28:256–272.
- [Powell, 1986] Powell, W. B. (1986). A local improvement heuristic for the design of less-than-truckload motor carrier networks. *Transportation Science*, 20(4):246–257.
- [Powell and Koskosidis, 1992] Powell, W. B. and Koskosidis, I. A. (1992). Shipment routing algorithms with tree constraints. *Transportation Science*, 26(3):230–245.
- [Powell and Sheffi, 1983] Powell, W. B. and Sheffi, Y. (1983). The load planning problem of motor carriers: Problem description and proposed solution approach. *Transportation Research-A*, 17A(6):471–480.
- [Powell and Sheffi, 1989] Powell, W. B. and Sheffi, Y. (1989). Design and Implementation of an Interactive Optimization System for Network Design in the Motor Carrier Industry. *Operations Research*, 37:12–29.
- [Savelsbergh and Song, 2008] Savelsbergh, M. and Song, J.-H. (2008). An optimization algorithm for inventory routing with continuous moves. *Computers and Operations Research*, 35:2266–2282.
- [Schmid et al., 2008] Schmid, V., Doerner, K. F., Hartl, R. F., Savelsbergh, M. W. P., and Stoecher, W. (2008). A hybrid solution approach for ready-mixed concrete delivery. *Transportation Science*, To appear.
- [Wieberneit, 2008] Wieberneit, N. (2008). Service network design for freight transportation: a review. *OR Spectrum*, 30(1).
- [Zhang, 2010] Zhang, Y. (2010). *Advances in LTL Load Plan Design*. PhD thesis, Georgia Institute of Technology.

Appendices

A Choosing an appropriate time discretization

One of the advantages of our approach is the use of a time discretization that is appropriate for the service oriented environment in which LTL carriers have to operate. Specifically, by modeling time at a fine granularity, we accurately capture when consolidation occurs and the impact of freight routing decisions. Here, we provide a more in-depth discussion of the impact a coarse time discretization has on a model’s estimation of when consolidation occurs and what trailer utilizations are achieved.

Consider the following two paths. The first for freight originating in Louisville, KY destined for Atlanta, GA:

- 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.

and the second for freight originating in Nashville, TN and also destined for Atlanta, GA:

- Travel for 5 hours from Nashville, TN to Atlanta, GA.

Now consider a time-space model that discretizes time with a single node for each terminal and each day, and arcs that only move forward in time. Given such a network, how can one model the path given above for the Louisville → Atlanta freight for the purpose of measuring trailer utilization? One could map the path above as depicted in Figure 11, dispatching from Louisville on Monday and from Nashville on Tuesday, and the freight arriving in Atlanta a day late. Note that this mapping creates a logical path from Louisville through Nashville to Atlanta, but the consolidation opportunity with the Nashville → Atlanta freight originating on Monday is lost. (Note that if origin-destination freight flow demands are regular throughout the week, then there is Nashville → Atlanta freight originating on Tuesday with which the Louisville → Atlanta freight could be consolidated.)

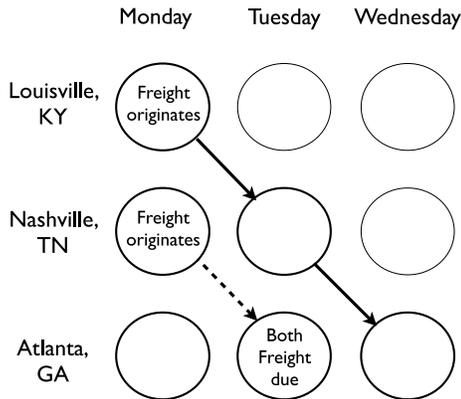


Figure 11: Freight Arrives Late

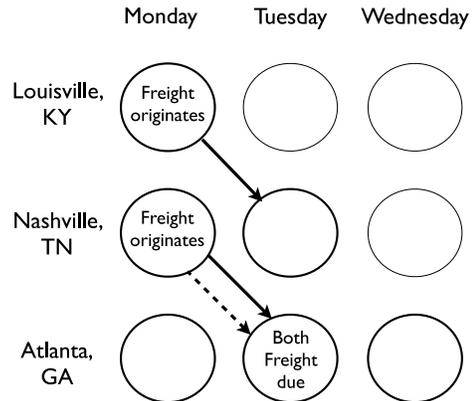


Figure 12: Freight Travels in Two Trailers At Same Time

Or, one could map the path as depicted in Figure 12, simultaneously dispatching from both Louisville and Nashville on Monday, with the same freight traveling in two trailers, on two different

lanes, at once. Note that this mapping preserves the consolidation opportunity with the Nashville \rightarrow Atlanta, but it also creates consolidation opportunities that do not exist in practice. Consider freight originating in Nashville, TN on Monday at 7 pm that is destined for Jacksonville, FL at 8 am the next day and that follows the path:

- Travel for 4.5 hours from Nashville, TN to Atlanta, GA.
- Handling for 2 hours in Atlanta, GA.
- Travel for 5.5 hours from Atlanta, GA to Jacksonville, FL.

With the mapping that dispatches the Louisville \rightarrow Atlanta freight from Nashville on Monday, both the Louisville \rightarrow Atlanta and the Nashville \rightarrow Jacksonville freight can be dispatched from Nashville in the same trailer. However, to make service, the Nashville \rightarrow Jacksonville freight must depart Nashville no later than 8 pm on Monday, while the earliest the Louisville \rightarrow Atlanta freight can depart from Nashville is 11:30 pm. This scenario is illustrated in Figure 13.

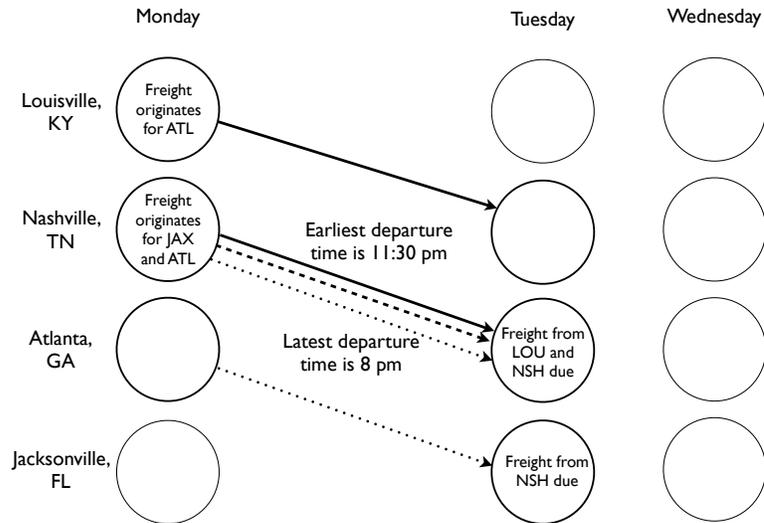


Figure 13: Overestimating Consolidation Opportunities

By modeling time at a finer level of detail, consolidation opportunities are more accurately captured. Realistic times for each dispatch in each of these paths can be modeled so that the Nashville \rightarrow Atlanta and Louisville \rightarrow Atlanta freight may be consolidated at Nashville into a common trailer (or trailers) outbound to Atlanta, with less chance of creating phantom consolidation opportunities.

We have investigated computationally the impact of the level of discretization used. Specifically, we solved the integer program that constructs a load plan (discussed in Section 4.2) using one week of freight originating from and destined to terminals in Alabama, Florida, Georgia, South Carolina, and Tennessee, and did so for three levels of discretization: (1) One time point per terminal per day (One), (2) Two time points per terminal per day (Two), and, (3) The discretization employed by our technology (Multi). In the first column of Table 6 (% Simultaneous Dispatch), we show the percentage of paths appearing in a large U.S. carrier’s load plan that, as in Figure 11, require simultaneous dispatches similar to what is shown in Figure 12. Next, we analyze what happens if we accommodate such paths by using a mapping that preserves path integrity, but creates unrealistic dispatch times (i.e., the mapping illustrated in Figure 11), and then optimize the load plan. In the second column of Table 6 (Model % Savings), we present the savings reported by the optimization.

Table 6: Effect of Different Discretizations

	% Simultaneous Dispatch	Model % Savings	Estimated % Savings
One	52.50	5.45	1.24
Two	40.50	4.99	1.73
Multi	0.00	4.45	3.79

Finally, in the third column of Table 6 (Estimated % Savings), we present an accurate estimate of the savings by evaluating the load plan produced by the optimization using the technology described in [Zhang, 2010]. We see that the reported savings were the lowest when a finer discretization was used, but that the actual savings were the highest.

B Reducing the time to solve Intree IPs

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, second, and last terminals in a path \mathbf{p} of directs be denoted by $f(\mathbf{p})$, $s(\mathbf{p})$, and $l(\mathbf{p})$ respectively. Then, we have the following valid path-continuation inequalities

$$\sum_{k' \in K(f(\mathbf{p}), l(\mathbf{p}))} \sum_{p \in P(\mathbf{p}, k')} x_p^{k'} \leq \frac{|K(f(\mathbf{p}), l(\mathbf{p}))|}{|K(s(\mathbf{p}), l(\mathbf{p}))|} \sum_{k' \in K(s(\mathbf{p}), l(\mathbf{p}))} \sum_{p \in P(\mathbf{p} \setminus f(\mathbf{p}), k')} x_p^{k'} \quad \forall \mathbf{p} \in \cup_{k \in K} P(k). \quad (10)$$

Note that while we have presented the path-continuation inequalities in the context of solving an In-tree IP, they are also valid when solving restrictions, as is done in Algorithm 2. Also, there are analogous inequalities for the In-tree IP solved to perform Day-Differentiated Load Plan Design.

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

$$\sum_{k \in K(d)} \sum_{p \in P(k): a \in p} q_k x_p^k + f_a \leq \tau_a \quad \forall a \in A. \quad (11)$$

We first observe that when solving IIP_d , we can bound the variable τ_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

$$\tau_a \geq \lceil f_a \rceil + (\lceil f_a + q_k \rceil - \lceil f_a \rceil) \sum_{p \in P(k): a \in p} x_p^k \quad \forall a \in A, \quad \forall k \in K. \quad (12)$$

Because 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 an 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. Because 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. Because the size of the time-space network can get large quickly for real-life 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.

We next investigate the benefits derived from the three classes of valid inequalities, *i.e.*, trailer disaggregate inequalities (TD), path-continuation inequalities (PC), and same-path inequalities (SP). 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 40 IIP_d instances, each associated with a different destination terminal d and based on freight volumes from the instance Apr08-W1. We note that these restricted instances are significantly smaller than the unrestricted instances as, on average, they contain only 540 commodities and 30,000 paths. In the results presented below, the values are averaged over these 40 instances. Because we have the load plan that was used by the carrier for the week of the instances, 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 7, 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 7: Optimality Gap Reductions

	# Rows	Gap Reduction	Root Solve Time
No Inequalities	37,252	-	21.35
TD	37,438	1.02	21.67
TD + PC	39,400	4.71	23.50
TD + PC + SP	44,587	12.88	27.30

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 8 how many IIP_d instances can be solved to within .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 8: Number IIP_d Solved to within .1%

	Number Solved	Avg. Time	Avg. Time*
No Inequalities	27	46.41	46.41
TD	27	40.74	40.74
TD + PC	29	38.97	35.44
TD + PC + SP	30	48.10	40.89

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 9 the savings that are obtained when running the neighborhood search for thirty minutes, again allowing three minutes to solve each instance of IIP_d , where savings are defined as the (cost of the initial load plan minus the cost of the load plan designed by the algorithm) divided by the cost of the initial load plan.

Table 9: % Savings in thirty minutes

No Inequalities	TD	TD + PC	TD + PC + SP
1.54	1.81	2.01	1.82

We see that again the Path-continuation and Trailer Disaggregate inequalities yield the best performance; enabling nearly .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, and the same-path inequalities were included as CPLEX *user cuts* in all experiments in Section 5.

Lastly, we note that 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 .52% for the instances used to produce the results reported in Table 7. 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 7, 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 .65%. Hence, we see that even though our 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.