

A Novel Modeling Approach for Express Package Carrier Planning

Sarah Root, Amy Cohn

Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109

Received 21 February 2007; revised 31 May 2008; accepted 7 June 2008

DOI 10.1002/nav.20310

Published online 25 August 2008 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Express package carrier networks have large numbers of heavily-interconnected and tightly-constrained resources, making the planning process difficult. A decision made in one area of the network can impact virtually any other area as well. Mathematical programming therefore seems like a logical approach to solving such problems, taking into account all of these interactions. The tight time windows and nonlinear cost functions of these systems, however, often make traditional approaches such as multicommodity flow formulations intractable. This is due to both the large number of constraints and the weakness of the linear programming (LP) relaxations arising in these formulations. To overcome these obstacles, we propose a model in which variables represent combinations of loads and their corresponding routings, rather than assigning individual loads to individual arcs in the network. In doing so, we incorporate much of the problem complexity implicitly within the variable definition, rather than explicitly within the constraints. This approach enables us to linearize the cost structure, strengthen the LP relaxation of the formulation, and drastically reduce the number of constraints. In addition, it greatly facilitates the inclusion of other stages of the (typically decomposed) planning process. We show how the use of *templates*, in place of traditional delayed column generation, allows us to identify promising candidate variables, ensuring high-quality solutions in reasonable run times while also enabling the inclusion of additional operational considerations that would be difficult if not impossible to capture in a traditional approach. Computational results are presented using data from a major international package carrier. © 2008 Wiley Periodicals, Inc. *Naval Research Logistics* 55: 670–683, 2008

Keywords: freight planning; large scale optimization; integer programming

1. INTRODUCTION

In the freight transportation industry, effective network planning is critical to success but often presents tremendous computational challenges. In this article, we show how a novel modeling approach (which embeds much of the system complexity within the variable definition, rather than capturing it through constraints), in tandem with the use of *templates* to construct promising candidate variables, can enable us to address many of these challenges within express package carrier planning. We believe this approach will have broader applicability in other areas of freight transportation planning as well.

Freight transportation is critical in moving virtually all of the products we use in our daily lives, from the food that we eat to the medical supplies that help keep us healthy to the computers that we use. In addition, raw materials such as coal, oil, and lumber used to generate energy and manufacture goods move through the transportation network. The amount of freight moved each year is staggering - in 2002,

more than 19 billion tons valued at \$13 trillion were moved in the United States [42].

High-quality network planning in this industry is important—consistent productivity gains are credited with improving overall productivity in the U.S. and directly increasing the nation's GDP [43]. Such planning can be quite challenging, however, for a number of reasons. First, there are several inter-dependent resources (packages, trailers, tractors, drivers) that must be considered. Second, freight transportation problems often have nonlinear cost structures. Third, many of the freight transportation problems that arise in practice are quite large. All of these characteristics lead to significant computational challenges.

Although there is an inherent network structure underlying most freight transportation problems, traditional network flow models such as multicommodity flow (MCF) are often intractable. In particular, moving a nontrivial number of commodities through a network gives rise to a prohibitively large number of constraints and variables. In addition, linearizing the cost structure can lead to intolerably weak linear programming (LP) relaxations.

In this article, we consider three planning problems from the express package industry—one of the fastest growing

Correspondence to: A. Cohn (amycohn@umich.edu)

segments in the freight transportation industry [42]. We begin by considering the load matching and routing problem (LMR) and demonstrate why a traditional MCF formulation is inadequate for solving this problem. We present an alternative formulation which embeds complexity within the variable definition, dramatically decreasing the number of constraints and simultaneously improving the strength of the LP relaxation. We also demonstrate how predefined templates can be used to identify promising candidate variables, in place of the use of delayed column generation.

We then focus on how to extend the LMR model to also include two additional problems, *trailer assignment* and *empty balancing*, simultaneously integrating three problems that would otherwise have to be solved sequentially. We demonstrate how this integration requires only minor changes to the formulation, with most of the changes being embedded in the variable definition. We provide computational examples to investigate both computational performance and solution quality, and discuss the trade-offs in run time and objective value between the integrated and disaggregated approaches.

The contributions of our work are threefold. First, we present a method for solving a challenging real-world problem (LMR) that is intractable under a traditional modeling approach. Second, we show how this approach facilitates *integrated planning*—we are able to solve three stages of the express package carrier planning process simultaneously, for improved solution quality. Finally, we suggest that the variable definition presented here will be applicable to other freight transportation planning problems as well.

The outline of the article is as follows. In Section 2, we give an overview of the planning process for express package carriers and introduce the load matching and routing problem. In Section 3, we explain why traditional modeling approaches are intractable for LMR and present an alternative *cluster-based* model, which we solve with the use of templates. Computational results based on data from a prominent express package carrier are presented. Next, Section 4 focuses on incorporating the trailer assignment and empty balancing problems in tandem with load matching and routing. Computational results for this integrated planning problem are given, as well as a discussion of the trade-offs between run-time and solution quality. Finally, in Section 5, we conclude by suggesting future areas of research.

2. PROBLEM DESCRIPTION AND LITERATURE REVIEW

Express package carriers deliver millions of packages each day [44]. Delivering all of these packages from origin to destination within their time windows gives rise to an elaborate planning process. The number of origin/destination (*O/D*)

pairs in the network is quite large, and the number of packages traveling between the majority of these *O/D* pairs does not justify a dedicated truck. Therefore, packages are instead routed through a series of intermediate facilities in the network. At each of these facilities, called *hubs* or *consolidation centers*, packages are sorted and then consolidated with other packages that can travel together to more efficiently utilize system resources.

The network planning hierarchy is outlined in Fig. 1a. It demonstrates four stages of the planning process for a single package, traveling from Port Clinton, OH (PC) to Beaumont, TX (BEA). First, the package's *operational routing* must be determined. In this case, the package will travel in one load from PC to Toledo, OH (TOL). This load will be broken down and sorted in TOL, and the package will then move on a second load from TOL to Houston, TX (HOU). Again, the load will be broken and sorted, and the package assigned to its final load, from HOU to BEA. The problem of assigning *O/D* pairs to routings is known as the *load planning* or *package routing* problem. [Note that all packages sharing a common origin and destination follow the same routing, for operational simplicity.] Operational routings are required to be time feasible and to respect capacity limitations at the intermediate sorting facilities.

After the load planning problem has been solved, it is easy to calculate the *volume* (total number) of packages flowing between any pair of *sorts* in the network (where a sort is defined by a fixed operational time period in addition to a location). This volume must then be assigned to one of two trailer types, each of which has a different capacity. This stage in the planning process is referred to as the *trailer assignment* problem (see Fig. 1b), and determines the set of *loads* to be moved through the network, where a load is defined by an origin, destination, set of packages, trailer type, earliest departure time, and latest arrival time. The earliest departure of a load is determined by the latest available time of all of its packages, and the latest arrival time is the earliest due date.

A load which has been assigned to the longer of the two trailer types (referred to as a *van*) typically must travel individually behind a tractor. However, two loads each assigned to the shorter trailer type (referred to as a *pup*) can often be combined and share a single tractor. It is significantly cheaper to pull two trailers behind a single tractor as part of a *double trailer configuration* than it is to pull each trailer individually because only a single driver is required, less fuel is used, and fewer tractors are needed. This cost savings may even justify having loads travel circuitous mileage so that they can be matched with other loads. In our example, the TOL-HOU load is routed through Cincinnati (CIN) so it can be matched with a CIN-HOU load (see Fig. 1c). The process of determining which loads should be matched together and how these matched loads should be routed is known as the *load matching and routing problem*.

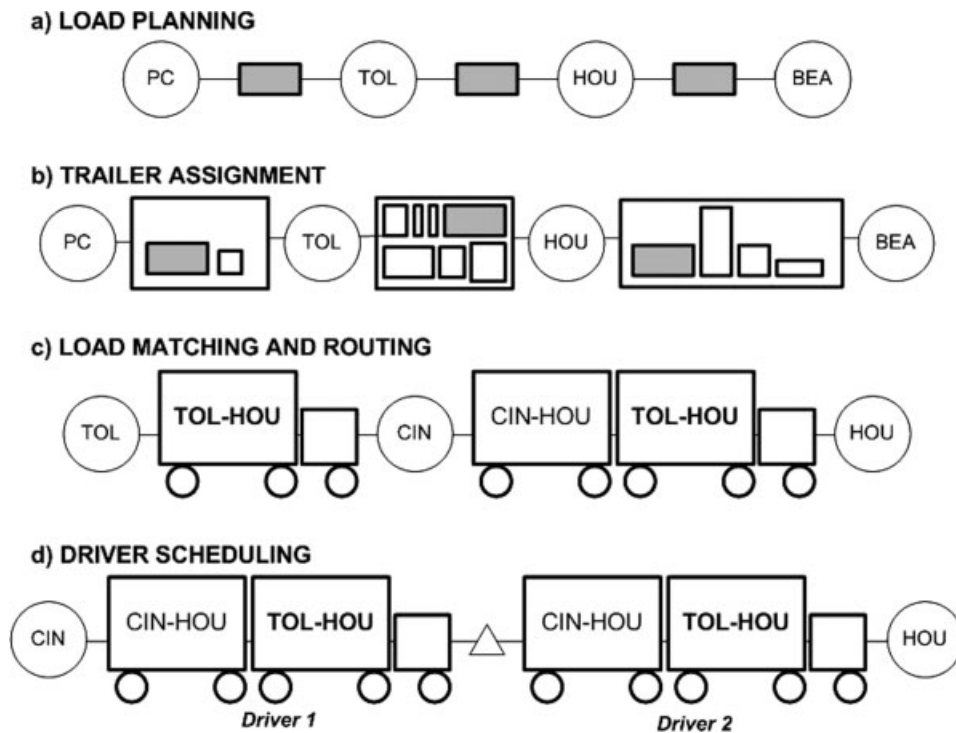


Figure 1. Network planning hierarchy for express package carriers.

Next, observe that delivering all loads from origin to destination may give rise to *imbalance* in the network. For instance, if there are more loads inbound to a node than outbound, then that node will begin to accumulate empty trailers. In the *empty balancing* stage of the planning process, empty trailers must be redistributed (by type) from surplus nodes to deficit nodes. Because the number of trailers available in the network is typically not a limiting factor, the timing of rebalancing is far less restrictive for empty trailers than for loads—it is sufficient to balance trailers over the duration of the planning process, rather than on an hourly or even daily basis.

Finally, drivers must be assigned to move all of the loaded and empty trailers through the network. This is referred to as *driver scheduling*. The drivers themselves have their own complex constraints arising from labor negotiations, company policies, and federal safety regulations. As a result, a load may be covered by several different drivers between origin and destination. For instance, because the trip from CIN to HOU is quite long, this movement may be broken down into two segments, with Driver One completing the first segment and then handing off the loads to Driver Two, who completes the second segment. The driver scheduling problem must consider both how to split the movement of each load into smaller segments and how to assign individual drivers to these segments.

Although it would be desirable to consider the entire planning process simultaneously to capture the interactions

between different levels, a global approach is intractable due to the complexity of the individual problems as well as their combined size. Instead, the planning process is typically decomposed into the five stages described above. Each stage is considered individually, and the output of one problem becomes the input of the next. This myopic view makes the planning problem more tractable but does not take into account the impact that decisions made in one stage of the planning process have on subsequent stages.

In this article, we first explore the load matching and routing problem, explaining how an approach which embeds much of the complexity within the variable definition rather than the constraints allows us to solve this challenging problem. We then focus on how this approach enables us to expand the scope of the problem, incorporating the trailer assignment and empty balancing problems concurrently with load matching and routing.

To the best of our knowledge, the only other study of LMR in the literature is our prior work [18]. In this article, we extend this prior work by including the additional problem of trailer assignment into the planning process. We also quantify the value of integrating the three planning problems and suggest where an integrated planning approach is likely to demonstrate substantial improvements over a sequential approach. In this article, we also demonstrate how rounding the solution obtained by a relaxation of the integrated problem allows us to solve much larger problem instances than

those solved in our prior work without any significant impact on the quality of the solutions obtained.

In addition to our research, many topics related to LMR have been well studied. Down-stream decisions in express package carrier networks are considered in the thesis of [40]. Papers addressing nonlinear cost functions in transportation planning problems include [9, 27, 31], and [33]. The issue of time windows is considered in related application areas such as vehicle routing, airline applications, and shipping [10, 11, 19, 35, 38]. More broadly, there is a vast body of literature addressing freight transportation problems closely related to LMR. For example, the matching of loaded and empty trailers in a hub-and-spoke network is considered in [24]. Ref. [8] also consider load matching in a less-than-truckload (LTL) network. Heuristics for building loads in the LTL industry are considered in [2]. The work of [23] in LTL planning discusses its hierarchical nature and the need for decomposition in the planning process. Dynamic aspects of freight transportation are discussed in [15]. Routing in a hub-and-spoke network, in conjunction with facility location, is addressed in [12]. Facility location is also considered by [32]. Finally, surveys of optimization models in freight transportation appear in [20] and [21].

3. SOLVING LMR

The objective of the load matching and routing problem is to find the least expensive way to move a set of loads through a network, taking advantage of the cost savings of double trailer configurations (possibly through the use of circuitous routing) while ensuring that all time windows are adhered to.

In our discussion of this problem, we initially assume, for the sake of exposition, that all loads are assigned to pups and that pups can be matched together as part of a double trailer configuration on all arcs in the network. We relax these assumptions in Section 4.

3.1. Arc-based MCF Formulation

Given the network structure underlying LMR, it is natural to formulate this problem as a variation of the multi-commodity flow problem [1, 14, 26, 30]. In an *arc-based* formulation of MCF, the variables represent the flow of commodities on each arc. Two sets of constraints apply to these variables. First, for each node/commodity pair, flow must be *balanced*—supply plus flow in must equal demand plus flow out. Second, the sum of flows across all commodities on a given arc cannot exceed the capacity of that arc. [Note that in our problem, arc capacities are infinite.] To model LMR as a variation of MCF, with commodities corresponding to loads that share both an *O/D* and a time window, two key changes must be made.

First, in order to satisfy the time-window requirements, a *time-space network* must be used. In such a network, each node corresponds not only to a facility f , but to a point in time t as well. An arc between $\{f_1, t_1\}$ and $\{f_2, t_2\}$ represents the flow of a commodity from facility f_1 to facility f_2 , leaving at time t_1 and arriving at time t_2 . The use of a time-space network enables the timing constraints to be captured while maintaining the pure MCF structure, but at the expense of a much larger network, since each facility is represented by not just one node but rather by one node per point in time.

The second change that must be made is to linearize the nonlinear objective function. [Recall that the cost of an even number e of trailers is the cost of $\frac{e}{2}$ double configurations, while the cost of an odd number o of trailers is the cost of $\lfloor \frac{o}{2} \rfloor$ double configurations plus one single configuration.] To linearize the objective function, the MCF structure can be modified by introducing two new sets of integer variables, one to represent the number of single trailer configurations on each arc and one to represent the number of double trailer configurations. These trailer configurations, rather than the actual commodity flows, are used in the objective function. [Additional details can be found in [18]].

This formulation is problematic for two important reasons. First, even modestly-sized problem instances become quite large after the introduction of the time-space network. For instance, assuming a node every five minutes, a problem with just 10 nodes, 50 arcs, and 20 commodities would have more than 300,000 variables and 70,000 constraints. Second, the cost structure naturally lends itself to very fractional solutions, because it is less costly for trailers to move as half of a double trailer configuration than to move as a single trailer. Whenever there is an odd number of loads flowing on an arc (i, j) , the fractional solution will assign the “odd” load to half of a double configuration rather than to the more costly single configuration. This gives rise to very large branch-and-bound trees in which each node of the tree is itself a prohibitively large LP, because of the time-space network. Thus, the arc-based MCF formulation is intractable for all but trivial instances of LMR.

3.2. Path-based MCF Formulation

Even without the special considerations introduced by LMR, the large number of balance constraints encountered whenever using an arc-based formulation of MCF can limit tractability. As an alternative, MCF problems are often modeled using a *path-based* formulation. In this approach, variables represent the flow of a commodity over an entire path from origin to destination, rather than just over a single arc [1]. Such a formulation eliminates the need for the large set of constraints required to ensure node balance in the arc-based formulation, replacing it with a much smaller set of constraints (one per commodity) requiring that the flow on all

paths associated with that commodity sum to its demand. This formulation, which is typically solved via *column generation* due to its very large number of variables, can be derived from the arc-based formulation via *Dantzig-Wolfe* decomposition [1].

As with the arc-based formulation, the path-based formulation can be modified to accommodate LMR with two key changes. First, the underlying network from which the path variables are generated must be a time-space network, so as to ensure that each commodity satisfies its time window. The MCF network itself must also be a time-space network, to ensure that matched loads are time-compatible.

Second, the nonlinear cost structure must be addressed, as in the arc-based MCF formulation. Single and double trailer configurations must again be built from the commodities flowing on each arc, giving rise to the same single and double trailer variables and objective function as in the arc-based formulation. The option to move every trailer as half of a double configuration continues to exist on each arc for each commodity. Thus, the very weak LP relaxation remains and, with it, the corresponding intractability.

3.3. Cluster-based Formulation

Although converting from an arc-based formulation to a path-based formulation of MCF is not sufficient to achieve tractability for LMR, it nonetheless demonstrates the potential power of capturing problem complexity within the variable definition. By defining variables representing the flow of commodities over an entire path rather than just over a single arc, the very large set of balance constraints can be eliminated. We further extend this idea in our research, embedding additional complexity within the variable definition to overcome the challenges of time-windows and a non-linear cost function when solving LMR.

The idea of capturing complexity within the variable definition rather than through constraints has appeared in the literature as *composite variable modeling*, *variable redefinition*, *inverse projection*, and *extended reformulation*. Extensive discussions of these broad topics appear in [3, 4, 6, 13, 16–18, 22, 25, 28, 29, 34, 37, 39], with [7] and [41] focusing specifically on these issues for integer and mixed-integer programs.

While in some cases, models are formed via a decomposition or other modification of an initial, “traditional” model, in cases such as ours the modeling philosophy focuses on designing variables from the start expressly with the goal of addressing sources of intractability arising in more traditional formulations. In LMR, there are two primary sources of intractability. The first stems from the challenge of incorporating timing considerations and the second comes from the weak LP relaxation associated with linearizing the objective function. Two key observations assist us in constructing

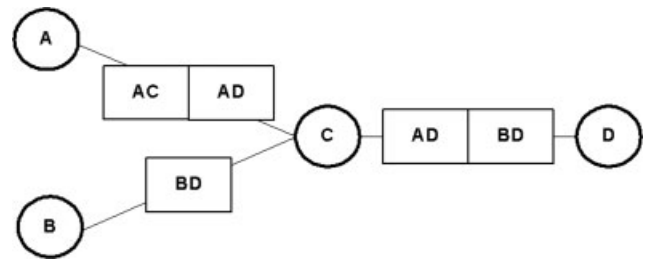


Figure 2. Sample cluster.

an alternative variable definition to address these challenges. First, many different feasible timings may be equivalent from a cost and feasibility perspective—all timings, in fact, in which the load is delivered on-time. For example, we are indifferent as to whether a load leaves as soon as it is available and arrives an hour early, leaves an hour after it is available and arrives promptly on time, or any time in between. The second observation of importance is that, whenever an arc does not have more than one load flowing over it, the opportunity for creating a double trailer configuration on this arc does not exist, except in a fractional solution.

Thus, we define variables that represent the *clustering of interacting loads*, rather than the flow of individual loads on arcs or paths through the network. Specifically, a *cluster* is a group of interrelated loads, the routes that they take, and the configurations in which they are pulled (see Fig. 2 for an example). By construction, a cluster is only defined if each of its loads is routed completely from origin to destination and can satisfy its time window. Given this definition, a number of important benefits are achieved. First, as in the path-based MCF formulation, we bypass the need for balance constraints. Second, recognizing that the cost of a given cluster can easily be computed off-line (given that a cluster is defined by the routing as well as the set of loads), the overall cost of a set of clusters then becomes a linear function. Third, we do not need to assign specific times to the loads, but merely ensure that valid times exist for which the loads will be delivered on-time. Finally, the opportunity to fractionally assign a single load to half of a double trailer configuration no longer exists, because individual loads are no longer treated independently. These benefits have significant impact on tractability, as we will demonstrate in Section 3.5.

A sample cluster is presented in Fig. 2. This cluster is comprised of three loads: one from A to C, a second from A to D, and a third from B to D. Loads AC and AD are matched to form a double trailer configuration between locations A and C, and loads AD and BD are matched to form a double configuration between nodes C and D; load BD travels as a single trailer from B to C. The cost of this cluster is simply the cost of double configuration moves from A to C and C

to D, plus the cost of a single configuration move from B to C. To determine whether this cluster is time-feasible, we assume that load BD leaves node B at its earliest available time, and that loads AC and AD leave node A at the later of their respective earliest available times. Similarly, loads AD and BD leave node C at the later of the two loads' arrival time at C. Based on these departure times, it is then trivial to determine whether the loads can all arrive at their destinations on-time (if not, then the cluster simply isn't included in the model).

Using this variable definition, LMR can be formulated as a simple variation of the classical *set partitioning* formulation [5], using the following notation.

Variables:

x_c = number of times cluster c is used in the solution

Parameters:

w_c = cost of cluster c

δ_{kc} = number of loads of commodity k that appear in cluster c (where a commodity is a set of loads sharing a common origin, destination, and time window)

b_k = number of loads of commodity k that must be moved through the network

Sets:

C = set of valid clusters c

K = set of commodities k

Formulation:

$$\text{Min } \sum_{c \in C} w_c x_c \tag{1}$$

s.t.

$$\sum_{c \in C} \delta_{kc} x_c = b_k \quad \forall k \in K \tag{2}$$

$$x_c \in Z^+ \quad \forall c \in C \tag{3}$$

The objective (1) is to minimize the cost of the selected clusters. Constraint set (2) requires that a combination of clusters be selected such that the correct number of loads of each commodity k are moved. Finally, constraint set (3) ensures that an integral number of each cluster is selected.

3.4. Generating Clusters

Traditionally, linear programs with very large numbers of variables are solved via *column generation* [7, 36], in which an embedded *pricing problem* uses dual information to identify promising variables to pivot into the basis, rather than enumerating all variables a priori. This approach depends on the ability to quickly solve this pricing problem, typically itself as a mathematical program. In our formulation, however, we have intentionally designed our cluster variables specifically to incorporate the difficulties (non-linear cost structure, time windows) that make a traditional

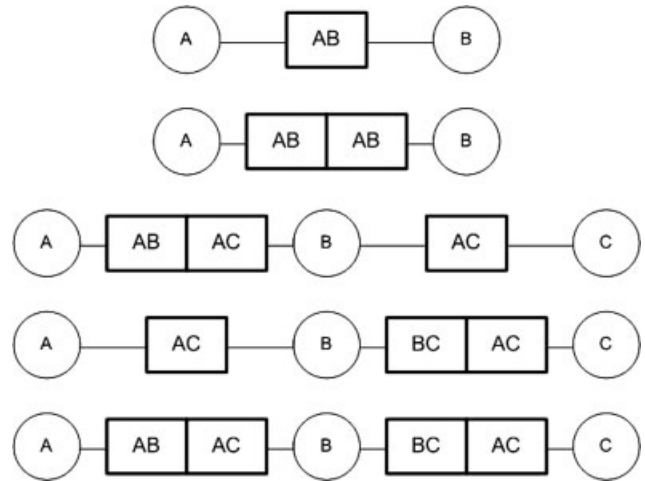


Figure 3. Templates for LMR computational experiments.

MP approach problematic. Given that both of these complications are captured implicitly within the variable definition, the problems that were present in the original MCF formulations—the large time-space network and the weak LP relaxation—would simply reemerge in the pricing problem. Because this pricing problem must be solved not once, but several times, this makes using a MP-based pricing problem approach impractical for the cluster-based formulation of LMR.

Given the computational challenges posed by a column generation approach, and the fact that in real-world problems we are concerned not only with solution quality (“provable optimality”) but also with runtime, we instead solve the problem by considering only a restricted set of variables, generated from *templates*, that are constructed a priori. Templates are predefined patterns of loads that have reasonable potential to appear in a high-quality solution. For each template, all valid load combinations are included in the formulation. Figure 3 displays the templates that we consider in our computational experiments.

Although limiting the problem to only these predefined cluster templates certainly might eliminate desirable solutions from consideration, we have found this not to be a significant disadvantage in our computational experiments. In an express package carrier network, loads' time windows are typically quite tight. As a result, this greatly restricts the number of intermediate stops that a load can make, because of the circuitous routing as well as the tractor reconfigurations, both of which are time-consuming. Very complex cluster structures, with large numbers of interconnecting loads, simply are not time feasible in most cases. Those that *are* time-feasible can often be broken into sets of smaller clusters without significant increase in cost. In our computational experiments, when we considered the inclusion of other,

Table 1. LMR problem instances.

ID	No. of Loads	No. of Comms.	No. of Nodes	No. of Clusters
1	2,067	1,257	263	8,955
2	3,539	2,426	352	21,666
3	13,636	6,394	568	116,031
4	30,554	12,404	1,106	324,622

more elaborate clusters, solution quality did not improve to any significant degree, but tractability was greatly impacted. Note also that complex clusters are actually undesirable from an implementation standpoint. This is because of the corresponding complexity in overseeing the operation of these clusters, with many pieces to be coordinated in downstream decisions such as driver scheduling. Furthermore, delays can propagate through large clusters with greater impact, as one delayed load may delay other loads in the cluster as well. Thus, smaller clusters may be viewed as more robust.

3.5. Computational Results for LMR

The purpose of our first set of computational experiments was twofold. First, we wanted to assess the tractability of our proposed cluster-based formulation. Second, we wanted to evaluate the impact of the heuristic approach—that is, could high-quality solutions be found while only considering a subset of the feasible clusters?

3.5.1. Tractability

We considered four different data sets, all provided by a major express package carrier. These range in size from approximately 2000 loads to approximately 30,000 loads. Table 1 describes the four data instances, giving the number of loads, the corresponding number of commodities (i.e. loads sharing a common origin, destination, and time window), the number of nodes, and the number of clusters (based on the templates shown in Fig. 3).

Each of these instances (as well as all other computational experiments in future sections) was solved using CPLEX version 11.0 on a 3.2 GHz Pentium D PC with 3.5 GB of real memory. Default CPLEX parameters were used, with an optimality gap of 1%. The results appear in Table 2, which shows the time (in seconds) to generate the set of clusters and then to solve the IP, the number of nodes in the branch-and-bound tree, and the optimality gap.

The results are encouraging. As expected, the LP relaxations are very strong—only one instance required branching in addition to CPLEX's standard preprocessing in order find an integer solution, and this instance required only 10 nodes in the branch-and-bound tree. In addition, because the model

is so small (recall that the number of constraints is the same as the number of commodities), the LP's solve quickly. In all four instances, solutions with an optimality gap less than one percent were found in less than 2 min.

3.5.2. Lower Bounds

These optimality gaps, however, are relative to the set of clusters provided, not to the original problem. To find a true lower bound against which to compare our heuristic results, we could solve the MCF formulation, which can guarantee a provably optimal solution. However, this is intractable for all but the smallest of problem instances, both because of its size (due to the time windows) and its weak LP relaxation (due to the linearization of the cost function). As an alternative, we set out to solve a relaxation of LMR in which we ignored time windows. Clearly, a solution to this problem will give a lower bound on the true optimal value. However, even when time windows are ignored, the size of the model and weak LP relaxation is still problematic. We attempted to run the MCF formulation of Instance One (with time windows ignored). This instance has 197,341 constraints and 1,364,362 variables. Even after fine-tuning the CPLEX parameters, the system runs out of memory after preprocessing and finding an initial integer solution (which is six percent larger than the heuristic solution found with LMR). When the system runs out of memory, the current lower bound of the MCF formulation shows that the solution to LMR is within eleven percent of optimality. Note, however, that this gap is based on a lower bound in which both time windows and integrality are relaxed; presumably, the true gap is substantially smaller.

3.6. Additional Benefits of the Cluster-Based Approach: Operational Considerations

Trading off between realism and tractability is always a challenge when modeling any real-world problem. In addition to the benefits of tractability achieved by the cluster-based model relative to a MCF-based approach, the cluster-based approach also enables us to incorporate several real-world operational considerations that could not be captured in a MCF-based approach.

Table 2. LMR results.

ID	Gen. time	Solve time	No. of BB Nodes	Opt. gap (%)
1	1	1	0	0.77
2	2	7	10	0.99
3	19	26	0	0.79
4	105	71	0	0.53

3.6.1. Allowances

When combining loads at intermediate locations to form double trailer configurations, there is a time delay associated with stopping, uncoupling existing loads, and joining new loads together. This time, referred to as the *allowance*, varies depending upon whether the loads are *inbound* or *outbound*, the type of facility where the stop occurs, and how many trailers are being coupled or decoupled. By ignoring these allowances in the calculation of travel time for a load, we may create a plan that violates the load's time window. Adding this operational consideration into a MCF-based formulation is not possible without making substantial structural changes, however, because the model does not include information about which loads are paired together, or when. On the other hand, it is trivial to include allowances into the cluster framework. We simply incorporate it in the off-line time-feasibility check we perform when enumerating clusters.

3.6.2. Triple Trailer Configurations

In some locations (for example, in parts of the mid-western United States), it is permissible to pull three pup trailers behind a single tractor as part of a *triple trailer configuration*. The potential cost savings of sharing a single driver and tractor between three load can be substantial. Capturing this opportunity in a MCF-based formulation, however, would require the addition of several new variables, representing the flow of triples in the network. Additionally, the model would become even more fractional, as the model would first try to route individual loads as one-third of a triple configuration, then one-half of a double, before finally finding an integer solution using a single configuration. Similarly, double loads would first be routed as two-thirds of a triple configuration. Thus, the amount of branching to find an integer solution would be substantially increased. Incorporating triple trailer configurations in the cluster-based model, on the other hand, requires only the addition of new cluster templates that incorporate the use of triple configurations on the appropriate arcs. Although this will certainly increase the number of variables in the model, the problem structure (and strength of LP relaxation) remain unchanged.

3.6.3. Intermodal Transport

Package carriers often move loads not only by road but also by rail. Transporting loads by rail is typically less costly but more time-consuming. In addition, the rail network interfaces with some but not all of the existing facilities in the network. Incorporating the rail network in a MCF-based formulation of LMR would give rise to an even larger time-space network, more variables, and more constraints. In the cluster-based approach, however, we can again incorporate

this additional opportunity by adding new cluster templates that capture the interactions of loads using rail (or other modes of transportation) for some or all of their journey.

3.6.4. Nonfacility Meets

Thus far, it has been assumed that trailers can only be coupled at carrier facilities within the network. In reality, however, these meets can occur at many other places as well. For instance, many parking lots near turnpike exits and interstate highways exist solely for this purpose. Incorporating these additional opportunities within a MCF formulation could be modeled by adding nodes for these meet opportunities, but this would dramatically increase the size of the time-space network. Permitting this option in the cluster-based formulation, on the other hand, would have no impact except on the pre-processing time of constructing the clusters, taking advantage of the carrier's knowledge of convenient meet locations.

3.6.5. User Expertise

More generally, another benefit of the cluster-based approach is that it provides a natural mechanism for leveraging the expertise of those who are most familiar with the problem. For instance, load planners and routers can play an active role in developing new cluster templates. Understanding the best practices of load routing as well as those things that have shown in practice to perform poorly is invaluable in developing a solution that is easy to implement and likely to succeed in practice.

3.6.6. Complexity Limitations

Finally, note that an optimal solution to a MCF-based formulation may be quite complex, with frequent matching and unmatching of loads and several stops at intermediate facilities. Although this solution may be the best in terms of cost, it may be significantly less desirable than a slightly more costly solution that is easier to implement. Limiting the set of clusters to manageable structures provides a natural way to address this operational issue.

4. INTEGRATED PLANNING: LOAD MATCHING AND ROUTING, TRAILER ASSIGNMENT, AND EMPTY BALANCING

As discussed in Section 2, the planning process for the express package industry is typically decomposed into five separate stages, due to its complexity. Each of these stages is considered individually, with the output from one problem becoming the input to the next. The cluster-based approach

to LMR has the benefit of not only providing a means to solve a difficult planning problem, but also of allowing us to expand the problem scope to consider additional stages of the planning process as well.

Integrating multiple stages of the planning process is advantageous both to ensure feasibility and to improve solution quality. Given that traditional modeling approaches to each of the individual stages are already computationally challenging to solve, directly integrating these models (for example, through the use of linking constraints), is not a viable option. The cluster-based formulation of LMR, on the other hand, naturally lends itself to the inclusion of trailer assignment and empty balancing decisions.

Recall that in the express package industry, volume is assigned to trailers prior to solving the LMR. The solution to the trailer assignment problem (TA) directly impacts LMR since the assignment of packages to pups and vans determines the set of loads to be moved in LMR and, consequently, which loads can be matched together. TA also impacts the empty balancing problem, because the set of loads also determines the imbalance of trailers at each node. Finally, LMR and EB are closely related, because loaded and empty pups can travel together as part of a double trailer configuration.

We refer to the integrated problems of trailer assignment, load matching and routing, and equipment balancing as TALMREB.

4.1. Modeling TALMREB

Incorporating TA and EB in a MCF-based formulation of LMR would require significant changes in model structure and size. Modifying the cluster-based LMR formulation to include TA and EB decisions, however, requires just a slight modification of the variable definition and the addition of one set of constraints. Instead of defining clusters to consider the movement of loads, we now consider the combined movement of *commodity-specific capacities* (associated with specific trailer types) and empty trailers. As before, we require the movement of the commodities in each cluster to be time-feasible and each commodity to be routed completely from origin to destination.

Consider the example presented in Fig. 4. In this cluster, two pups (one with the capacity to carry up to 800 packages from A to C and the other with the capacity to carry up to 800 packages from A to D) move together from A to C. On the B-C leg, there is one pup (with the capacity to carry 800 packages from B to D) and one empty pup. Finally, two pups travel together from C to D, one the continuation of the A to D packages and the other the continuation of the B to D packages.

Associated with each cluster is a set of local node imbalances. In the example in Fig. 4, because there are two pups departing from node A but none arriving, there is a net loss

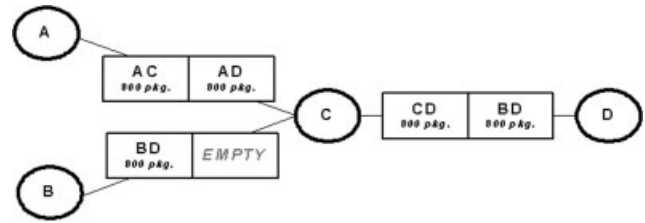


Figure 4. Sample cluster for integrated model.

of two pups. At node C, since there are four inbound pups but only two outbound, there is a net gain of two pups. As before, it is easy to compute the cost and time-feasibility of a cluster.

Using this new definition of a cluster, it is straightforward to modify the formulation presented in Section 3.3. To do so, we introduce the following notation.

Variables:

x_c = number of times cluster c is used in the solution

Parameters:

w_c = cost of cluster c

v_{kc} = maximum volume (in packages) of commodity k that can be transported in cluster c

b_k = demand (in packages) for commodity k

η_{tfc} = impact on balance of trailer type t at facility f for cluster c

Sets:

C = set of time-feasible clusters c

K = set of commodities k

T = set of trailer types t

F = set of facilities f

Formulation:

$$\min \sum_{c \in C} w_c x_c \quad (4)$$

st

$$\sum_{c \in C} v_{kc} x_c \geq b_k \quad \forall k \in K \quad (5)$$

$$\sum_{c \in C} \eta_{tfc} x_c = 0 \quad \forall t \in T, f \in F \quad (6)$$

$$x_c \in Z^+ \quad \forall c \in C \quad (7)$$

The objective (4) is to minimize the total cost of the clusters selected. Constraint set (5) requires that the selected clusters provide enough capacity to move the demand for each commodity through the system. Constraint set (6) requires trailer balance at each facility for each trailer type. Finally, constraint set (7) requires that an integral number of each cluster be selected.

This formulation, like the original LMR formulation, does not require the use of a time-space network. The number of constraints is manageable—one for each commodity and

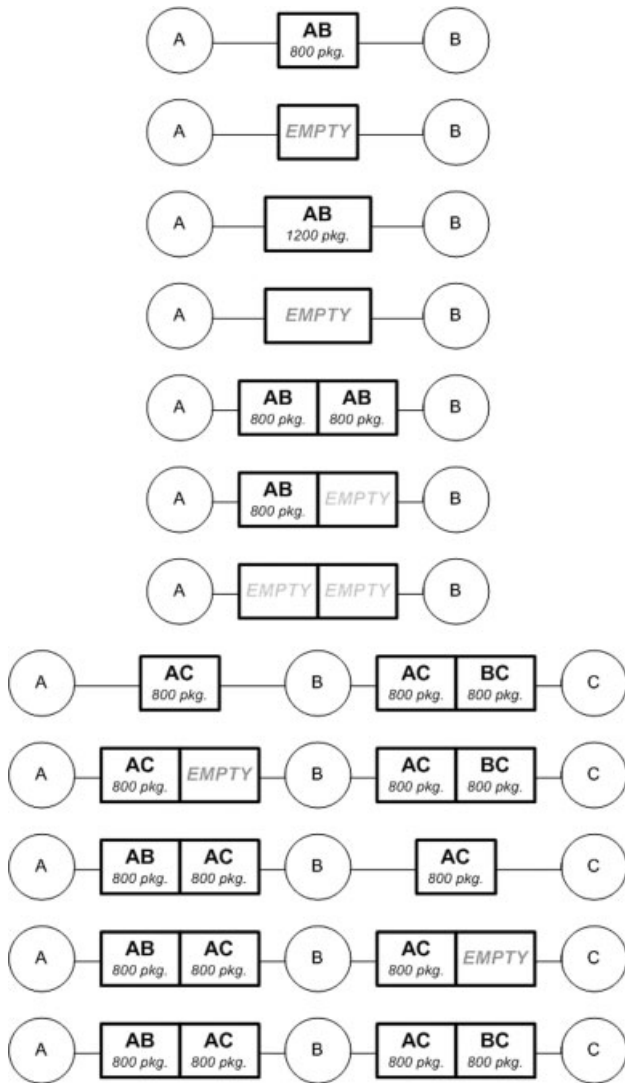


Figure 5. Templates for integrated model computational experiments.

one for each trailer type/facility combination. In addition, this formulation allows us to incorporate the operational considerations described in Section 3.6. One problem introduced by the integrated formulation, however, is that the cover constraints (5) lend themselves naturally to fractional solutions. For instance, if a cluster has a capacity of 800 packages of commodity k but only 200 packages need to be moved, it would be feasible (and desirable from a cost perspective) for the cluster to assume a value of one-fourth. To improve performance, we therefore introduce a set of cuts designed to strengthen the LP relaxation. Defining max_cap to be the maximum capacity for a given commodity across all feasible clusters, the cuts are

$$\sum_{c \in C: v_{kc} > 0} x_c \geq \left\lceil \frac{b_k}{max_cap} \right\rceil \quad \forall k \in K. \quad (8)$$

In other words, for each commodity, there is a minimum number of clusters that would be required to move this volume through the network. (8) requires that the number of clusters used in the solution be at least this minimum number. (For a similar result, see Ref. [4].)

4.2. Computational Experiments for TALMREB

In our computational experiments, we set out to determine the impact on tractability of adding the trailer assignment and empty balancing problems to the load matching and routing problem. We used the same four instances considered in Section 3.5. We again leveraged the idea of cluster templates introduced in 3.4, using the set of templates shown in Fig. 5. CPLEX defaults were used for all parameters, with the exception of MIPSearch, which was set to “dynamic” and VarSel, which was set to “MaxInfeas.” In addition, for the two larger data sets, MemoryEmphasis was set to “on”.

Results are presented in Table 3, which provides the number of commodities, the number of clusters, the run time (in seconds), the number of nodes in the branch-and-bound tree, the input optimality gap, and the actual gap of the final solution.

Note that we were unable to solve the largest instance - after approximately two days of run-time, the machine ran out of memory, having solved about 16,700 nodes without finding an integer-feasible solution.

We also ran the first three instances without cuts (8) to evaluate the impact of these cuts. As seen in Table 4, the smallest instance is actually faster without cuts - the benefits of a stronger LP relaxation are outweighed by the increased number of constraints. The other two instances, however, show the benefits of the cut. We also note that in the third instance, the solution quality is not as good (a 3.49% optimality gap versus a 4.7% gap). However, when we tried to run the fourth instance without cuts to achieve the lower gap, we ran out of memory.

4.3. Enhanced TALMREB

In Table 3, we observe that the smaller instances still solve very quickly. On the other hand, they are clearly becoming more fractional than LMR alone. This is largely due to the inclusion of empty trailers. Because empty trailers can be

Table 3. TALMREB computational results.

ID	No. of Comm.	No. of Clusters	Time (s)	Nodes	Set gap (%)	True gap (%)
1	1,257	20,604	79	400	1	0.66
2	2,426	48,442	228	640	1	0.98
3	6,394	235,080	16,429	5,569	5	3.49
4	12,404	621,312	X	X	5	X

Table 4. TALMREB computational results without cuts.

ID	No. of Comm.	No. of Clusters	Time w/cuts (s)	Time w/o cuts (s)
1	1,257	20,604	79	49
2	2,426	48,442	228	295
3	6,394	235,080	16,429	17,539

routed between any pair of nodes, and because it is always theoretically possible to match two empties on any arc, we must create these options in the cluster templates. Thus, whenever it is necessary to move a single empty trailer, this empty is initially instead assigned in the LP relaxation to one-half of a double empty configuration, so as to reduce cost. This is, on a smaller scale, similar to the fractionality of loaded trailers that is seen in a MCF-based formulation.

In the larger instances, we observe that this fractionality of empty clusters has far greater impact than in the smaller instances, resulting in dramatically slower run times. This motivated us to further investigate the impact of empty balancing (and, particularly, the trade-off between single empty pup configurations and double empty pup configurations) on performance. In addition, we have observed that, especially in the larger instance, it takes a very long time (independent of CPLEX parameter settings, of which we tried many combinations) to find the first integer-feasible solution. However, once this solution is found, it is typically close to optimal and the tree converges soon thereafter.

Thus, we decided to introduce a new step to the process—identifying an initial integer-feasible solution to use as a starting upper bound. To do so, we made the following simple yet powerful observation: if we relax the integrality of the double empty pup clusters and solve the problem (while still requiring all other clusters to take on integer values), then the solution can easily be converted to a feasible one—the only caveat is that in those locations where half of a double empty pup cluster has been assigned, this actually corresponds to a single empty pup cluster and thus the cost has been underestimated. Therefore, we can solve this relaxed problem, re-compute the accurate (and therefore higher) cost of this solution, and use it as an initial upper bound to enhance performance of *TALMREB*.

We implemented this and made two exciting discoveries. First, the relaxed version of the problem (again, where only the integrality of the double empty pup clusters has been relaxed and all other cluster variables are still restricted to be integer) solves very quickly. Second, the objective value of this problem is far better than we had expected—in some cases, we find a *better* solution (i.e. one with a smaller optimality gap) and in far shorter run time.

In retrospect, the high solution quality makes sense for a number of reasons. First, the data instances are relatively balanced—that is, empty balancing plays a somewhat limited role (on the order of 10%) in the solution cost. Second, not all arcs have odd numbers of empty pups assigned to them—those arcs with even numbers of pups are feasible in the relaxed version of the problem and the cost of these arcs are not under-estimated. Third, when there *are* large numbers of empty trailers on an arc, at most one of the clusters will be under-priced. For example, an arc with seven empty pups will have 3.5 double configurations in the relaxed version whereas the actual values would be three doubles and one single. As the number of empties gets larger, the impact of these remainders dilutes further.

These observations are supported by the results shown in Table 5. These instances were run on the same machine as those in Table 3 and using the same CPLEX parameter settings. The three gaps are the following: *set gap* is the CPLEX parameter set for the stopping criterion. *Realized gap* is the gap of the solution at termination. *True gap* is based on recomputing the solution value to take into account the inaccuracies in cost associated with half double empty pup clusters. For data sets 3 and 4, we considered multiple set gaps to demonstrate the trade-off between solution quality and run time.

The most important observation is the dramatic reduction in run times—for the largest instance, instead of failing to solve after 2 days, the solution can now be found in just a few hours! We also note that, for small instances, the original model actually performs better. This is because the empties play a proportionally larger role in the objective value (e.g. arcs might have one or two empties moving across them, rather than seven or eight in a larger network, and so the inaccuracy of a single empty that must be re-cost for a given arc carries greater weight). Finally, we point out that the

Table 5. Revised TALMREB computational results.

ID	No. of Comm.	No. of Clusters	Time (s)	Nodes	Set gap (%)	Real. gap (%)	True gap (%)
1	1,257	19,308	7	0	1	0.95	2.64
2	2,426	25,950	68	270	1	0.93	1.48
3	6,394	228,484	189	0	5	4.36	4.71
3	6,394	228,484	3,141	975	1	0.99	1.36
4	12,404	608,640	885	0	5	4.74	4.97
4	12,404	608,640	5,086	491	2.5	2.5	2.72

number of clusters in the model decreases, because there is no point in including clusters for single empty pups, as they will never be used in the optimal relaxed model.

We conclude this section by observing that the largest data set still poses some computational challenges, due to its size. Specifically, we ran out of memory when trying to reduce the optimality gap further. This was not because of the size of the branch-and-bound tree (with regards to number of nodes) but rather due to the size of the individual LPs within the tree. Clearly, running on a machine with more than 3.5 GB of real memory could help to improve performance. We also suggest that *delayed column generation*, in which not all clusters are stored initially, but instead are generated at each node of the branch-and-bound tree as needed, might improve memory utilization.

4.4. Value of Integration

We conclude our computational experiments with a brief exploration of the value of integration. Specifically, we also solved each of the four instances by breaking the problem into three steps—first trailer assignment, then load matching and routing (based on the loads constructed in trailer assignment), then empty balancing (taking advantage of the existing loads for matching opportunities). Note that it is still necessary to solve the load matching and rounding problem heuristically (i.e. with our cluster-based approach), as even for this single problem, a true optimal solution cannot be found for the reasons discussed in Section 3.

For these four instances, the benefit ranged from 1 to 3% percent (with the disaggregated solutions being found in under an hour of run time). The majority of this benefit comes from improvements in empty balancing costs (which were on the order of 10% of total system costs). The reason for this is that the link between trailer assignment and load matching and routing in and of itself does not cause real benefits. For example, if the demand on a lane is small enough to fit into a single pup, then there is no value in assigning it to a van, which has higher per-mile cost and decreases the potential opportunity for matching loads. Similarly, if the volume fits in a single van, there is no benefit in assigning it to a set of two pups, because this double configuration will be higher cost and will not yield matching opportunities (as the loads are already matched). It is only when the interaction with empty balancing is incorporated that all three pieces of the problem become tightly connected.

It is interesting to note that the data sets that we considered came from a fairly balanced network and thus the improvements in empty balancing cost have somewhat limited impact on overall system cost, as empty balancing does not dominate cost. We observe that as the network becomes more imbalanced, the benefits of integration grow substantially. For example, we considered a variation of the original data

sets in which each node was earmarked as either predominantly outbound or predominantly inbound (which is the case in many networks). In this case, the impact of aggregation was a savings of almost twenty-eight percent of total system cost!

Thus, this seems to suggest that for certain networks, the disaggregated approach can yield very high quality solutions in short run times. For those networks with less system balance, on the other hand, the integrated approach can yield substantial benefits. In these cases, the disaggregated approach may also provide benefits in terms of providing initial integer-feasible solutions.

5. CONCLUSIONS AND FUTURE RESEARCH

In this article, we show how a novel modeling approach (which embeds much of the system complexity within the variable definition, rather than capturing it through constraints), in tandem with the use of templates to construct candidate variables, can enable us to address many of the challenges found within express package carrier planning. In particular, we show how the load matching and routing problem, which is intractable using a MCF approach, can be solved through the use of variables that represent clusters of loads. In addition to achieving tractability, this cluster-based approach also allows us to pose a more realistic version of the problem (for example, including allowances in time calculations, and incorporating additional trailer configurations and transportation modes).

Perhaps more importantly, we show that this approach not only enables us to solve one challenging planning problem that is intractable under more traditional methods, but can actually be extended to simultaneously solve three such problems (trailer assignment, load matching and routing, and empty balancing), which under certain network configurations can lead to substantial savings in cost. This integrated formulation is simply a minor modification of the original, single-problem model, rather than an explicit linking of three individual large-scale mathematical programs.

In our approach, the cluster-based variables, representing groups of interacting loads, are designed to target sources of intractability in a more traditional MCF approach—the large time-space network needed to capture timing constraints and the weak LP relaxation associated with linearizing the objective function. Although this cluster-based formulation has a theoretically exponential number of variables, we show that a much smaller number of variables, derived from promising templates, can be used to achieve high-quality solutions for real-world problem instances in acceptable run times. These templates are developed to exploit operational considerations and in recognition of the difficulties posed by the more-typical approach of column generation, in which the

pricing problem would suffer from the same difficulties as a MCF approach to the original problem.

We suggest that a cluster-based approach, in conjunction with a template-based solution methodology, may have applicability in other freight transportation problems as well—one area of future research is to investigate this hypothesis. For example, the LTL industry has many structural similarities to the express package industry, and also faces the problems of trailer assignment, load matching and routing, and empty balancing, albeit with different operational factors heavily influencing the decision-making process. Within the express package industry, other outstanding research questions include: further integration to include load planning and/or driver scheduling decisions; the development of new templates to improve solution quality (or tighter lower bounds to improve the optimality gap of existing solutions); and the inclusion of variability within the planning process to improve solution robustness. Finally, we note the outstanding question of how to address memory issues for large instances; delayed column generation (in which not all clusters are stored explicitly in the LP's) could potentially improve the performance of the integrated model.

ACKNOWLEDGMENTS

The authors wish to acknowledge the technical support of Mr. Chris Konrad in enabling our computational experiments, and to the anonymous referees for the insightful feedback.

REFERENCES

- [1] R. Ahuja, T. Magnanti, and J. Orlin, *Network flows: Theory, algorithms, and applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [2] M. Akyilmaz, An algorithmic framework for routeing LTL shipments, *J Oper Res Soc* 45 (1994), 529–538.
- [3] L. Appelgren, A column generation algorithm for a ship scheduling problem, *Transport Sci* 3 (1969), 53–68.
- [4] A. Armacost, C. Barnhart, and K. Ware, Composite variable formulations for express shipment service network design, *Transport Sci* 35 (2002), 1029.
- [5] E. Balas and M. Padberg, Set partitioning: A survey, *SIAM Rev* 18 (1976), 710–760.
- [6] C. Barnhart, A. Farahat, and M. Lohatepanont, Airline fleet assignment: An alternative model and solution approach based on composite variables, *Oper Res*, [accepted for publication].
- [7] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, and P. Vance, Branch-and-price: Column generation for solving huge integer programs, *Oper Res* 46 (1998), 316–329.
- [8] C. Barnhart and D. Kim, Routing models and solution procedures for regional less-than-truckload operations, *Ann Oper Res* 61 (1995), 67–90.
- [9] C. Barnhart, N. Krishnan, K. Daeki, and K. Ware, Network design for express shipment delivery, *Computat Optim Appl* 21 (2002), 239–262.
- [10] O. Braeysy and M. Gendreau, Vehicle routing problem with time windows. I. Route construction and local search algorithms, *Transport Sci* 39 (2005), 104–118.
- [11] O. Braeysy and M. Gendreau, Vehicle routing problem with time windows. II. Meta-heuristics, *Transport Sci* 39 (2005), 119–139.
- [12] J. Campbell, Freight consolidation and routing with transportation economies of scale, *Transport Res B* 24 (1990), 345–361.
- [13] V. Caraffa, S. Ianes, T. Bagchi, and C. Sriskandarajah, Minimizing makespan in an blocking flowshop using genetic algorithms, *Int J Prod Econ* 70 (2001), 101–115.
- [14] J. Castro and N. Nanoba, Implementation of linear and nonlinear multicommodity network flows, *Eur J Oper Res* 92 (1996), 27–53.
- [15] R. Cheung and B. Muralidharan, Dynamic routing for priority shipments in LTL service networks, *Transport Sci* 34 (2000), 86–98.
- [16] A. Cohn and C. Barnhart, Improving crew scheduling by incorporating key maintenance routing decisions, *Oper Res* 51 (2003), 387–396.
- [17] A. Cohn and C. Barnhart, Composite-variable modeling for service parts logistics, *Ann Oper Res* 144 (2006), 17–32.
- [18] A. Cohn, S. Root, A. Wang, and D. Mohr, Integration of the load matching and routing problem with equipment balancing for small package carriers, *Transport Sci* 41 (2007), 238–252.
- [19] Y. Chen and L. Hsiao, Shipping problems with body clock constraints, *Comput Oper Res* 30 (2003), 1037–1049.
- [20] T. Crainic, A survey of optimization models for Long-Haul freight transportation, *Handbook of Transportation Science*, Boston, Kluwer, 2003.
- [21] T. Crainic and G. Laporte, Planning models for freight transportation, *Eur J Oper Res* 97 (1997), 409–438.
- [22] T. Crainic and K. Rousseau, The column generation principle and the airline crew scheduling problem, *INFOR* 25 (1987), 136–151.
- [23] D. Dave, Patterns of freight flow and design of a less-than-truckload distribution network, Georgia Tech, 2003.
- [24] J. Eckstein and Y. Sheffi, Optimization of group line-haul operations for motor carriers using twin trailers, *Transport Res Record* 1120 (1987), 12–23.
- [25] A. Geoffrion, Elements of large-scale mathematical programming, *Management Sci* 16 (1970), 652–691.
- [26] I. Ghamlouche, T. Crainic, and M. Gendreau, Cycle-based neighborhoods for fixed-charge capacitated multicommodity network design, *Oper Res* 51 (2003), 655–667.
- [27] R. Hall, Direct versus terminal freight routing on a network with concave costs, *Transport Res B* 21 (1987), 287–298.
- [28] J. Ho, Recent advances in the decomposition approach to linear programming, *Math Program Study* 31 (1987), 119–128.
- [29] K. Jones, I. Lustig, J. Farvolden, and W. Powell, Multicommodity network flows: The impact of formulation on decomposition, *Math Program* 62 (1993), 95–117.
- [30] J. Kennington, A survey of linear cost multicommodity network flows, *Oper Res* 2 (1978), 209–236.
- [31] D. Kim and C. Barnhart, Multimodal express shipment service design: Models and algorithms, *Comput Indus Eng* 33 (1997), 685–688.
- [32] J. Kliniewicz, Solving a freight transport problem using facility location techniques, *Oper Res* 38 (1990), 99–109.
- [33] M. Kubly and R. Gray, The hub network design problem with stopovers and feeders: The case of federal express, *Transport Res A* 27 (1993), 1–12.

- [34] M. Lohatepanont and C. Barnhart, Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment, *Transport Sci* 38 (2004), 19–32.
- [35] C. Lin, The freight routing problem of time-definite freight delivery common carriers, *Transport Res B* 35 (2001), 525–457.
- [36] M. Lübbecke and J. Desrosiers, Selected topics in column generation, *Oper Res* 53 (2005), 1007–1023.
- [37] R. Martin, Large scale linear and integer optimization: A unified approach, Kluwer Academic Publishers, 1999.
- [38] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy, Airline fleet assignment with time windows, *Transport Res* 34 (2000), 1–20.
- [39] H. Sherali and W. Adams, A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems, *Discrete Appl Math* 52 (1994), 83–106.
- [40] R. Tardy and C. Barnhart, Optimization models and algorithms for large-scale, capacity constrained pick-up and delivery problems with time windows, MIT, 2005.
- [41] F. Vanderbeck and L. Wolsey, An exact algorithm for IP column generation, *Oper Res Lett* 19 (1996), 151–159.
- [42] U.S. Department of Transportation, Freight in America: A new national picture, Bureau of Transportation Statistics, 2006.
- [43] U.S. Department of Transportation, Economic impact of transportation, Bureau of Transportation Statistics, 2006, www.bts.gov/programs/freight_transportation/html/transportation.html.
- [44] Reuters Corporate Profile of UPS, www.investor.reuters.wallst.com/stocks/company-profile.asp?rpc=66&ticker=UPS, 2006.