INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Optimizing Transportation by Inventory Routing and Workload Balancing: Optimizing Daily Dray Operations Across an Intermodal Freight Network

Xiaoqing Sun, Manish Garg, Zahir Balaporia, Kendall Bailey, Ted Gifford

Interfaces
Vol. 44, No. 6, November–December 2014, pp. 579–590
ISSN 0092-2102 (print) | ISSN 1526-551X (online)
http://dx.doi.org/10.1287/inte.2014.0746

Optimizing Transportation by Inventory Routing and Workload Balancing: Optimizing Daily Dray Operations Across an Intermodal Freight Network

Xiaoqing Sun, Manish Garg, Zahir Balaporia, Kendall Bailey, Ted Gifford
Schneider National Inc., Green Bay, Wisconsin 54306 {sunp@schneider.com, gargm@schneider.com, balaporiaz@schneider.com, kbailey@schneider.com, giffordt@schneider.com}

In rail-based intermodal freight operations, full containers are moved by truck from shipper locations to rail ramps, transported via train to destination ramps, and then moved again by truck to consignee locations. We commonly refer to the roadway portions of this activity as dray operations. In a large metropolitan hub area, such as Chicago or Los Angeles, this drayage activity may involve several hundred drivers and up to 500 daily container moves to or from several distinct rail ramps. Both cost and environmental considerations drive the need to maximize driver productivity and minimize the time and miles not directly associated with moving loaded containers to or from rail ramps. In this paper, we describe a solution to this problem using a set-partitioning formulation and column-generation heuristic and report on a large-scale implementation. We focus on real-world implementation details that include (1) fast solve times to support near-real-time re-solving in the face of constantly changing data, (2) adjustments to account for traffic congestion and other operational considerations, and (3) integration with a commercial transportation management system to provide real-time data to the optimizer and to send solution recommendations to a driver-assignment process.

Keywords: computer science; transportation/shipping; tree algorithms; networks/graphs; deterministic sequencing; production/scheduling; Benders decomposition; algorithms; integer programming; column generation.
History: This paper was refereed. Published online in *Articles in Advance* June 30, 2014.

Increasing fuel prices and the geographic expansion of distribution networks have made intermodal freight transport systems an increasingly competitive alternative to single-mode transportation services. The use of multiple transport modes leverages tradeoffs between cost and efficiency on high-volume modes and access and flexibility on low-volume ones. The term intermodal, as distinct from multimodal, is normally reserved for the sequential movement of a container or full-truckload shipment across several modes of transportation. Multimodal is a more general term that includes less-than-truckload and small-package delivery, and often involves consolidation and deconsolidation of individual freight units as they move through a defined transportation network.

The two prevalent forms of intermodal transport are road-rail-road for continental shipping and road-sea-road for intercontinental shipments. To facilitate mode changes, freight is normally loaded into containers, with capacities ranging from 1,000 to 4,000 cubic feet that can be easily transferred between modes. The middle leg of the freight transport, train, or sea-going vessel realizes low per-unit cost and high efficiency by moving many containers at a time. The largest cargo vessels now in service carry 15,000 twenty-foot equivalent units and the longest trains carry as many as 500 double-stacked 53-foot containers. The end legs of transport, termed drayage, are characterized by much shorter transit distances and the movement of individual containers to or from time-constrained customer locations. The simultaneous arrival or departure of a large number of containers at a port or rail hub creates a challenge for dray transportation resources. Although this activity is often outsourced to third-party draymen, which we refer to as foreign carriers in this discussion, cost and environmental impact considerations can make it attractive to employ a dedicated fleet of tractors and drivers and use route and schedule optimization software systems to maximize the utilization of the fleet.

Figure 1: This map depicts railroad routes for the intermodal freight network.

Schneider National, Inc. operates a large intermodal freight transportation network, which encompasses the continental United States with significant coverage in Canada and Mexico. This network has 24 rail hubs, served by a fleet of more than 1,300 trucks and 14,000 containers, and moves more than 4,000 dray shipments per day, including pickup, delivery, and crosstown transfers between railroads, and repositioning moves (see Figure 1). Dray shipment distances range from several miles to several hundred miles. Drivers transporting the longer-distance loads may be away from home overnight, but most drivers return home at the end of each daily shift.

In addition to company-employed and independent contractor (IC) drivers, whose schedules are managed directly by the company, dray freight is also moved by foreign carriers who provide flexible capacity to support seasonal shifts in volume and shorter-term demand fluctuations.

Specific freight characteristics must also be considered. Examples include shipments or locations that require specific driver certifications and weight limits that necessitate the use of lighter day cabs (a tractor that does not have a sleeper berth). Actual pickup and delivery times also vary, depending on whether the driver must wait for the container to be loaded or unloaded or can quickly drop (and depart) or pick up a preloaded container.

In this paper, we describe the implementation of a computer-based solution to address the recurring daily problem of assigning drivers to transport drayage to both maximize driver productivity and minimize the total operations cost. This project consists of (1) the development of a mathematical programming optimization model, (2) the implementation of the model into an operational decision support system, (3) the integration via near real-time data flows into a suite of commercial software systems, and (4) the development of corresponding business processes to realize significant tangible benefits. We begin with a discussion of the business processes, follow with a review of relevant prior work, and describe the implementation and a number of modeling details. We then review the financial benefits and discuss model extensions and enhancements now in development. Appendix A covers algorithmic details and the mathematical formulation.

Business Process for Intermodal Operations

The flow of information, with corresponding processing and decision steps, is shared across a set of software modules (see Figure 2).

Intermodal freight orders are initially captured in the order entry system via either electronic transfer from a customer or direct entry with information received by fax or telephone (step 1). Order information includes origin pickup location, destination delivery location, and associated time constraints. Some orders may

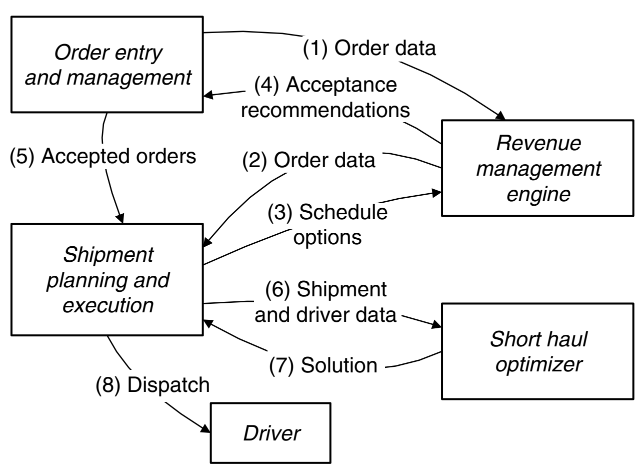

Figure 2: This diagram illustrates the process flow for intermodal freight orders. The numbers in parentheses represent steps in the process flow.

require more than one pickup (delivery) stop to complete the loading (unloading) of the container. As part of the order acceptance process, the order is validated for geographic and transit feasibility, train options are determined, and preliminary profitability analysis is carried out (steps 2, 3, and 4, respectively). If the order is confirmed, it is passed to the transportation management system (TMS) for splitting into component shipments (step 5). For example, an order from Chino, California to Woburn, Massachusetts would plan into five shipments:

1. Origin dray from shipper in Chino to railroad ramp in Los Angeles, California;
2. Rail shipment from Los Angeles to Chicago, Illinois on western U.S. rail service provider;
3. Crosstown dray in Chicago from western U.S. to eastern U.S. rail service provider;
4. Rail shipment from Chicago to Worcester, Massachusetts on eastern U.S. rail service provider;
5. Destination dray from Worcester ramp to consignee in Woburn.

The number of component shipments for an order varies from three (for the simple single-rail-leg case) to as many as 10 when several railroads and international (e.g., Canada-U.S.-Mexico) border crossings are required.

The rail shipment components of each order are planned first using train schedule information and relevant feasibility and cost information. When this step is complete, the corresponding dray shipments (origin, destination, and crosstown) are grouped into subsets according to the rail hub regions and time frames that apply. As we describe in the *Implementation Details* sections, the data integration process manages these shipment subsets and presents them, together with current information on the appropriate drivers and relevant cost and feasibility data, to instances of the short haul optimizer (SHO) (step 6).

The SHO provides a user interface that displays shipments and drivers for a dispatching region. Separate tabs for pickups, deliveries, and crosstown moves display key information and allow users to quickly filter and sort the data to facilitate dispatch-related tasks. The assignment optimization process runs on a schedule for each dispatching region, typically at 10-minute intervals. The user interface displays a summary of the solution from the last optimization run and provides detailed assignment information. It also provides information about infeasible shipments and potential excessive use of foreign carriers. This enables the user to identify potential problems to be addressed on an exception basis. The user is then able to accept SHO recommendations, and driver-shipment assignments are communicated back to the TMS for dispatch to the driver (steps 7 and 8, respectively).

Figure 3 illustrates a tour for a single driver. In this example, the first leg is a loaded move, 37 miles from the BNSF ramp at Haslett, Texas (near stop 7 on the map in Figure 3) to a customer location in Lewisville, Texas (stop 3). The driver then moves empty 37 miles to the KCS ramp in Dallas, Texas (stop 4) to pick up a loaded container and deliver it to a customer location in Grand Prairie, Texas (stop 5)—a distance of 29 miles. The driver next moves two miles to location (stop 6) and picks up a loaded container for delivery to the BNSF back in Haslett (stop 7).

Literature Review and Solution Methodology Description

A considerable body of literature pertaining to vehicle routing in general and intermodal freight transportation specifically is available. Caris et al. (2008) discuss various planning problems in intermodal freight transport and provide an overview of research in this area. Because our discussion focuses on the daily drayage planning problem of determining driver tours, previous work on vehicle routing is generally more relevant. Erera and Smilowitz (2008) provide a good overview of dray planning problems; in particular, they note that the problem is NP-hard. They describe a column-generation approach for solving these problems; however, the modeling constructs are different. Both Desrochers et al. (1992) and Savelsbergh and Sol (1998) provide relevant examples of applying column generation to successfully solve vehicle routing problems. Baldacci et al. (2011) give an exact solution technique for solving pickup and delivery problems with time windows based on a branch-and-cut-and-price algorithm; however, exponential time complexity makes this approach impractical for our problem size and response-time requirements.

Our approach combines a mixed-integer programming (MIP) master problem with subproblems that

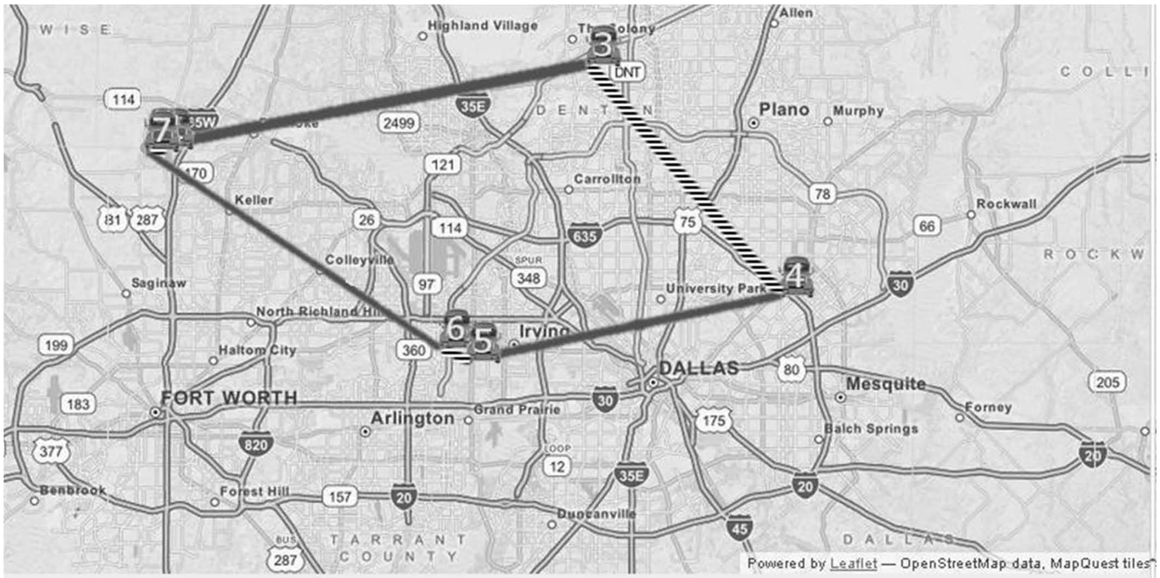

Figure 3: This map illustrates a representative driver tour. The numbers represent stops along the tour.

solve resource-constrained shortest-path problems using a labeling heuristic. Namboothiri and Erera (2008) propose a similar approach for managing drayage services in port operations. Their approach, referred to as a layered shortest-path heuristic, uses dynamic programming to find new tours that will be time feasible. As Erera and Smilowitz (2008) note, this heuristic appears to perform well, relative to exact methods, for loosely constrained problems. Our case can include a large percentage of narrow appointment windows and significant variation in shipment transit duration; therefore, we would not expect this approach to work particularly well. As such, we account explicitly for both the elapsed time and the accumulated reduced cost along partially generated tours and use this information to determine those to extend and include in the next MIP iteration. Our work extends and refines the work of Ileri et al. (2006), particularly with respect to details and implementation of the tour generation process. Ileri et al. (2006) generate tours for groups of drivers that share similar attributes and insert empty-equipment movements required to pick up a shipment during the column generation. Because grouping drivers is practical only for a start-of-day solution, we generate routes for individual drivers to support our iterative re-solve approach described in the *Implementation Details* section.

At the beginning, we precompute the set of feasible assignments of drivers to shipments and transitions between shipments. The column-generation process can then follow these transitions and correspond to feasible tours that string together the sequential execution of one or more shipments for a single driver. We generate two types of tours—those assigned to foreign carriers and those assigned to the company-managed fleet. The cost and operating characteristics of the former allow us to precompute the relatively small number of tours that are reasonable for them. For the company fleet, the number of possible tours is exponentially large. We provide a detailed explanation of how we generate these tours in Appendix A.

Implementation Details

Integration: Data Sources and the Transportation Management System

The dray optimization software module (i.e., SHO) sits within an integrated software suite that includes order management, shipment planning and execution, driver management, and financial components. The SHO integrates closely with the TMS, which maintains current details and status information related to shipments and drivers. To meet the requirement for timely data while minimizing the performance impact on the TMS, we maintain data hubs that are refreshed with new information every few minutes to supply current shipment and driver data to the SHO. The SHO then computes any necessary changes in model data, and accordingly updates the tour-generation graphs and mathematical program, removing shipments that no longer require planning and tours that are no longer feasible or in play. The SHO is then run with a warm start to produce a new solution.

When the user accepts the SHO solution, the corresponding shipment-to-driver assignments are posted back to the TMS, where they are reviewed and communicated to the drivers via integration from the TMS to the drivers' in-cab communication system.

Objective Function and Cost Factors

The objective function combines the cost of serving the tours to be executed with penalties for leaving certain shipments unassigned. If a shipment's time windows fall completely within the specified time horizon, the penalty cost is set high to ensure execution. In other cases, a modest penalty is imposed to encourage the achievement of various operational objectives.

The cost of executing a tour for company drivers includes driver pay, fuel costs, driver delays, stop-off (extra-stop) charges, and equipment depreciation. Foreign carrier costs are based on contractual agreements that include these various costs in the contracted rates. Other factors, including tour robustness, chassis rental cost, ramp storage cost, and equipment utilization, are also considered.

In the *Extensions* section, we briefly discuss fleet-size optimization. In the present operations, foreign carrier coverage varies from 5 to 15 percent on a daily basis, with the per-shipment cost premium averaging about 60 percent. Although this cost differential would seem to create a strong incentive to increase the size of the company fleet, an oversized fleet can quickly degrade productivity because of demand variation and geographic imbalance.

Driver Groups and Work Schedules

Several groups of drivers work within typical dray operations. Company drivers are company employees and have pay packages that include fixed and variable compensation components. ICs are similar to company drivers; however, they are responsible for their own tractors and have pay agreements that reflect this. Both company and IC drivers are part of either a local or regional fleet. Local drivers transport shipments up to a maximum distance (typically 150 miles, but it can vary by hub region) and return home each day. Regional drivers transport shipments subject to a minimum distance (typically 100 miles, but it can vary by hub region) and are sometimes away from home for one or two nights. These drivers are subject to standard U.S. Department of Transportation hours-of-service rules and must take a 10-hour sleep break after no more than 11 driving hours or 14 total work hours.

Generally, when a foreign carrier is selected to move a shipment, we incur the full round-trip cost, but may accept a two-shipment tour if it can be executed within a specified transit constraint. Therefore, the number of potential carrier tours is small enough to enumerate in advance.

Container Management

A critical requirement of intermodal freight transport is ensuring that adequate numbers of containers are available where and when they are needed. This issue has two important components, as we discuss next.

(1) Addressing imbalance in freight flows across the rail network. As freight flows across our 24-hub rail network, containers accumulate in regions with net inbound flow and become depleted in those with net outbound flow. This necessitates repositioning empty containers among the hub regions on a regular basis. The problem of determining the appropriate empty-repositioning moves to achieve the desired hub inventory levels is handled by a reasonably straightforward model and corresponding subsystem that are outside the scope of this discussion. To facilitate this rail-based repositioning at the local level, we need to move empty containers between rail ramps and customer locations or container storage yards. We currently plan these moves outside the SHO process; however, we plan to incorporate them into the optimization process.

(2) Meeting specific requirements at individual customer locations. In addition to executing empty shipments, drivers must frequently make intermediate stops to acquire or dispose of containers to arrive at the next location with (without) empty containers, based on

customer requirements. As currently implemented, the system does not explicitly evaluate and plan these stops; instead, it introduces a penalty cost and time delay into the tour to estimate the impact of any required detour. Although we have designed model enhancements to explicitly address container repositioning, we have not yet implemented the required data integration into the TMS. We are evaluating whether the resultant solution quality improvement would justify the additional problem complexity and solve-time impact.

Time Horizon and Re-Solve Frequency

The drayage optimization problem is formulated and solved independently for each rail hub region. Problems are formulated with a finite time horizon, currently set for 36 hours, to allow us to look ahead through the following day. New information about cancellations, additions, changes to shipments, and updates to driver locations and availability arrives continuously. To account for this information flow, we update the problem and re-solve it frequently. The re-solve interval is scheduled by region and controlled by the user, with 10 minutes as a typical value. Successive re-solves are run from a warm start, preserving feasible columns from the previous solution and reusing the tour-generation graphs with suitable adjustments. A cold start is required very rarely, usually only when a new rail hub region is instantiated. In this case, we construct the initial problem by enumerating the set of feasible single-shipment tours.

Additional Model Attributes

To obtain solutions that are both implementable and robust in the face of operational variability and specific circumstances, we capture and apply a number of additional modeling attributes. These conditions are embodied in a collection of business rules, data filters, conditional time adjustments, and penalty and (or) bonus cost factors. To effectively implement these, we have developed several ancillary tools, including (1) a database that supports clauses, parameters, and factors to capture these attributes, (2) a preprocessor to convert these database representations into efficient in-memory structures, and (3) a functional operator that applies this information, as appropriate, during the graph construction and tour-generation processor. Although the details of these components are beyond the scope of this paper, we will briefly note a few of them, as follows:

Tractor rule	Maintains tractor-location feasibility conditions relative to tractor types.
Drop pick	Enforces location-specific sequencing.
Hazmat	Enforces hazmat-certification feasibility.
Location delay	Specifies time delays for certain location combinations.
Outgate	Increases outgate time for certain shipments.
Last load	Allows for shipment carryover to next shift.
Priority appt	Pads times for priority shipment stops to decrease fail probability.
Priority stop	Applies stop-time penalty to encourage earliest delivery.

Time Windows

Railroad train schedules dictate a portion of the time constraints associated with a dray shipment. For each train, the railroad specifies a cutoff time, which can vary from two to 24 hours before train departure, by which outbound containers must arrive at the rail ramp for transfer from chassis to railcar. For the corresponding destination ramp, the railroad announces a grounding time that indicates when containers from a specified train are available for pickup at the ramp. Cutoff times are usually stable; however, grounding times are subject to periodic revision during transit because a train may be subject to various delays.

Conversely, the time constraint for the customer side of dray shipments varies according to customer and (or) location requirements. Some shipments may specify prearranged appointment times with little or no tolerance; others may specify an interval window; and some are open-ended with only no-sooner-than or no-later-than directives.

Rush-Hour Transit Time Adjustments

The geographic areas covered by our dray operations overlap urban areas and are subject to considerable traffic congestion variation during the day. To produce solutions that are both implementable and efficient, transit times used in the optimization process must account for time-of-day variations. Arcs and nodes are repeatedly explored during the column-generation

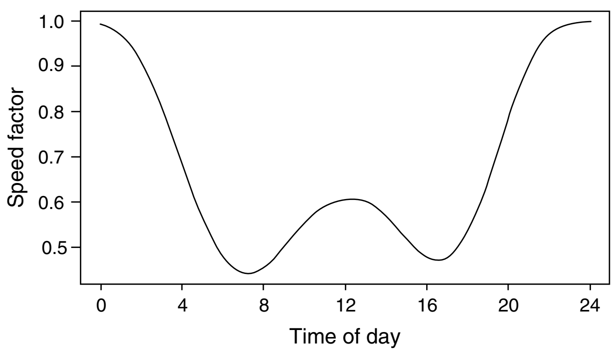

Figure 4: This graph shows a typical speed-reduction factor for urban areas by time of day.

Shipments	
Total	407
Must move today	207
From customer to ramp	180 (44%)
From ramp to customer	225 (55%)
Other (repositioning)	2
Average distance	117 miles
Drivers	
Regional	145
Local	90
Rail ramps	6
Customer locations	192

Table 1: This table shows details for a typical problem instance.

phase and each exploration may induce a different start time. Consequently, the speed of the transit time calculation or retrieval plays a significant role in overall computational performance. To achieve a good trade-off between performance and accuracy, we precompute and cache data that are used to quickly estimate transit times. We begin by segmenting each dray operations region into a rush-hour sensitive urban and a rush-hour indifferent rural area. Using historical data and nonparametric regression, we can generate a rush-hour profile for each urban area; in the example in Figure 4, the x-axis specifies a start time and the y-axis a speed-discount factor.

For each arc (i.e., route between locations), we identify the set of segments determined by intersecting the route with the urban and rural areas and then maintain a list of tuples, $\{(t_1, p_1)(t_2, p_2), \ldots\}$, where t_i is the rush-hour-free transit time and p_i is the profile number associated with the segment or 0, if the segment lies within a rural area. During the search process, we quickly calculate transit times for segments by applying the discount (to standard speed) corresponding to the (simulated) start time of the segment.

Results and Computational Performance

To illustrate the computation details of the SHO, we show problem-size data for a large representative dray-scheduling problem in Table 1, and then summarize the corresponding solution and performance characteristics. For reference, we provide hardware and software platform details in Appendix B.

The total processing time for the solution was 119 seconds. Initial graph construction consumed 2.68 seconds, optimization processing time was 88.5 seconds, and the remaining time was associated with data movement and other overhead. During graph construction, roughly 100,000 arcs were generated to capture feasibility and precedence relationships between shipments or shipments and drivers.

Figure 5 shows the reduction in the linear programming (LP) objective value and the number of new columns generated at each successive iteration. Iteration final (9) represents the integer programming (IP) solution of the last master problem (Iteration 8).

Figure 6 shows the time spent in the column-generation phase and LP-IP solver phase for each iteration.

We also observed SHO performance across a wide range of problems. In Figure 7, we plot optimizer processing time as a function of the number of shipments for 6,889 problem instances across 29 distinct problem

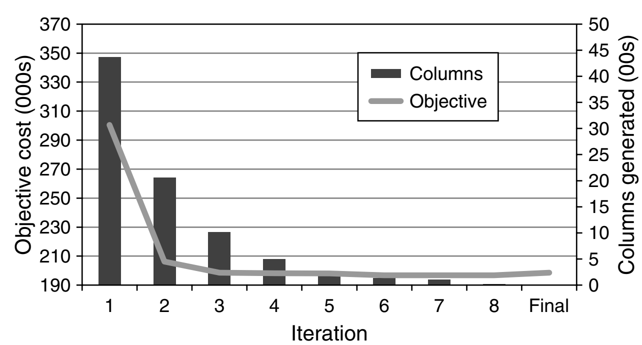

Figure 5: This chart shows the objective value and columns generated per iteration.

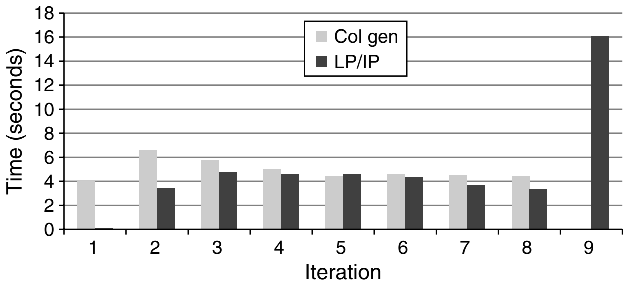

Figure 6: This chart shows the processing time for the LP-IP and column-generation phases at successive iterations.

settings. Note that the vertical processing-time axis is displayed in logarithmic scale. As we would expect, the fit line, given by $y = 1.5e^{0.01x}$ with $R^2 = 0.915$, shows a strong exponential correlation of processing time to the number of shipments.

Business Benefits

As we note in the *Implementation Details* section, the cost of outsourcing a shipment to a foreign carrier is approximately 60 percent more expensive than covering the shipment with a company driver. Since implementing the SHO, we have been able to realize a five percent decrease in our reliance on foreign carriers, resulting in a roughly three percent reduction in overall drayage cost. The other significant cost lever is company fleet utilization. This is typically measured either as loaded ratio [i.e., (miles driven with loaded container)/(total miles driven)], or as drays per driver day [i.e., (number of shipments moved)/(number of drivers working)]. Although these two metrics tend to move together, they are both important. Although loaded ratio is ignorant of freight characteristics that affect stop time at customer sites, it provides a reasonable measure of the geographic and timing quality of solution tours. However, drays per driver day provides a better measure of fleet throughput, because it accounts for all related activity; thus, it informs the relationship between fleet size and foreign carrier outsourcing. By either measure, fleet utilization has improved by about 10 percent since we implemented the SHO. This 10 percent improvement in fleet utilization enables the operations team to convert a large number of shipments from foreign carrier outsourcing to coverage by company and IC drivers without increasing the fleet size. Because of the significant premium paid to use foreign carriers, the corresponding annualized savings are in the range of \$8 to \$10 million. We also note that, in addition to this direct cost reduction, increased utilization provides a direct increase in driver pay, which results in lower driver turnover,, which in turn has a strong positive impact on driver hiring and safety-related costs.

Extensions

As we mention in the *Implementation Details* section, we currently use penalty adjustments to account for (1) repositioning empty containers between customer locations and storage facilities or rail ramps, and (2) the necessity for drivers to sometimes make intermediate stops to pick up, drop, or swap containers. We intend

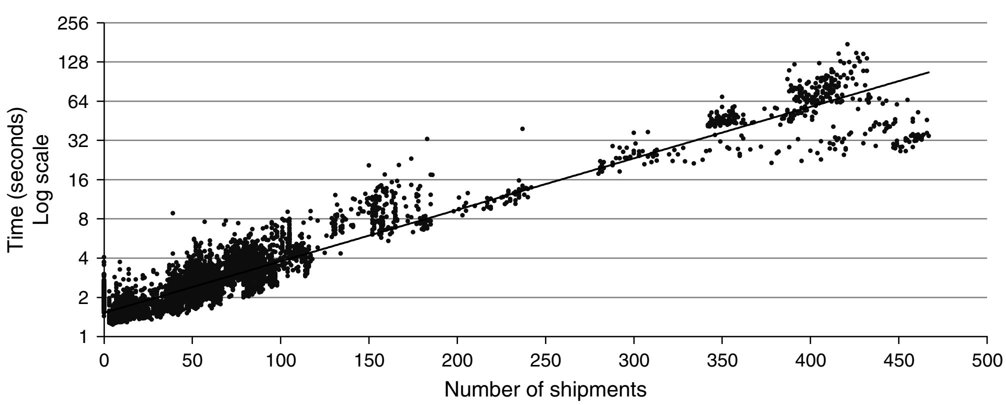

Figure 7: This graph shows processing times for a broad range of problems.

to improve the model to explicitly handle these issues by creating empty shipments. Although model enhancements are straightforward and we anticipate some computational challenges with the growth in problem size, the principal obstacle is in the work required to enable the TMS and data management components to provide accurate container location and status information at the required level of detail. We are also working on a separate model that will recommend optimal empty-repositioning shipments.

In the *Implementation Details* section, we also describe how the cost-minimization objective attempts to balance utilization of the company fleet (company drivers and independent contractors) with the use of foreign carrier resources. We are working on two closely related problems to determine the best mix of schedules (start times) for drivers and the optimal size for the company fleet for each rail-ramp region. The scheduling problem will consider variations in foreign carrier time-of-day usage patterns and recommend start-time adjustments to improve company fleet coverage on a day-to-day basis. The fleet-sizing problem must account not only for daily and seasonal variation in demand, but also for driver turnover rates and issues related to driver recruiting and hiring.

The current model is driven entirely on the basis of shipments that have become actualized in the TMS. Outbound drays (i.e., ramp to customer) are known well ahead of time (i.e., when the train departs the origin). In contrast, a number of inbound drays may either require or allow pick up the same day that the order is placed. Although these shipments will appear in the SHO re-solve process soon after they are made known, advance knowledge in an earlier planning cycle may lead to lower-cost solutions. We are investigating whether adding a shipment forecasting component and some stochastic characteristics to the model can add value.

Our current approach is somewhat naïve with regard to unexpected delays or unanticipated events. We use a combination of time padding for certain events or situations and conservative estimates for driving speed to provide solutions that have a measure of robustness and resiliency. The disadvantage of this approach is that we may execute routes and schedules that include unrecoverable and (in hindsight) unnecessary slack time, with no regard for how the risk costs may differ. Therefore, we are also looking at robust optimization approaches, such as Erera et al. (2009) suggest that proactively address robustness versus cost trade-offs.

Appendix A. Algorithm Details and Mathematical Formulation

As we allude to in the *Solution Methodology* section, we structure our problem as a set-partitioning integer program. We begin our formulation by defining D as the set indexing the company fleet drivers, a pool of generic foreign carrier drivers S as the set indexing the shipments, and T as the set of tours generated for all drivers. We also define T_k as the subset of T associated with driver k, T^j as the subset of T that covers shipment j, c_i as the execution cost (including preferences and penalties) of tour i computed during the column-generation process, and variables $x_i = 1$ if tour $i \in T$ is selected in the solution, 0 otherwise. We also account for the possibility that a given shipment j is not covered in the solution by defining slack variables s_j, which take the value 1 when shipment j is not covered, 0 otherwise, and a corresponding penalty p_j for not covering the shipment. With these defined, we formulate the following IP set-covering problem.

Objective function:

$$\min \left\{ \sum_{\mathbf{i}\in\mathbf{T}} c_i x_i + \sum_{\mathbf{j}\in\mathbf{S}} p_j s_j \right\} \tag{A1}$$

subject to:

$$\sum_{i\in T^j} x_i + s_j = 1 \quad \forall\, j \in S; \tag{A2}$$

$$\sum_{i\in T_k} x_i \le 1 \quad \forall\, k \in D; \tag{A3}$$

$$x_i \in \{0, 1\} \quad \forall\, i \in T. \tag{A4}$$

The objective function minimizes the sum of selected tour costs and penalties for uncovered shipments. Equation (A2) ensures that each shipment is either covered exactly once or incurs the appropriate penalty and Equation (A3) limits a driver to at most one tour.

Given a (generated) set of tours for a collection of drivers, the IP problem stated previously is relaxed to a LP problem by relaxing constraints (A4) to $x_i \in [0, 1]\ \forall\, i \in T$ and solved. Dual values are obtained for constraints (A2) and (A3). These respectively reflect the marginal cost of covering each shipment and the marginal value provided by each driver. These dual values are used to generate additional tours (new columns). This approach follows the general scheme that Erera and Smilowitz (2008) describe as a root column-generation heuristic. The tour-generation process is based on finding resource-constrained shortest paths in directed cyclic graphs. The graphs are constructed in the following manner: We begin with a generic base-level graph that captures all relevant information unrelated to a specific driver, including appointment windows, transit time and distance between stops or shipments, and associated costs (see Figure A.1). The nodes of the graph correspond to shipments and the directed arcs indicate a feasible precedence between shipments. If time-window and transit conditions

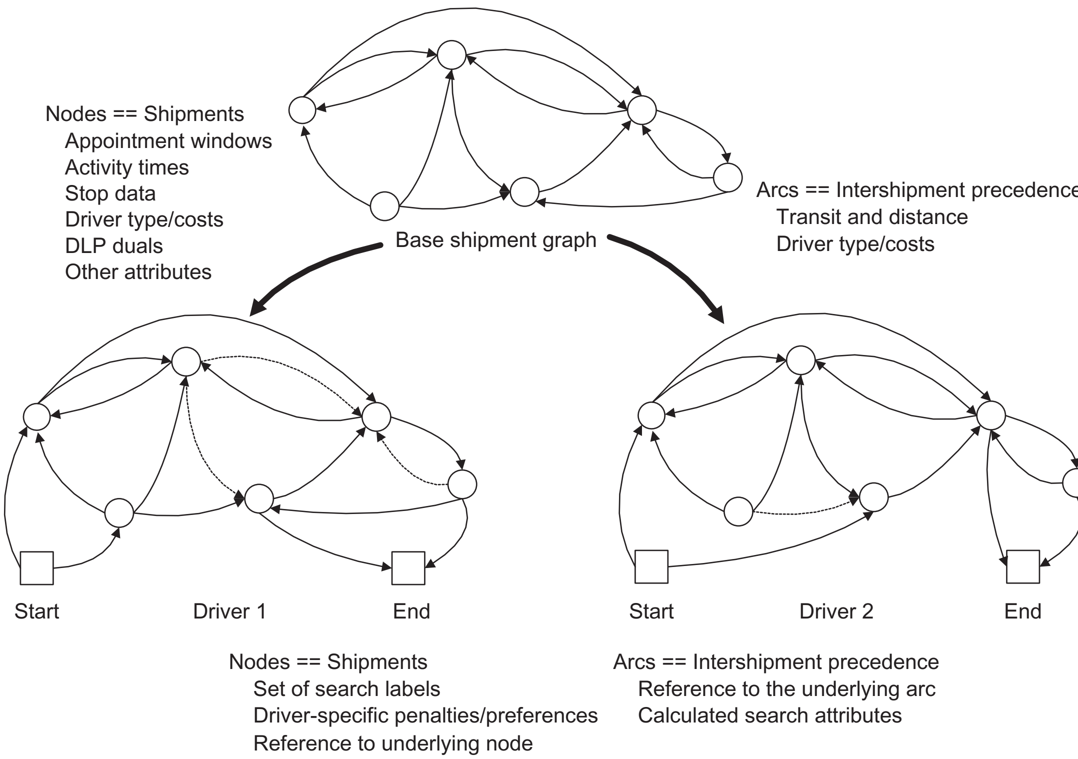

Figure A.1: This diagram illustrates the relationship between base shipment and driver-view graphs.

are favorable, two nodes are linked by arcs in both directions. The dual value from the corresponding shipment constraint in the current LP solution is also associated with each node. It represents the cost of covering (or penalty for leaving uncovered) the shipment in the LP current solution. Costs and transit times between the two shipments are associated with each arc. To support parallel processing and efficient memory management, we construct and maintain a second-level collection of driver-view graphs, one for each driver, that overlay the base graph with driver-specific cost and feasibility adjustments. These graphs capture additional driver-dependent information related to costs, preferences, schedule constraints, and starting and ending positions. The starting node also includes as a cost the dual value from the corresponding driver constraint in the current LP solution.

Given a collection of updated driver-view graphs, the column-generation step identifies a set of shortest paths (tours) with negative reduced cost. This is achieved via depth-first search and a corresponding labeling scheme. Labels associated with nodes in the graph are of the form (t, c), where t represents the elapsed time of a tour (path) at the completion of the current shipment (node) and c represents the corresponding reduced cost. These values are obtained by applying a simulation of driver movements, subject to the various relevant conditions (e.g., transit time, driver schedule, time-window constraints) from the delivery of the previous shipment to the delivery of the new shipment.

A set of labels corresponding to distinct paths that have been traversed to the node is maintained for each node. Whenever a node is searched, a new label is constructed, which captures the elapsed time and accumulated reduced cost of the (new) path to this node. If the new label is dominated (i.e., $t_{\text{old}} \leq t_{\text{new}}$ and $c_{\text{old}} \leq c_{\text{new}}$) by an existing label, the new label is discarded and the search backtracks. Otherwise, the new label is added to the current set and any labels that are dominated by the newly created label are removed from the set. Cycles are avoided by simply checking that no node is a self-parent in the search tree. We attempt to uncover the most promising tours first using a simple heuristic of exploring nodes according to the smallest ratio c/t among the labels. To limit the number of new columns entering the master problem, we restrict ourselves to those that have lower reduced costs than a maintained current list of best columns for the corresponding driver.

We summarize the overall algorithm as follows:

Step 1. (Initialization)

Construct driver graphs.

Apply business rules, cost, and transit information.

Generate initial set of columns.
(Cold start) Enumerate all single-shipment tours (or)
(Warm start) Remove infeasible columns from previous solution and enumerate new single-shipment tours.
Construct and populate the initial MIP-LP master problem.

Step 2. (Master problem)
Solve master problem as LP.
Record primal and dual solutions, including objective function value.
Update graphs with new dual values.

Step 3. (Column generation)
Perform depth-first searches (in parallel) on driver graphs, finding new negative-reduced-cost tours.
Add new columns to master problem.

Step 4. Repeat Steps (2) and (3) until objective value improvement is less than threshold (i.e., 0.1 percent).

Step 5. Solve the current master problem as an integer program and report solution.

One key advantage of maintaining separate driver-view graphs of the shipment network is to enable parallel generation of columns. For smaller regions, we use three processing threads; for the largest regions, six threads appear to be sufficient. At each iteration of Step 1, we select drivers (typically 10) for whom we generate new tours. Because the variable component of driver pay is sensitive to the quality of the tours assigned, we introduce randomness into the driver selection process to increase fairness across driver work assignments.

As we describe previously, we normally solve only the relaxed LP problem until the final step. For most problems, this works well because the integrality gap at the last iteration is usually less than one percent and the final IP problem solves reasonably quickly. However, for certain problems with many short-duration shipments, we can encounter long solution times for the final IP problem. For these problems, we have had very good initial success directly solving the IP problem at the intermediate steps, and then fixing the equal-to-one variables and solving the corresponding LP to obtain the necessary dual values.

Appendix B. Hardware and Software

Hardware	
Preprocessor and column generation	Xeon X7550 2.00 GHz, eight virtual CPUs, Linux VM, x86-64, 16 GB RAM.
LP and MIP solution	Xeon X7460 2.66 GHz, four virtual CPUs, Linux VM, x86-64, 12 GB RAM.
Software	
Operating system	Oracle Enterprise Linux v5.
Preprocessor code	Sun Java 1.6, 64-bit.
Column-generation code	Gnu GCC 4.1.2.
Integer and LP solver	CPLEX v12.4.

References

Baldacci R, Bartolini E, Mingozzi A (2011) An exact algorithm for the pickup and delivery problem with time windows. *Oper. Res.* 59(2):414–426.

Caris AN, Macharis C, Janssens GK (2008) Planning problems in intermodal freight transport: Accomplishments and prospects. *Transportation Planning Tech. 31*(3):277–302.

Desrochers M, Desrosiers J, Solomon M (1992) A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* 40(2):342–354.

Erera AL, Smilowitz KR (2008) Intermodal drayage routing and scheduling. Ioannou PA, ed. *Intelligent Freight Transportation* (CRC Press, Boca Raton, FL), 171–188.

Erera AL, Morales JC, Savelsbergh M (2009) Robust optimization for empty repositioning problems. *Oper. Res.* 57(2):468–483.

Ileri Y, Bazaraa M, Gifford T, Nemhauser G, Sokol J, Wikum E (2006) An optimization approach for planning daily drayage operations. *Central Eur. J. Oper. Res.* 14(2):141–156.

Namboothiri R, Erera AL (2008) Planning local container drayage operations given a port access appointment system. *Transportation Res. Part E* 44(2):185–202.

Savelsbergh M, Sol M (1998) Drive: Dynamic routing of independent vehicles. *Oper. Res.* 46(4):474–490.

Verification Letter

Don Aiken, Vice President, Intermodal Operations, Schneider National, Inc., Green Bay, WI 54306, writes:

"Since its earliest implementation, the Short Haul Optimizer project has been a critical tool in our dispatch tool box. As it has grown, matured and changed over the years, it continues to be a cornerstone of Intermodal's dispatch strategy. It is the key enabler of our 'Direct Dispatch to Truck' long term strategy.

"Thanks to the Short Haul Optimizer, benefits have been enabled in multiple parts of our organization, but the primary metrics impacted are loaded ratio, lower purchased transportation expense and higher driver productivity. We have been able to realize a five percent decrease in the reliance on purchased transportation, resulting in a roughly three percent reduction in overall dray cost. Loaded ratio and driver utilization have improved by 20 percent since the original implementation.

"We look toward even more key factor improvements as we incorporate optimization of our empty moves and leverage a large investment in container tracking systems. As a result, the short haul optimizer will continue to be a critical piece of our operations decision support and execution system for a profitable future."

Xiaoqing Sun works for Schneider National, responsible for designing and developing decision support tools. He also worked for Sabre Inc. American Airlines, and Chinese Academy of Sciences in the areas of capacity planning, revenue management, and computer integrated manufacturing systems. He has a PhD in industrial engineering, MSc and Bachelors of Engineering in systems control. His publications include articles in *IIE Transactions* and *Manufacturing Systems*.

Manish Garg works as a lead engineer at Schneider National. He has over eight years of experience in designing and developing decision support systems in transportation and supply chain domain. He earned a masters in industrial engineering from the University of Arizona.

Zahir Balaporia is the director of process and technology in the Intermodal Operations at Schneider National. He is responsible for the implementation of dispatch optimization and related processes. He has over 15 years of experience in network optimization, business process design, and operations analysis. He has degrees in industrial engineering and computer engineering.

Kendall Bailey is a senior engineer at Schneider National where he has worked on a variety of decision support systems for the last 17 years. He earned an MSc in applied mathematics from North Carolina State University where he was a recipient of a Department of Education GAANN fellowship. His interests include high performance, parallel and distributed computation, agile development, and open source software.

Ted Gifford is a distinguished member of technical staff at Schneider National. His current activities include development of algorithms and computational engines to support on-line decision support systems, supply chain optimization, and tactical planning models; and the application of statistics and predictive analytics to problems in vehicle maintenance and human resources. He has an MS in operations research from Georgia Institute of Technology, and an MA in mathematics from the University of California, Berkeley.