IMPLEMENTING REAL-TIME OPTIMIZATION MODELS: A CASE APPLICATION FROM THE MOTOR CARRIER INDUSTRY

WARREN B. POWELL

Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544, powell@princeton.edu

ARUN MARAR

Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey, marar@alumni.princeton.edu

JACK GELFAND

Department of Psychology, Princeton University, Princeton, New Jersey 08544, jjg@princeton.edu

STEVE BOWERS

Mark VII Logistics, sbowers@markviiww.com

(Received April 2000; accepted August 2001)

Online models for real-time operations planning face a host of implementation issues that do not arise in more strategic arenas. We use the seemingly simple problem of assigning drivers to loads in the truckload motor carrier industry as an instance to study the issues that arise in the process of implementing a real-time dispatch system. Although the project was moderately successful, our focus is not on documenting the benefits, but rather on summarizing the challenges that arose. The most significant theme running through the implementation hurdles we encountered was the lack of information available to the model. Computers are very good at processing vast quantities of information; humans are very good at challenging the information that is in the computer and augmenting computer-provided data with head knowledge. Our study includes a careful comparison of actual decisions with model recommendations, using a six-month database of actual transactions. This comparison is the first we have seen of its kind and provides the most rigorous evaluation of an online dispatch model that we have seen. Although the model was well used, the results demonstrate that significant improvements could have been obtained if the level of model utilization had been even higher.

INTRODUCTION

With the explosion of information technologies over the last decade, collecting information can seem a little like drinking from a fire hose. Increasingly, the problem is not finding information, but using it effectively. Optimization models offer the promise of a powerful technology for processing information as fast as it arrives, but they often struggle in actual field implementations. In this paper, we take a disarmingly simple problem (the load-matching problem of truckload trucking) and try to understand why optimization models in this setting have not met expectations. There are literally thousands of truckload motor carriers, and hundreds are large enough to really benefit from an online dispatching tool. Yet as of this writing, only a few dozen have actually implemented these systems, with mixed success. As is often the case, there are several reasons why optimization models have struggled to develop a foothold in this large and important dimension of the logistics industry. But somewhat surprisingly, one of the most important conclusions that emerged was that they suffer from a lack of information.

Over a two-year period, Burlington Motor Carriers, a leading truckload motor carrier with over 1,200 trucks (at the time), worked with CASTLE Laboratory at Princeton University to develop and implement a real-time optimization model to help planners with the task of assigning drivers to loads. We used this project not only to develop and implement what at the time was a state-of-the-art optimization model for managing drivers, but also to study the process of implementing a real-time optimization model in an effort to learn why people do not follow model recommendations, and how the process of implementing models

Subject classifications: Transportation: model implementation issues.
Area of review: TRANSPORTATION.

0030-364X/02/5004-0571 $05.00
1526-5463 electronic ISSN

Operations Research © 2002 INFORMS
Vol. 50, No. 4, July–August 2002, pp. 571–581

can be improved. Many of the lessons from this project (the model was still running two years after project completion) are likely to be applicable to the implementation of real-time optimization models in other settings. Some of these lessons are familiar ones in a fresh setting, but the special role that information plays in the success of an optimization model appears to offer some new insights.

There is a long history of trying to implement operational models for vehicle routing and scheduling. Bell et al. (1983), Brown and Graves (1981), Brown et al. (1987), and Powell et al. (1988) have all attempted to implement computerized optimization models for real-time operations. Dynamic routing and scheduling problems have long been recognized to provide special challenges in terms of algorithmic strategies (see, for example, the discussions in Psaraftis 1988, 1995), and some have even attempted to use artificial intelligence instead of mathematical optimization (Bagchi and Nag 1991). A number of studies have demonstrated the benefits of real-time scheduling systems (Regan et al. 1995, 1996, 1998, 1999; Powell et al. 2000b). In addition, modern communication technologies make the use of optimization algorithms even more attractive (Schrijver 1993). Taylor et al. (1999) provides a summary of the different approaches that have been used to solve the load-matching problem for truckload trucking. Although considerable attention has been put into handling the combined problem of known and forecasted customer demands (Powell et al. 1988, Powell 1991), the most common approach in engineering practice is to solve sequences of static, largely myopic, assignment problems. This is the approach we followed in this project.

Despite the tremendous potential and apparent successes, there are many failures that are rarely documented. This paper summarizes the results of a moderately successful project. However, our focus is an analysis of the problems that arose rather than a summary of the benefits. We believe that many of our experiences will arise in other operational applications of optimization.

1. THE LOAD-MATCHING PROBLEM

The load-matching problem of truckload trucking seems, on the surface, to be surprisingly simple. At any one point in time, there may be several hundred drivers that are available to be assigned to a load (or are expected to become available in the near future), and several hundred loads that need a driver to pick up the load and carry it to the destination. The challenge is finding the right driver to assign to the right load. The most obvious objective is to minimize the total number of miles a driver has to move empty to cover the load (carriers typically pay by the mile). But there are a number of other goals the carrier must juggle. Loads have time windows in which they need to be picked up and delivered. Loads may require special driver skills (such as experience crossing a border, or the training to handle hazardous materials) or equipment (longer trailers, refrigerated trailers, or air-ride shock absorbers). Perhaps the most difficult challenge is the problem of getting drivers home. A long-haul truckload driver may be away from home for two weeks or more. After a period of time on the road, a driver will request to be put on a load that gets him close to his home (although drivers vary in their willingness to be away from home).

Other challenges face the truckload dispatcher. Foremost is a highly dynamic environment. Shippers typically call in new requests one or two days in advance but sometimes make requests for the same day. Drivers who are in the process of moving a load will have an estimated time of arrival when they next become available for a new load, but there can be a lot of uncertainty in these times—partly because of variation in travel times, but often because of delays incurred while unloading the truck at the destination. A planner may preassign a driver coming inbound to a region to pick up a load but has to make contingencies if the driver does not become available when we expected.

Humans use a form of cache memory called working memory for processing information. Because people are limited in the amount of material they can process simultaneously in working memory (Card et al. 1989), carriers deal with the challenge of juggling all this information by dividing the country into regions of manageable size. Planners are evaluated based on their ability to cover loads on time within their region and the number of empty repositioning miles drivers incur within their region. Their computer screens are designed to show drivers and loads in their own region. Not surprisingly, then, dispatch decisions can exhibit a boundary effect, where a driver on one side of a regional boundary is invisible to a dispatcher who has an uncovered load on the other side of the boundary.

In addition to these systematic problems, there is the usual flow of human oversights. It is easy to assign a driver to a load without enough available hours to get to the destination on time. It is easy to assign a driver who is 30 miles away, missing the driver who is 20 miles away but in another region. Or, the planner may pick the driver who is 20 miles away, missing the driver 30 miles away who needs a load to get home on time.

In short, the sheer volume of information that needs to be handled seems to be an obvious application for computerized decision support. However, as the project demonstrated, a significant dimension of the project related to the difference between the data in the computer and information that was known only to the dispatcher who would speak directly to the driver, as well as to the customer service agent in contact with the shippers. As a result, the dispatchers would override the model between 20 percent and 40 percent of the time. Not all these user overrides reflected missing information, but the difference between what the human knew and what the computer knew was a significant issue and was often used to explain discrepancies between model recommendations and actual decisions. Several strategies were used to deal with the difference between model information and dispatcher information, but

one of the most important was the ability of the model to respond in real-time to actual decisions.

A separate but significant dimension of the project arises from the usual challenges of balancing competing objectives from different perspectives. Truckload motor carriers have to minimize costs (in this industry, the total empty miles is a major cost element), maximize customer service (picking up and delivering loads on time), and manage drivers effectively (avoiding long waits for a load assignment and returning drivers to their home on a regular basis). Managing competing goals is itself a challenge. However, our project highlighted the importance of measuring these objectives from three different perspectives: the model (which required that all the issues be quantified into a single cost function), management (which focused on a series of statistics which measured performance), and the dispatchers (who used their own rules and patterns of behavior to make decisions).

2. THE OPTIMIZATION MODEL

The load-matching problem can be formulated as an assignment problem between drivers and loads, where the driver is modeled as a driver/tractor combination, and the load is a trailer filled with freight. We ignore the problem of balancing empty trailers. The model requires the following notation.

Resources and tasks:

$\mathscr{R}_t$ = Set of drivers available at time t.

a_r = Attribute vector of driver $r \in \mathscr{R}_t$.

$\mathscr{L}_t$ = Set of loads known to the system at time t.

b_l = Attribute vector of load $l \in \mathscr{L}_t$.

Decisions:

$\mathscr{C}^D$ = Decision classes
= {reposition empty(e), move load(l), do nothing (*hold*)}.

$\mathscr{D}^{(c)}$ = Set of decisions in class $c \in \mathscr{C}^D$.

$\mathscr{D}^{(e)}$ = Set of locations a driver may reposition empty to.

$\mathscr{D}_t^{(l)}$ = The decision to move to a load, pick it up, and complete the move. There is an element of $\mathscr{D}_t^{(l)}$ for each element in $\mathscr{L}_t$. Thus, $\mathscr{D}_t^{(l)}$ is the set of loads available at time t.

$\mathscr{D}^{(hold)}$ = A set containing a single element representing the action to "do nothing."

$\mathscr{D}_t = \mathscr{D}_t^{(l)} \bigcup \mathscr{D}^{(e)} \bigcup \mathscr{D}^{(hold)}$
= Set of all possible decisions

$$x_{rdt} = \begin{cases} 1 & \text{If decision } d \text{ is applied} \\ & \quad \text{to driver } r \text{ at time } t, \\ 0 & \text{Otherwise.} \end{cases}$$

Information processes:

$\mathscr{C}^I$ = Set of dynamic information classes
= {Drivers, loads, decisions}.

$\mathscr{E}^{(c)}$ = Information elements in class $c \in \mathscr{C}^I$.

$\mathscr{E}_t$ = Set of information elements arriving in time t.

κ_{et} = An update to information element $e \in \mathscr{E}^{(c)}$ at time t.This can be changes in a driver status, new loads, changes in times, and decisions made by planners.

$\kappa_t = \{\kappa_{et}, e \in \mathscr{E}_t\}$.

K_t = Knowledge base at time t
$= \{a_r, r \in \mathscr{R}_t; b_l, l \in \mathscr{L}_t\}$.

$U^K(K_t, \kappa_t)$ = Knowledge update function, where $K_{t+1} = U^K(K_t, \kappa_t)$.

In our representation, a decision (e.g., "assign driver r to load l" or "move driver r empty to region n") arrives as an information class, and we use the update function U^K to describe how this information changes the system.

Costs:

$\mathscr{I}$ = Set of cost indices
= {empty miles, on-time pickup performance, destination delivery performance, routing drivers through home, serving a load}.

$c_{it}(r, d)$ = Value in terms of cost index $i \in \mathscr{I}$ of acting on a driver r with decision $d \in \mathscr{D}_t$ at time t. If $d \in D^{(l)}$, then we generally write $c_{it}(r, d) = c_{it}(a_r, b_{l_d})$ where l_d is the load associated with d.

w_i = Weight given to the ith cost component.

$c_{rdt}(w)$ = The total cost of acting on a driver r with decision d at time t given weight vector w
$= \sum_{i \in I} w_i c_{it}(r, d)$.

The load-matching problem is solved as a sequence of myopic assignment problems using the knowledge base at time t:

$$\min_x F(x|K_t) = \sum_{r \in \mathscr{R}_t} \sum_{d \in \mathscr{D}_t} c_{rdt}(w) x_{rdt}, \tag{1}$$

subject to

$$\sum_{d \in \mathscr{D}_t} x_{rdt} = 1 \quad r \in \mathscr{R}_t, \tag{2}$$

$$\sum_{r \in \mathscr{R}_t} x_{rdt} \leqslant 1 \quad d \in D_t^{(l)}, \tag{3}$$

$$x_{rdt} = (0, 1). \tag{4}$$

System dynamics are handled using

$$K_{t+1} = U^K(K_t, \kappa_t). \tag{5}$$

Equation (5) compactly summarizes the evolution of information (or knowledge) for the system. We use this representation to capture updates to what we know about assignment costs, new loads, or changes in the attributes of a driver or load.

The optimization problem posed by our model is a simple assignment problem. Because it is formulated in terms of what we know about drivers, loads, and arrival times, it needs to be updated each time our information changes. Fortunately, the algorithms are quite fast, and optimality is assured as long as we remind ourselves that this is a myopic model (no forecasts of future customer demands are included). Note that it is possible in principle that we will be able to assign a driver to a sequence of two or more loads that are already known to the system. However, this segment of the trucking industry is characterized by relatively long loads, so a driver often does not complete a task for a day or more into the future. Because a lot can change before a driver completes a load, the standard practice is to assign a driver to a single load. Only after this assignment is made would we forecast the driver's arrival status in the future, and then make the driver available (in the model) for an assignment to another load (in the future).

On the surface, the benefits of a model such as this seem tremendous. To start, we are no longer constrained by the need to assign drivers to loads within a particular region; the model can easily consider loads that are outside of a driver's region. The model can also simultaneously consider a number of factors. We list four factors in our model, but a production system may consider more than a dozen. The cost function is extremely flexible, and the company may easily adjust the weights w_i, thereby controlling the importance of different objectives. The algorithm is fast and reliable.

Perhaps the most appealing dimension of the optimization model to senior management at a company is the ability to directly and uniformly adjust management goals. If driver turnover is a problem, a higher weight can be placed on getting drivers home. If customer service is not up to market standards, a higher weight can be allocated to on-time service. Without a model, management is forced to send out memos encouraging the planners to shift priorities, and then monitor individual performance. Of course, the real hope is that the company will simply achieve better service, getting drivers home more reliably, all with lower empty miles.

There is a cottage industry of software vendors who have attempted to market this technology. Dr. Richard Murphy was the first to formulate the load-matching problem as a network for one of the major truckload carriers, and he later started a company to sell this and other products. The Innovative Computing Corporation attempted to solve the same problem using artificial intelligence. Micromap® was a product that was first developed as part of a senior thesis at Princeton University (Cape 1987) and then sold through the Princeton Transportation Consulting Group (founded by Cape). A more thorough description of this model can be found in Powell (1991, 1996) and the references cited there. None of these early models captured more than a tiny fraction of this extremely large market, and while some companies continue to use the technology with apparent satisfaction, it was more often the case that companies were turning off or ignoring the system several years after implementation.

In 1995, Burlington Motor Carriers contracted with CASTLE Laboratory at Princeton University to implement a real-time optimization model for matching drivers and loads. Although a more advanced algorithm was used that allowed the model to assign a driver to a sequence of two or more loads (see Powell et al. 2000a), the modeling strategy was basically the same as that used in the earlier models. The goal of BMC was, of course, to obtain an advanced load-matching system. Our goal was to study the implementation process to see if we could identify the core causes of why these models were not being embraced on the dispatch floor. The research team represented a combination of skills: operations research (Powell), artificial intelligence (Gelfand), the vice president in charge of operations (Bowers), and the vice president in charge of information technology (Mr. Robert Lamere). The original optimization model was developed by Derek Gittoes, and the implementation at Burlington Motor Carriers was handled by Arun Marar.

3. MEASURING PERFORMANCE

The previous section follows the standard methodology for OR models by formulating an objective function. We have followed standard notational convention when we wrote our objective function (in Equation (1)) when we represented our objective function as $F(x)$. Over time, however, we have learned that there are three objective functions at work, which we conveniently refer to as F, G, and H:

F—This is the objective function that the computer tries to minimize (or maximize). It usually represents a combination of real costs and a series of artificial bonuses and penalties that are used to guide the model to produce an "acceptable" solution.

G—These are the management goals. Virtually every organization is managed through a series of statistics expressed in the form of goals that can be represented as the family $G = \{G_g\}_{g \in G}$. It would seem natural to form a single goal function of the form $G = \sum_{g \in G} w_g G_g$, but it is our experience that companies do not do this; instead, they focus on the "goal du jour," emphasizing the goal that is most out of line with expectations.

H—This represents user happiness, or more precisely, the degree to which the users accept the recommendations of the model. At BMC, we had an explicit measure of user compliance.

Although we sometimes joke that H represents the width of the user's smile, it is actually a hard, quantifiable statistic that we watch closely. Specifically, let D^a be a set of actual decisions made by users, and for a given decision $d \in D^a$ (representing the assignment of a driver to a particular load), let r_d be the driver involved in the decision, let l_d be the load, and let t_d be the time the decision was made. At the time the decision was made, we can compute a reduced cost for the decision, given by

$$\bar{c}_{r_d,d,t} = c_{r_d,d,t} + \pi_{l_d,t} - \pi_{r_d,t} \quad (6)$$

where π_r and π_l are, respectively, the dual variables for driver r and load l (obtained directly from the optimization model). If the user accepts the top model recommendation, then we are assured that $\bar{c}_{rdt} = 0$ (because of ties, this may also be true for decisions other than the top model recommendation). It has been our convention to assume that the user agrees with the model if he makes a choice where $\bar{c}_{rdt} \leqslant 10$. Our "happiness function" H, then, is the percentage of all dispatches where $\bar{c}_{rdt} \leqslant 10$.

The $F/G/H$ trio of measures is critical and the measures need to work together. The behavior of the optimization model is controlled by the design of F. We impact the company only if H is high, and management evaluates our performance only in terms of G. A common mistake in modeling is when we try to design F to improve the G statistics at a time when the users, whether correctly or not, continue to pursue established patterns of behavior. For example, in trucking, it is very easy for a dispatcher to be overly responsive to a driver talking on the phone ("I really need to get home!") or a shipper on the phone ("I really need this load picked up, and now!") compared to the memo that your manager gave you on Monday emphasizing reductions in empty miles. Alternatively, a dispatcher may choose not to satisfy a short-term goal such as making a pickup time to satisfy the overall goal of that memo about empty miles. If the supervisor subsequently reprimands that particular dispatcher for a service failure, then he or she is likely to ignore the overall goal in favor of the short-term goal. This is an example of the principal-agent problem causing the agent, in this case the dispatcher, to become risk averse in certain decisions contrary to the overall goals of the principal, in this case the supervisor. These examples stress the need to review and set rewards based upon overall performance rather than specific individual events.

4. THE IMPLEMENTATION PROCESS

Because our software was derived from previous research, we were able to adapt the model to the new application fairly quickly. The company's in-house systems staff developed the communication between the Unix workstation that ran the model and the AS/400 that handled all the data processing. Updates to the AS/400 would be sent to the optimization model in a continuous stream. After each model cycle (typically a few seconds), the Unix workstation would then send back a set of recommended assignments. Recognizing the need for flexibility, the model would send back a ranked list (the rankings were based on dual variables) of loads for a particular driver, or drivers for a particular load. At that time, it would take anywhere from 30 seconds to as much as two minutes for a planner to see new recommendations that reflected new information.

4.1. Collecting Feedback

Once the system went into production, we put in place a weekly conference call to answer questions and discuss modeling issues. We quickly found, however, that very little was accomplished with these phone calls. Typically, one of the planners would give a situation and ask why the model made the recommendation that it did, and we would offer various theories as explanations. Because the situation would have occurred several days ago, we had no means of checking our theories. As a result, the model did not change, the data did not improve, and the user remained in the dark.

In response to this problem, we developed the ability where the user could request a data dump any time something came up that looked odd. We would then bring this data set up to Princeton and load it into a special diagnostic tool that we had developed (dubbed "Pilotmatch"). As a result of this tool, we were able to provide specific, informed explanations to questions that were raised. In some cases, this resulted in adjustments to the data feeds or changes in the model. Most of the time, however, we were simply explaining why the model made a recommendation. The only problem was that our explanation would come 24 hours after the decision was made.

After working with this process, it became clear that answers to queries had to be almost immediate, which meant that they had to be on site. We had been developing a diagnostic tool (called Pilotview) that we were able to install at the company. However, the tool was never really adopted, and we were not able to get a true on-site diagnostic process in place. Part of the problem was that the tools were relatively primitive, which meant that a floor dispatcher lacked the training to use it easily and effectively.

During the implementation process we made several trips to the carrier to sit with planners, watch them make decisions, and answer questions. It was sometimes amazing how little we would learn from these trips. Illustrating a human version of the Heisenberg uncertainty principle, observing the process of decision making also distorted the process itself. What people did as we leaned over their shoulders was not what they did after we left.

As an alternative, we began a process of collecting data in the background. Over a six-month period, we would collect (for most days) the initial snapshot of drivers and loads, and then every single transaction that came to the model. These transactions contained new loads, updates to the driver status, and the decisions actually made by the planner. Using these datasets, we were able to completely

resimulate an entire day of dispatching using different variations of the model. We could also stop the model at any point to examine a decision to analyze discrepancies. This special tool, which was dubbed the transactional data analyzer (TDA), proved to be an exceptionally valuable diagnostic device. Our most productive use of the tool was to adjust the weights on different objectives in an effort to increase the number of times the model would agree with the users. This proved to be far more effective than adjusting the weights and depending on oral feedback.

4.2. The User Compliance Problem

User complaints about the model were, of course, a mixture of substantive objections combined with vague complaints that could easily be categorized as "but I just don't do it that way!" Of course, part of the goal of the model was to change exactly this behavior. As a result, management put in place a statistic that monitored individual user compliance, which were then tied to monthly bonuses.

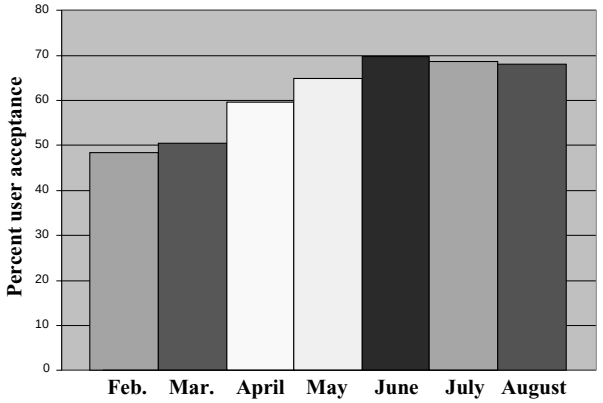
With this management incentive, user compliance rose from below 50 percent to more than 70 percent over a five-month period, as illustrated in Figure 1. Notably, there was wide variation among individual users. When the company was scoring 70 percent, some users were using the model over 85 percent of the time, while others remained below 60 percent.

The issue of user noncompliance raised an interesting question concerning the algorithm. Simply put, if the users did not use all the suggestions of the model, then in what way was the optimal solution to our assignment problem actually optimal? This question spawned a research project described in Powell et al. (2000). This study assigned drivers based on the formula

$$\max_d \{c_d + \alpha(\pi_{l_d} - \pi_{r_d})\}, \tag{7}$$

where d is a specific driver to load decision, r_d is the driver that would be assigned, and l_d is the load the decision would assign the driver to. π_r and π_l are the dual variables of driver r and load l, and α is a "dual discount factor."

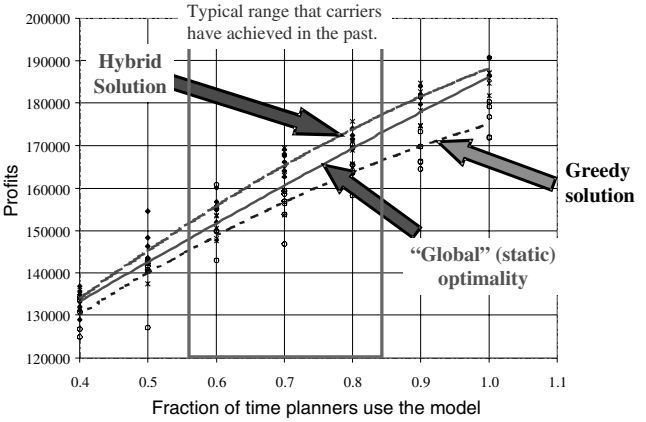
Figure 1. Evolution of user acceptance during initial implementation period.



If $\alpha = 0$, then we have the effect of greedily assigning a driver to the "best" load; $\alpha = 1$ is (roughly) the same as doing exactly what the network model recommends, while $0 < \alpha < 1$ represents a range of intermediate solutions. We then simulated each strategy for four weeks assuming different levels of user compliance. A compliance level of 1.0 meant the users picked the best recommendation of the model; lower levels meant that the user considered each recommendation in order, and picked it with fixed probability. The results, shown in Figure 2, indicate that the model can significantly outperform users if they do exactly what the model says (these results also assume that all the data in the computer is correct). When the user compliance drops below 70 percent (the range that we were working in), then the value of the network model (referred to misleadingly as the "globally optimal solution" by some software vendors) over a simple greedy solution drops significantly. Interestingly, we found that using $\alpha = 0.75$ in Equation (7), which means that we use a discounted version of the dual variables, performed noticeably better than both methods. This analysis provides an indication of why carriers who have about 70 percent user compliance (which is considered normal) do not notice dramatic benefits from the use of a model such as this.

One of the challenges we faced in tuning the model to improve user acceptance is that each user is different. The situation is not unlike the challenge of fitting a line through a set of points, where the line is the model and each point is a different person. Interestingly, as much as management wants to achieve high levels of model utilization, the same managers are typically unwilling to let us "customize" the model for each user. Also, most companies have been unwilling to bring anything other than modest pressure on users to adopt model recommendations over their own judgment.

Figure 2. The effect of user compliance on the performance of a dispatch system using global, local, and hybrid assignment strategies.



Note. The graph shows simulated profits as a function of user compliance (the probability of accepting a model recommendation).

4.3. The Response Time Problem

Perhaps the most frustrating dimension of the project was the slow response as perceived by the user. The fact that the algorithm could update the model in a second was of little concern to users who had to wait a minute to see the effects of data entered in the system reflected by the model. The problem lay in the design of the flow of information between the host database and the model. Although this dimension was outside of the scope of the model development, it proved to be a critical dimension of the overall system.

The response time challenge arose from the need to respond to information as it came to the system. Frequently, a dispatcher would learn something new about the driver while the driver was on the phone (the driver wanted to go home, or would be arriving earlier than expected). The problem was that the dispatcher had to make a decision while the driver was on the phone. Of course, it was generally the case that at least one of the recommendations in the recommended list would be fine, but if not, then a decision would have to be made while the driver was on the phone. Even more problematic was when a shipper would call with a new load to be picked up the same day. Often, the planner wanted to assign a driver right away before even accepting the load. It was not unusual for us to see a driver assigned to a load that the model did not even know about when the decision was made.

4.4. The Management Problem

Ultimately, there develops a tension between the users—who have their own way of making decisions—and management, which is struggling to get the benefits of the model. In between is the model that is applying objectives designed by management but is then supposed to be something the users agree with. In theory, the original motivation behind the project was the consensus among management that the planners were not making the right decisions. This raises a classic catch-22 situation: The model is successful if the users agree with the model, but the users often do not follow the model; and if the model did exactly what the users wanted to do, then the model would not be producing any changes.

In an earlier project at a different company, a vice president of operations noted, "Sometimes when you can't change the people, you have to change the people." In practice, this is quite hard to do. No model is perfect (and this model, based on concepts from the early 1990s, had several major limitations), which makes it difficult to fire people for not following a model, even if it would, on average, outperform the human. In the choice between who are you going to fire, the model or the human, the model usually loses.

5. MEASURING PERFORMANCE

With any optimization model, the question always arises: What was the impact? A partial answer to this question lies in the calculation of empty miles, a closely watched statistic in the trucking industry. During the period that the company emphasized model utilization, we received reports that indicated that empty miles had dropped between 5 percent and 10 percent. In a commodity business with tight profit margins, this translates to significant savings, adding as much as a point to the company's operating ratio.

The question we were most frequently asked was, How well would we have done if we had just followed the model? The problem with this question is that it ignores the fact that once we make a decision that is different from what was made in the real world, we start going down a path with different outcomes and options. It is possible to compare an actual decision to a simulated decision at a point in time, but then carrying the effects of those decisions forward in time creates a problem.

As an alternative, we used our transactional data analyzer to compare each actual decision to what the model would have done at the same point in time. The TDA system allowed us to completely resimulate an entire day's dispatching, comparing each decision, transaction by transaction, to what the model would have done with the same information. To ensure the fairest possible comparison, we collected three sets of statistics for each load in the system. Thus, for a given load we would compute the empty miles (for the driver who was assigned to the load, and the driver the model would have assigned to the same load), whether the load was being picked up on time, and whether the load was able to route the driver to his home.

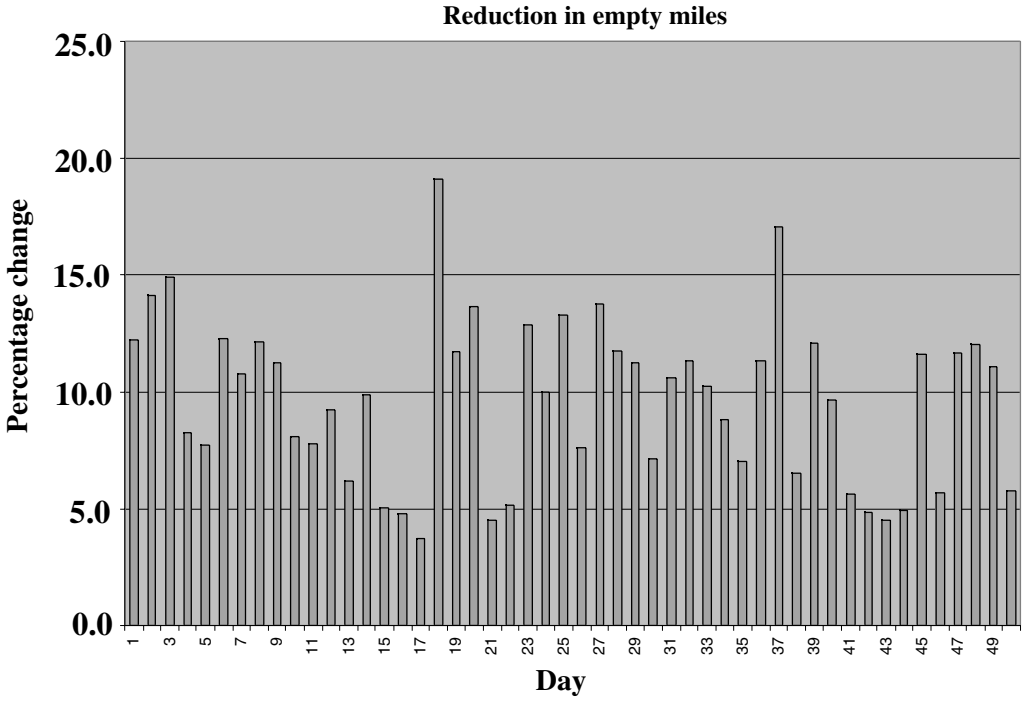
The results of these comparisons are shown in Figures 3, 4, and 5, which show, respectively, the improvement in empty miles, on-time service, and routing drivers through home. Each bar reflects average performance over a single day, usually consisting of several hundred dispatches. It was not unusual for the actual dispatch to be better than the model recommendation on one or two dimensions (or even all of them for a single dispatch), but rarely did the actual decisions outperform the model recommendations in any one dimension when averaged over all the loads for a given day.

The experiments indicate that had the dispatchers followed the model, the empty miles would have been reduced between 5 percent and 10 percent, on-time service would have improved between 1 percent and 3 percent, and routing drivers through home would have improved between 1 percent and 4 percent.

These comparisons represent, we believe, the most careful, rigorous evaluation of the differences between what a model would have done versus what actually happened.

Given that it is impossible to run two identical trucking companies side by side, and given the inherent errors in comparing performance of a company over two nonoverlapping periods of time (for example, before and after the model implementation), our analysis is the most controlled study that we can envision for evaluating the potential impact of a model for real-time planning.

Figure 3. Percent reduction in empty miles for each day.



6. IF IT'S SO GOOD, WHY DON'T THEY USE IT?

At this point, we can legitimately ask: So why doesn't the carrier simply follow the model? The reasons are somewhat subtle and offer insights that, we believe, are likely to arise in the context of many uses of optimization models in a real-time setting.

6.1. It's the Data, Stupid!

Models for load matching would work best if we knew everything we needed about the driver, we employed dispatchers who followed instructions perfectly, and used drivers who did exactly what they were told (and arrived on time) and shippers who would tell the carrier all of their

Figure 4. Improvement in on-time performance for each day.

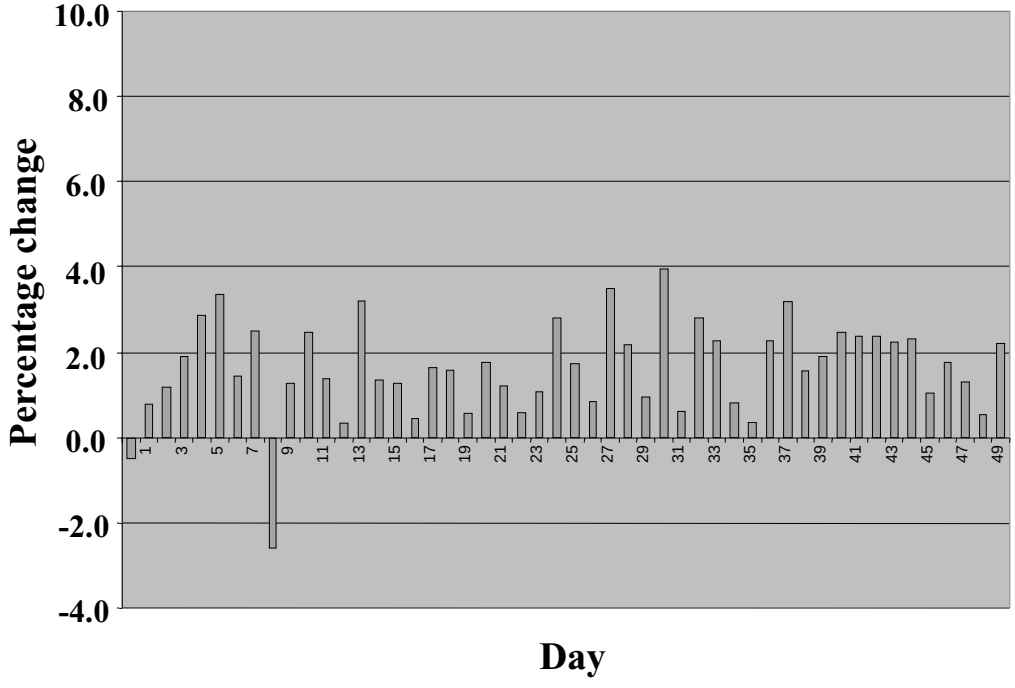
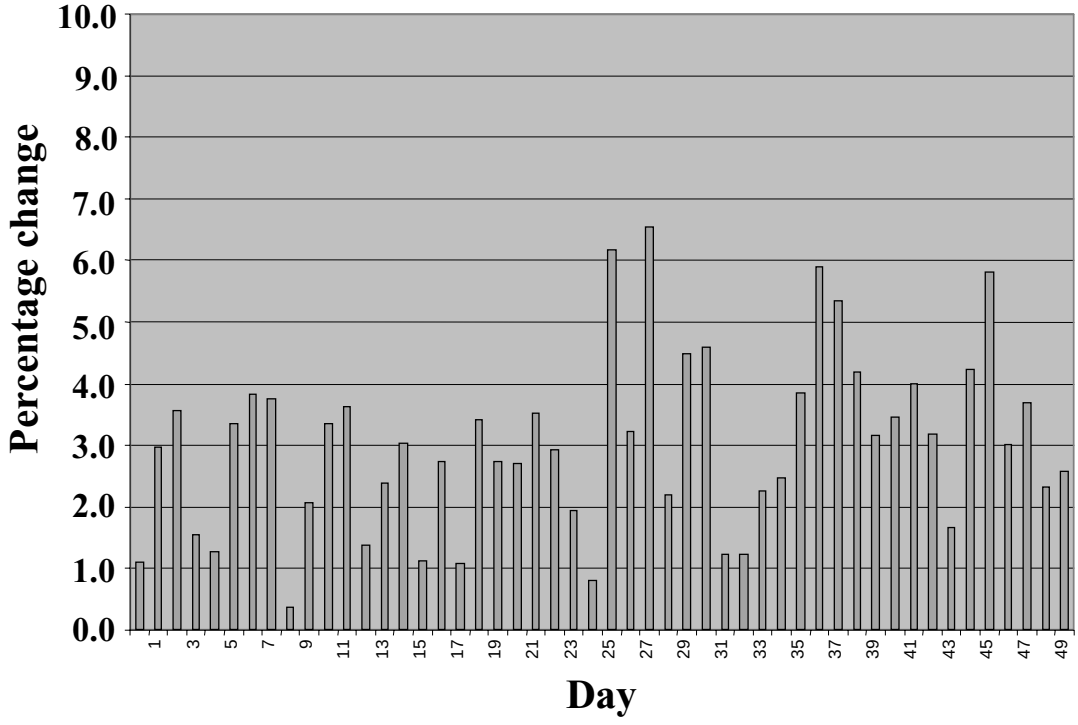


Figure 5. Improvement in routing drivers through their home.



requests several days in advance. The problem with optimization models is that they can be very good at giving you the right answer if you can just give them all the right data. If you sit and watch a dispatcher at work, you quickly realize that they are spending *most* of their time collecting and verifying data. The thinking (which is admittedly imperfect) does not take much time. The situation is reminiscent of the old joke about the engineer, the scientist, and the economist lost at sea with a can of food without a can opener. After the engineer and the scientist offer technical solutions, the economist (famous for developing models with strong assumptions) grabs the can and remarks: "Let's assume we have a can opener!" In our version of this joke, the operations researcher is arriving to a complex problem that is awash in bad data, and then solves the problem by assuming that the data are correct.

The biggest problem facing the use of an optimization model in this setting is that not only are the data in the computer imperfect, they are incomplete. A shipper, for example, may ask that a load be picked up in the morning before noon, when his dock is not as busy. The computer system has no way of recording a soft request such as this, so the load is entered as being *required* to be picked up in the morning. The computer model now treats an afternoon pickup as a service failure, when in fact all the shipper was trying to do was express a preference. Because senior management has insisted on a high penalty for service failures, the model now recommends that a driver run empty overnight 100 miles to pick up this load in the morning, whereas a sensible planner would take the driver coming inbound to this location arriving the next day (but not until the afternoon, which would produce an apparent service failure).

Computers are not the only ones suffering from data problems. The planners themselves struggle with what is going on inside the driver's head. The planner would like to assign the driver to a load going to Minnesota. The driver wants to go to New Orleans, where it is warm. After moving the load to Minnesota, the driver returns home and quits (costing the company several thousand dollars in recruiting and training expenses). The problem is sufficiently serious that at least one motor carrier has started allowing the drivers to pick their own loads.

The challenge is: Who should make a decision? The planner has information the computer does not (and the driver has information the planner does not). On the other hand, the computer is able to bring a global perspective to the problem that the planner is unable to do. The planner can bring a regional perspective to the problem that the driver is unable to do. Is it better to have good local information or good global information? When we consider the value of using an optimization model part of the time (Figure 2), we see that the value of the global perspective of the model is not that high. However, the balance between the value of local information versus global perspective, we believe, will remain an open question for some time.

6.2. Mixed Mode Reasoning

Humans have a particular strategy for solving problems. Stated simply, they start by identifying and prioritizing goals, and then apply strategies for achieving these goals. The identification of strategies for solving a problem is

a form of pattern recognition; identifying the state of the system, and then choosing actions that the planner has learned work well to solve the problem (Klein and Calderwood 1991). In the load-matching problem, this may be nothing more than finding the most important load to be moved, and then finding the best driver for that load. However, other situations are more complicated. A driver needs to get home, and there are no loads close by to get him home. The planner has learned to look for opportunities to move a driver part of the way, and then switch loads with another driver. The strategy is triggered by a particular state (a driver needs to get home, and there is no convenient load to get him home) and the identification of a particular action (swapping loads with another driver).

Computer models use a completely different strategy for solving problems, known in the cognitive science community as a different mode of reasoning (Gelfand et al. 1998). Optimization models require that we formulate a cost function that serves as an index for comparing decisions. The mathematics of optimization algorithms then brings to each decision a very high information content (typically through dual variables), which can be a problem when the data are not reliable. The difficulty is that it is not immediately apparent which data are impacting a decision, so the planner does not know which information has to be verified to be sure that the decision is the right one. We refer to this issue as *diagnosability*, and call models *diagnosable* when we can trace the output of the model to the input. As a rule, optimization models are not diagnosable.

The challenge of calibrating a model is that we are often faced with the problem of trying to find the right set of weights w_i to produce the answer we want and expect. Our answer, however, is derived through *state/action* reasoning, while the model's decision is based on cost minimization. Changing the behavior of the model requires modifying costs, a process that is generally foreign to most planners.

The issue of diagnosability, combined with the challenge of working with artificial costs, means that it can be quite difficult to determine how to get an optimization model to adopt a particular behavior ("when in this state, we like it when the model does that"). However, the challenge is not insurmountable. Management routinely sets goals and then asks people in operations to meet those goals. Dispatchers have to make trade-offs in the process of making individual decisions that imply a weight. Should he move the driver 20 extra miles to get him home? How he answers this question provides an indication of his implicit weight on moving empty.

6.3. Knowing When to Trust the Model

A properly designed and implemented optimization model is going to give sensible results most of the time, even in the presence of the usual stream of data errors. Many decisions are going to be fairly obvious, meaning that human and computer will agree in these instances. The problem is when the "higher reasoning" of the computer produces decisions that disagree with the pattern-based reasoning of the human. Now the human has a dilemma. Is the discrepancy a result of "higher reasoning" or a simple data error? The culture of optimal models has been to focus on solution quality at the expense of diagnosability. As a result, if we do not agree with the recommendation being made by a model, it is very difficult (often impossible under the time constraints of real-world operations) to find out why the recommendation is being made. The outcome is not hard to predict—the dispatcher will typically go with his own intuition.

7. EPILOGUE

The primary conclusion of this project is that the optimization model, which was still running three years after the initial development was completed, struggled from a lack of information. Despite the real-time access to the host computer that held all the information in the database, there was a lot of information not only in the head of the dispatcher but also in the head of the driver (who might elect to refuse an assignment for completely unknown reasons). Also, we would have to fight established patterns of behavior in the planners (another form of information we had not captured), and while some may argue that these patterns encapsulate bad practices, it is certainly not the case that all, or even most, of these practices were in fact "bad." Without question, however, the most frustrating lack of information arose from nothing more than poor communications performance between the host database and the workstation holding the computer; comparing actual decisions to those generated by the model, it was frustrating when we would receive a message assigning a driver to a load that had not even been passed to the model.

Despite these challenges, careful simulations of actual dispatch decisions and comparisons against model recommendations at the same point in time demonstrated that the model would have produced better operating statistics along several key dimensions measured by management. It is a common question to ask: How much better would the company have operated if they had just followed the model? We claim that our analysis of actual decisions is the closest that anyone can come to answering this question, and the analysis suggests that following the model would produce significant benefits. While overcoming these hurdles requires solving some of the standard problems of user acceptance, it also requires a deeper understanding of the reasons why users do not follow the model. These include:

(1) Humans have to respond to information they have that is not available to the computer and that is unlikely to become available as a result of the sheer cost of getting information into the computer. Instead, computers need to be able to respond to actual user decisions very quickly.

(2) Dispatchers make decisions using certain modes of reasoning that are different from those used by a computer. The logic of "how" the computer arrived at an answer differs from how a dispatcher would handle a problem (which

they would all do somewhat differently), making it difficult to resolve logical discrepancies.

(3) The mix of measures, and the differences in how these measures are calculated and weighted, will produce discrepancies. In particular, not all dispatchers put the same weight on all issues.

In a complex, operational setting, no model will ever be perfect, simply because it is too expensive to get all the necessary data into the computer. However, there is a point at which the value of a global perspective will exceed the value of local knowledge. As of this writing, we are aware of one trucking company that has virtually turned their operation over to computerized dispatch (in sharp contrast with another company that has decided in favor of letting drivers dispatch themselves). The evolution of "automated thinking" is probably going to parallel the use of robots in automotive assembly. In the 1980s, this was an overhyped technology with more failures than successes. In the 1990s, robotics is now firmly entrenched in certain applications (e.g., welding) on the assembly line. It seems likely that optimization models will become the next wave of robots, with a similar evolutionary path.

ACKNOWLEDGMENTS

This research was supported in part by the Air Force Office of Scientific Research, Project No. AFOSR-F49620-93-1-0098, and from a contract with Burlington Motor Carriers. For more information on the work of the authors, visit the Castle Lab website at ⟨http://www.castlelab.princeton.edu⟩.

REFERENCES

Bagchi, B. K., B. N. Nag. 1991. Dynamic vehicle scheduling: An expert systems approach. *Internat. J. Physical Distribution and Logist. Management* **21**(2) 10–18.

Bell, W., L. Daberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, P. Prutzman. 1983. Improving the distribution of industrial gases with an online computerized routing and scheduling optimizer. *Interfaces* **13** 4–23.

Brown, G., G. Graves. 1981. "Real-time dispatch of petroleum tank trucks. *Management Sci.* **27** 19–32.

——, C. Ellis, G. Graves, D. Ronen. 1987. Real-time, wide-area dispatch of mobile tank trucks. *Interfaces* **17**(1) 107–120.

Cape, D. J. 1987. MICROMAP: A dynamic dispatch and planning model for truckload motor carriers. Senior thesis, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ.

Card, S. K., T. P. Moran, A. Newell. 1983. *The Psychology of Human Computer Interaction*. Erlbaum Assoc., Hillsdale, NJ.

Gelfand, J., S. Epstein, W. Powell. 1998. Integrating pattern learning in multimodal decision systems. *Proceedings of the AAAI Spring Symposium on Multimodal Learning*. AAAI Press, Menlo Park, CA 96–101.

Klein, G. A., R. Calderwood. 1991. Decision models: Some lessons from the field. *IEEE Trans. Systems, Man, and Cybernetics* **21** 1018–1026.

Powell, W. B. 1991. Optimization models and algorithms: An emerging technology for the motor carrier industry. *IEEE Special Issue on Intelligent Vehicle/Highway Systems*. **40**(1) 68–80.

——. 1996. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Sci.* **30**(3) 195–219.

——, Y. Sheffi, K. Nickerson, S. Atherton. 1988. Maximizing profits for North American Van Lines' truckload division: A new framework for pricing and operations. *Interfaces* **18** 21–41.

——, W. Snow, R. K.-M. Cheung. 2000a. Adaptive labeling algorithms for the dynamic assignment problem. *Transportation Sci.* **34**(1) 67–85.

——, M. T. Towns, A. Marar. 2000b. On the value of globally optimal solutions for dynamic routing and scheduling problems. *Transportation Sci.* **34**(1) 50–66.

Psaraftis, H. N. 1988. Dynamic vehicle routing. B. L. Golden, A. A. Assad, eds. *Vehicle Routing: Methods and Studies*. North Holland, Amsterdam, The Netherlands, 223–248.

——. 1995. Dynamic vehicle routing: Status and prospects. *Ann. Oper. Res.* **61** 143–164.

Regan, A. C., H. S. Mahmassani, P. Jaillet. 1995. Improving efficiency of commercial vehicle operations using real-time information: Potential uses and assignment strategies. *Transportation Res. Record* **1493** 188–198.

——, ——, ——. 1996. Dynamic decision making for carrier fleet operations using real-time information. *Transportation Res. Record* **1537** 91–97.

——, ——, ——. 1998. Evaluation of dynamic fleet management systems: A simulation framework. *Transportation Res. Record* **1645** 176–184.

——, ——, ——. 1999. Dynamic fleet management for trucking operations. World Markets Research Centre, eds. *Global Truck and Commercial Vehicle Technology.* 31–35.

Schrijver, P. R. 1993. Supporting fleet management by mobile communications. Ph.D. dissertation, Delft University, The Netherlands.

Taylor, G. D., T. S. Meinert, R. C. Killian, G. L. Whicker. 1999. Development and analysis of alternative dispatching methods in truckload trucking. *Transportation Res. E: Logist. Transportation Rev.* **35**(3) 191–205.