

# On the Value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Noncompliance

WARREN B. POWELL, MICHAEL T. TOWNS, AND ARUN MARAR

*Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey 08544*

*The most common approach for modeling and solving routing and scheduling problems in a dynamic setting is to solve, as close to optimal as possible, a series of deterministic, myopic models. The argument is most often made that, if the data changes, then we should simply reoptimize. We use the setting of the load matching problem that arises in truckload trucking to compare the value of optimal myopic solutions versus varying degrees of greedy, suboptimal myopic solutions in the presence of three forms of uncertainty: customer demands, travel times, and, of particular interest, user noncompliance. A simulation environment is used to test different dispatching strategies under varying levels of system dynamism. An important issue we consider is that of user noncompliance, which is the effect of optimizing when users do not adopt all of the recommendations of the model. Our results show that (myopic) optimal solutions only slightly outperform greedy solutions under relatively high levels of uncertainty, and that a particular suboptimal solution actually outperforms optimal solutions under a wide range of conditions.*

The challenge of optimizing routing and scheduling problems in a real-time setting has been receiving increased attention (see, for example, PSARAFTIS, 1988). The most common approach to solving these problems is to solve a model using the data as they are known at a certain point in time, and then reoptimize as new data become available. Researchers in logistics typically work to find the best possible solution for a problem instance. In some cases, this may involve solving little more than an assignment problem or network problem (e.g., POWELL, 1996) or a more complex integer programming problem (such as those described in DESROSIERS, SOLOMON, and SOUMIS, 1995). In harder problems, such as classical vehicle routing, optimal solutions are very hard to find, raising the question of whether the additional effort is worth it. In related areas, such as machine scheduling, it is common to solve these problems using greedy heuristics (see GRAHAM, 1966; ALBERS, 1997; and the general book PINEDO, 1995, and the references cited there). This field has received considerable recent attention from the computer science

community in the form of the study of on-line algorithms (SLEATOR and TARJAN, 1985).

Routing and scheduling in a dynamic environment has been studied by a number of authors. One line of research has sought to analyze the properties of greedy heuristics for stochastic, dynamic problems (BERTSIMAS and VAN RYZIN, 1991; BERTSIMAS and RYZIN, 1991; BERTSIMAS and SIMCHI-LEVI, 1996; BERTSIMAS, CHERVI, and PETERSON, 1995). Others have sought to develop explicit stochastic, dynamic models that incorporate both here-and-now information and some forecasts of future activities (see, for example, DROR, LAPORTE, and TRUDEAU, 1989; LAPORTE and LOUVEAUX, 1990; STEWART and GOLDEN, 1983; TRUDEAU and DROR, 1992; POWELL, 1996).

One of the challenges of classical vehicle routing problems is that it is looking to solve a stochastic version of a problem where the deterministic version remains computationally intractable and, for most problems, impossible to solve to optimality. Related research has focussed on problems where the deterministic versions are relatively easy to solve, mak-

ing the stochastic version somewhat easier to analyze. One problem in this class looks at managing fleets of vehicles over time, and the closely related problem of assigning drivers to full-load movements. One of the first deterministic models of a fleet management problem was posed by WHITE (1972), which was followed by a series of papers suggesting and solving different approximations of a multistage, stochastic dynamic program (JORDAN and TURNQUIST, 1983; POWELL, 1986, 1987, 1988; and FRANTZESKAKIS and POWELL, 1990). Recently, CHEUNG and POWELL (1996) showed that, for this problem class, explicit stochastic models can significantly outperform deterministic models in rolling horizon experiments. Despite this research, most models in practice are basically deterministic and myopic. All of these papers were oriented toward the management of large fleets of vehicles.

Recent research (SCHRIJVER, 1993; Powell, 1996) has focussed on the problem of matching individual drivers to full-truckload movements, a problem called the load matching problem in truckload trucking. This problem class is much closer to vehicle routing problems in its basic characteristics, with the important exception that there is no in-vehicle consolidation. In addition, the loads are typically very long in duration, often requiring one or more days to complete. The advantage of this problem structure is that a single instance of the problem is a network assignment problem, considering the matching of drivers to loads. This problem is easy to solve to optimality, and easy to update with real-time information. Schrijver (1993) used a greedy heuristic for matching drivers to loads in a study of the value of real-time communications. The research in Powell (1996) compared a myopic model to an approximation of the stochastic, dynamic problem, and showed that the stochastic, dynamic model outperformed the myopic model in rolling horizon experiments.

In this paper, we consider only myopic models of the load matching problem for truckload trucking. Following common engineering practice, our plan is to solve a given instance of the problem to optimality, and then reoptimize as new information becomes known. This is the approach that is widely used in commercial applications. Our interest is in comparing optimal solutions of sequences of deterministic, myopic models, obtained by solving the network assignment problem, to various approximations including greedy solutions that simply assign drivers to the best load. We study these algorithms under three forms of system stochasticity. First, we consider the random process of shippers calling in loads. Second, we investigate the effects of network

performance by modeling randomness in travel times, which creates uncertainties around the arrival time of a driver at the destination. And third, we introduce and investigate the general issue of user noncompliance. Because no model is perfect, users routinely override the solutions of driver assignment models. As a result, an additional level of randomness arises in the actual use of the recommendations of these models. In practice, several companies report that average usage of these models is typically below 60%, and good performance is considered around 70%.

We believe that the issue of user noncompliance is fundamental to all models, static and dynamic, and that it raises the question of whether so-called optimal algorithms are indeed optimal. For example, if real-world schedules do not use 100% of the solution produced by a crew scheduling model (in a static, planning environment), was the original solution even optimal? In this paper, we show that, in the presence of user noncompliance, optimal solutions will be outperformed, over time, by algorithms that are more local in nature. We feel this has broad implications for the design of algorithms in other settings (vehicle routing, crew scheduling), which seek to find optimal solutions to sequences of deterministic, myopic problems.

To address these issues in a systematic fashion, we have developed a simulation architecture to simulate truckload dispatching operations. This architecture consists of three modules: a network optimization model for assigning drivers to loads, a fleet simulator/dispatcher module, and a demand generation module. The network optimization model is a production system that is in use at a major motor carrier. The modules communicate with each other through files, closely simulating the actual flow of data at a carrier (where the optimization model typically runs on a separate workstation, communicating with the carrier's primary management information system through a database). Our experiments used the actual data from a large motor carrier, scaled down to simulate a smaller fleet (of 400 trucks), a step that accelerated the running of each simulation (which need to run much faster than real time).

The primary contributions of this paper are as follows.

- We show that different forms of system stochasticity have the effect of reducing the value of optimality when solving sequences of subproblems in dynamic settings, and we show that a solution that is suboptimal for a particular prob-

lem instance actually provides higher solution quality over an extended simulation.

- We introduce and study for the first time the issue of user noncompliance, and show that this form of uncertainty can, in particular, reduce the value of optimal solutions. Noncompliance is an easy to measure barometer of the quality of an optimization model as perceived by the user. Even relatively low levels of user noncompliance (say, 30% modification of model recommendations) indicate that partially greedy solutions can outperform optimal solutions. At the same time, similar levels of user noncompliance significantly reduced the gap between an optimal solution and a greedy solution.
- We experimentally quantify the effects of different forms of system uncertainty, providing a measure of the value of reducing these forms of uncertainty on actual system performance.

The paper is organized as follows. Section 1 describes different perspectives of optimality in the context of dynamic routing and scheduling. Section 2 presents an overview of dispatching operations in the truckload motor carrier industry, which we use as our problem setting. Section 3 presents the optimization model that is used to solve the problem. Next, Section 4 discusses some of the issues that arise in global versus local optimization. These issues are central to any optimization problem where the results will be implemented over time, during which the status of the system may change. Then, Section 5 gives an overview of our simulation architecture, which takes advantage of a production load matching system that is in use at a major motor carrier. This is followed in Section 6 by a description of the specific modules. Section 7 describes the experimental design, and the results are reported in Section 8. Section 9 summarizes the major conclusions of the paper.

## 1. PERSPECTIVES ON OPTIMALITY

TO PLACE THIS PAPER in the proper context within the research community, it is important to briefly discuss the meaning of the word “optimal.” It is common engineering practice to formulate a mathematical model, typically with numerous simplifications to ensure tractability, and then to seek an optimal solution to this model. We all accept that the word optimal is used in the narrow sense of solving a specific mathematical representation, without making any claims regarding the impact on the real problem, because these impacts fall outside of what can be measured experimentally using scientifically

rigorous standards. In the context of dynamic problems, it is necessary to solve a sequence of problems over time as new information arises, thereby simulating the real process.

In this setting, we must explicitly acknowledge three models: the sequences of subproblems that are solved over time (each of which is a model with its own objective function), the larger simulation (with its own cost stream), and the even larger problem in the real world, which must reflect data errors and other biases.

Let  $\omega = \{\omega_0, \omega_1, \dots, \omega_t, \dots, \omega_T\}$  be a sequence of random outcomes over a horizon  $\mathcal{T} = \{0, 1, \dots, T\}$ , where  $\omega \in \Omega$ . We would then define a probability space  $(\Omega, \mathcal{F}, \mathcal{P})$  where  $\mathcal{F}$  is the  $\sigma$ -algebra on  $\Omega$ , and  $\mathcal{P}$  is a probability measure defined over  $\mathcal{F}$ . Let  $\mathcal{F}_t$  represent the sequence of sub- $\sigma$ -algebras containing the information known up to time  $t$ , where  $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$ . Let  $f_t^\pi(x_t | \mathcal{F}_t)$  be the objective function at time  $t$  given an information set  $\mathcal{F}_t$  at time  $t$ . The superscript  $\pi$  captures both the structure of the function  $f_t$  and any parameters used to create the function (including, for example, planning horizons, discount factors, and any cost factors introduced so that solving  $f_t$  at time  $t$  will produce a solution with good long-run behaviors). We can think of  $f_t^\pi$ ,  $\pi \in \Pi$  as the set of all possible functions (or policies, if you wish) that we could use at time  $t$ . Now let  $x_t^\pi(\mathcal{F}_t) = \arg \min_{x_t \in X} f_t^\pi(x_t | \mathcal{F}_t)$  be a sequence of  $\mathcal{F}_t$ -adapted decisions (meaning that the function  $x^\pi$  can only use information in  $\mathcal{F}_t$ ). (Note: The concept of  $\mathcal{F}_t$ -adapted decisions is similar, but not identical, to the concept of nonanticipativity used in the stochastic programming community, which is expressed as a constraint requiring that one decision be chosen in time period 0 for all possible outcomes in the future. Instead of saying that a decision is  $\mathcal{F}_t$ -adapted, some authors prefer to say that the decision is  $\mathcal{F}_t$ -measurable.)

The decision function  $x_t^\pi(\mathcal{F}_t)$  returns a set of decisions that are to be implemented at time  $t$  given the events  $\mathcal{F}_t$ . We write  $x^\pi(\omega)$  to represent the decision made for a particular realization  $\omega$ , with the implicit understanding that the function  $x^\pi$  is  $\mathcal{F}_t$ -measurable. In determining this vector, we need to consider both here-and-now costs at time  $t$ , and possibly approximate estimates of costs that may be incurred in the future, but which are subject to solving  $x_{t'}^\pi$  for  $t' > t$ . For example, we may decide that a vehicle should go from customer  $i_1$  to customer  $i_2$  right now; this is because we have formed a tour that consists of  $(i_1, i_2, i_3, i_4)$ . The objective function  $f_t^\pi$  would be formulated to capture these future decisions, and may also include approximations of what would happen even farther in the future (as was done in Powell, 1996). However, only the vector  $x_t^\pi$  is actually

implemented. Let  $c_t$  represent the cost of these decisions. Then, the costs incurred at time  $t$  would be  $c_t x_t^\pi$ , and the total costs over the entire simulation would be  $F(x, \omega) = \sum_{t \in \mathcal{T}} c_t x_t^\pi(\omega)$ , where  $x = (x_0^\pi, x_1^\pi, \dots, x_T^\pi)$ .

We are now ready to talk about different types of optimality. Our goal as modelers is to solve

$$\max_{x \in X} F(x) = \mathbb{E} \left\{ \sum_{t \in \mathcal{T}} c_t x_t^\pi(\omega) \right\},$$

where  $X$  is the space of all feasible  $\mathcal{F}$ -adapted controls  $x_t^\pi$ ,  $\mathcal{T}$  is the planning horizon, and the expectation operator is over all outcomes in  $\Omega$ . In our hierarchy of optimization problems, this is our larger problem. In practice, we solve  $F(x)$  using a series of subproblems  $f_t^\pi(x_t)$  in an effort to find a solution that maximizes  $F(x)$ .  $F(x)$ , in turn, is nothing more than a proxy for the even larger problem in the real world, which must also capture issues that we are unable to put into our model.

In the arena of routing and scheduling, it is very common to use myopic solutions that are then updated as new information arrives. Although this is the simplest possible option for most classes of routing and scheduling problems, it remains a combinatorically difficult problem. Thus, even finding the optimal solution to this relatively simple problem is a nontrivial task. Because  $f_t^\pi(x_t)$  represents a single instance of a classical optimization problem, it is understandable that a considerable amount of effort will be devoted toward optimizing this problem, realizing that it is simply a single snapshot of a larger problem. Of course, even the larger problem, represented by  $F(x)$ , is little but an approximation of an even larger problem that encompasses information that is not captured within the model. Because dynamic routing models must run in production, this even larger problem, which must reflect the insights and biases of real users, must begin to be addressed within the context of our research.

In this paper, we model the effect of the even larger problem through a process we call user compliance. In this process, we present a recommendation to the user, at which point the recommendation is accepted or rejected. We make the assumption that the cause for rejecting a recommendation reflects information outside the domain of the model, and, hence, we model it as an exogenous random variable. Specifically, for a particular decision, say  $x_{ij}$ , we introduce a random upper bound  $u_{ij}(\omega) = 1$  with probability  $p^d$ , and 0 with probability  $(1 - p^d)$ . Only when we choose  $x_{ij} = 1$  do we then learn the value of  $u_{ij}(\omega)$  where  $u_{ij}(\omega) = 1$  represents the user accepting the recommendation, whereas  $u_{ij}(\omega) = 0$

represents a rejection. In the event of a rejection, it would be possibly to exclude a solution and reoptimize. In our work, we followed industry practice in this setting and used the dual variables from the optimal solution to suggest alternatives.

Because the physical process representing a user accepting a decision is extremely complex, it is customary to model a process such as this as a random variable. This modeling approach is comparable to adding a noise term at the end of a function to capture all the nonmeasurable effects. We feel that this representation of the user compliance issue reasonably captures the effects of the model adoption process, and offers the benefit of being simple and experimentally analyzable. Given that we cannot observe the real process of why users reject model rejections (and we have tried very hard to do just this), the next best alternative, and far superior than ignoring the issue entirely, is to model the acceptance process as a random variable.

In the presence of user noncompliance, as well as other forms of uncertainty, it is reasonable to ask the question: what is the value of an optimal solution of the function  $f_t^\pi(x_t)$ ? Our work focuses on a problem where finding the optimal solution, given conventional modeling practice, is actually quite easy, allowing us to compare solutions that solve  $f_t^\pi(x_t)$  optimally to those that solve it only approximately. We consider only myopic models, in part because these are the models that are most widely used in practice, and more importantly, because we do not want to get into the tremendously rich and complex arena of developing approximations of future activities. Given the emerging research into the application of vehicle routing models in dynamic settings, which focuses primarily on the heuristic solution of myopic models, we feel that our choice of research methodology is appropriate and produces interesting results. Most significantly, we feel that the issue should be considered when testing different search algorithms in the context of harder problems such as dynamic vehicle routing. For example, in the presence of user noncompliance, time-consuming search heuristics may prove to add little or no value in a dynamic environment.

We do not claim that our simple model of user compliance precisely captures the complex physics of human thought. In support of our approach, we draw on the long history of mathematical models that use random variables to capture highly complex processes. We do claim, however, that the estimates that we obtain of the value of optimal solutions in a dynamic environment are more accurate using our model of user compliance than they would be if we ignored the issue altogether. We further claim that

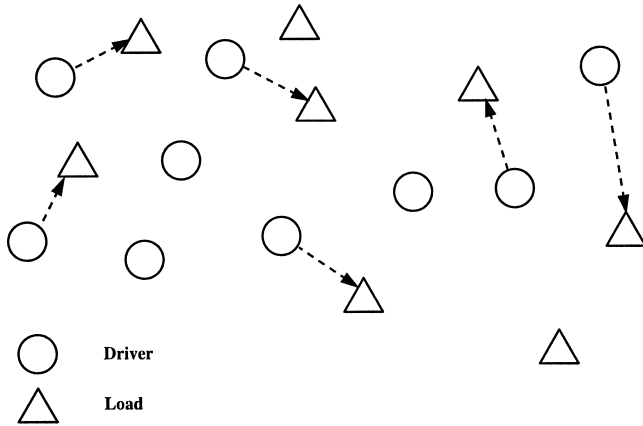


Fig. 1. A simplified spatial assignment network.

an algorithm that provides a better solution to our model of user compliance is likely to be a more robust algorithm that will provide better results in practice.

## 2. THE DYNAMIC ASSIGNMENT PROBLEM IN TRUCKLOAD OPERATIONS

WE FIRST PRESENT an introduction to the dynamic assignment problem and its relationship to truckload motor fleet management (see Schrijver, 1993 and Powell, 1996 for a much more complete discussion). The dynamic assignment problem involves the allocation of resources to perform tasks over a discrete-time dynamic network. Only some fraction of the relevant tasks in the network is known initially, with additional tasks becoming known in real time as time progresses. This means that the optimal solution for the problem is constantly changing as new information becomes known. The result may be that the viability of a previously optimal solution is reduced significantly in light of new information.

In truckload applications, drivers and loads are spread almost continuously over space. At a point in time, the challenge is to match drivers to loads, as illustrated in Figure 1. The problem is complicated by the fact that there is much more to the problem than simply finding the closest driver to a load. When finding a driver, a planner has to balance deadhead (empty) miles, the availability of the driver (some of the drivers may only be available late in the day), the pickup and delivery windows of the load, where the driver lives and whether the load will get him home on time, the skill and experience of the driver (some shippers require top-rated drivers), whether the driver is a single driver (which can only drive 10 hours at a stretch) or a team (which can drive almost continuously), and other characteristics. In addition, the planner needs to consider the

possibility of loads that will be called in later in the day.

The operations of a truckload carrier are relatively simple in concept. A shipper calls the dispatching office to request the pickup of a load at a specific location within a specific time window. This time window may begin on the day of the call-in or it may begin some time in the future. The load must be delivered to a specified location within a specific time window. Other relevant information, such as special driver and trailer requirements, is also supplied by the shipper.

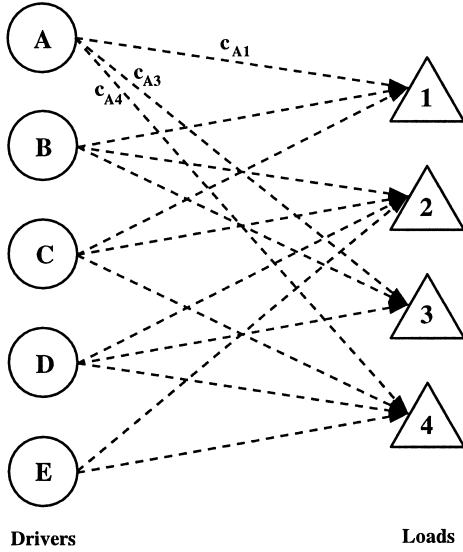
The dispatching situation for a major truckload carrier generally involves multiple human dispatchers each working within a specified geographic region. In the absence of decision support systems, assignment decisions are made more or less independently of the actions of other dispatchers. Such a pattern of multiple dispatchers acting this way in relative isolation from one another tends to produce a geographic patchwork of locally greedy optimizations. These local optimizations mostly neglect the effects downstream in space and time of assignments. The result is a solution that is suboptimal from a mathematical standpoint as compared to a solution derived from an optimization decision support system.

## 3. THE LOAD MATCHING OPTIMIZATION MODEL

THE LOAD MATCHING problem assumes that at a given point in time  $t$ , we have a set of loads  $\mathcal{L}_t$  and a set of drivers  $\mathcal{D}_t$ , and we need to match drivers to loads. Some of the loads may not be available for pickup until some time in the future (possibly a day or more) and some of the drivers are currently pulling loads, and also will not be available until some point in the future (the driver estimated time of arrival (ETA)).

For each potential assignment of a driver  $d \in \mathcal{D}_t$  to a load  $l \in \mathcal{L}_t$  we can assign a cost  $c_{dl}^a$  (this cost will change with time). In addition, we must consider the cost of refusing to service a load, and the cost of holding a driver idle. The problem here is to decide which drivers to assign to which loads to minimize the total costs (maximize total profits) throughout the network. An example of the driver assignment problem (using only assignment costs) is shown in Figure 2.

Loads are characterized by a) an origin and destination, b) a pickup time window and a delivery time window, and c) a vector of attributes, which might include such items as type of driver required and load priority. Drivers are characterized by a) a location at time of availability, b) a time of availabil-



**Fig. 2.** A simple driver assignment problem. The  $c_{dl}$  on selected arcs represent the arc cost for the assignment of driver  $d$  to load  $l$  at time  $t$ .

ity, which may be in the future (if the driver is currently assigned to a load) or the past (if the driver is waiting for assignment), and  $c$  a vector of attributes including special information like time spent on duty, experience level, and home domicile.

We can formulate this assignment problem as a simple static assignment model as follows.

$\mathcal{L}_t$  = the set of all loads known to the system at time  $t$

$\mathcal{D}_t$  = the set of all drivers in the network at time  $t$

$c_{dl}^a$  = the cost of assigning driver  $d$  to load  $l$  at time  $t$

$c_{lt}^r$  = the cost of not assigning any drivers to load  $l$  at time  $t$  (the refusal cost)

$c_{dt}^h$  = the cost of not assigning driver  $d$  to any load at time  $t$  (the holding cost).

Our decision variables are

$$x_{dl}^a = \begin{cases} 1 & \text{if driver } d \text{ is assigned to cover} \\ & \text{load } l \text{ at time } t \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{lt} = \begin{cases} 1 & \text{if load } l \text{ is refused at time } t \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{dt}^h = \begin{cases} 1 & \text{if driver } d \text{ is not assigned to any load} \\ 0 & \text{otherwise.} \end{cases}$$

The objective function is then given by, for a given point in time  $t$ ,

$$\min_{x^a, x^h, y} f_t(x_t) = \left( \sum_{d \in \mathcal{D}_t} \sum_{l \in \mathcal{L}_t} c_{dl}^a x_{dl}^a + \sum_{d \in \mathcal{D}_t} c_{dt}^h x_{dt}^h + \sum_{l \in \mathcal{L}_t} c_{lt}^r y_{lt} \right) \quad (1)$$

subject to

$$\sum_{d \in \mathcal{D}_t} x_{dl}^a + y_{lt} = 1 \quad l \in \mathcal{L}_t \quad (2)$$

$$\sum_{l \in \mathcal{L}_t} x_{dl}^a + x_{dt}^h = 1 \quad d \in \mathcal{D}_t \quad (3)$$

$$x_{dl}^a, y_{lt}, x_{dt}^h \geq 0 \quad l \in \mathcal{L}_t, \quad d \in \mathcal{D}_t.$$

We let  $v_l$  be the dual variable for Eq. 2 and  $v_d$  be the dual variable for Eq. 3. Note that  $f_t(x_t)$  represents the costs that would be incurred at time  $t$  if  $x_t$  were chosen. This function is purely greedy and makes no effort to incorporate factors that might improve the selection of  $x_t$  to find solutions that would produce higher values for the multistage problem. In Section 1, the notation  $f_t^\pi(x_t)$  refers to a function that incorporates not only the immediate effect of a solution (represented by  $f_t(x_t)$  above) but any other adjustments that might be included to produce better long-run solutions. In the remainder of this paper, we use only  $f_t(x_t)$  because we are considering only solutions of myopic models.

The optimal solution of Eq. 1 can be found by solving a simple assignment problem as depicted in Figure 2. In more complex problems, such as those that involve routing a driver through a sequence of points (for example, if the loads are short, or if there is in-vehicle consolidation), we may have to resort to heuristic solutions. In a dynamic setting, it is reasonable to consider the use of suboptimal or even greedy solutions. For this reason, we let  $\Pi$  be the class of algorithmic strategies (or policies), and let  $x_t^\pi$ ,  $\pi \in \Pi$  be the solution obtained using policy  $\pi$ .

The cost function  $c_{dl}^a$  for a truckload motor carrier has to capture a number of performance measures. A successful carrier must minimize empty miles, maximize on-time service for loads, maintain a high level of driver satisfaction, and maximize equipment productivity. In the truckload arena, where drivers often spend several weeks away from home, driver satisfaction is measured by our ability to get drivers back to their home by a certain time. Other issues include whether a driver has the right training to handle the needs of a particular load; whether his truck is too heavy to handle a particular load; and whether he is considered sufficiently reliable to meet the needs of an account.

We denote all these factors using a multidimensional cost function that is then collapsed into a single utility function. Suppressing the time param-

eter  $t$ , we may define

$\mathcal{M}$	= set of cost factors to be considered in the assignment of a driver (such as empty miles, on-time service, and so on)
$\mathbf{a}_d$	= vector of attributes of driver $d$
$\mathbf{b}_l$	= vector of attributes of load $l$
$r_l$	= revenue, or reward from picking up a load
$\rho$	= vector of parameters that are used to translate a performance measure (such as arriving late to pick up a load) into a cost.
$c_{mt}^a(\mathbf{a}_d, \mathbf{b}_l, \rho)$	= cost of factor $m \in \mathcal{M}$ of assigning a driver with attribute vector $\mathbf{a}_d$ to a load with attributes $\mathbf{b}_l$ at time $t$ , given utility parameter vector $\rho$

The net total contribution of assigning driver  $d$  to load  $l$  at time  $t$  can now be expressed as

$$c_{dl}^a(\rho) = r_l - \sum_{m \in \mathcal{M}} c_{mt}^a(\mathbf{a}_d, \mathbf{b}_l, \rho). \quad (4)$$

In this paper, we are using the cost function that is in production at a major motor carrier. The parameters have been carefully chosen to reflect what the carrier believes is an accurate measure of the factors that impact the performance of the behavior. Thus, when we translate a late pickup into a \$50 cost, we treat this as a real cost, comparable to incurring \$50 in transportation costs.

#### 4. GLOBAL VERSUS LOCAL OPTIMIZATION OF THE SUBPROBLEM

THE EASE WITH WHICH the driver assignment model can be solved has spawned a cottage industry of software vendors who supply models of this sort to the truckload industry. A major selling point was the ability to perform global optimization in an industry where the driver assignment problem was solved manually, using methods that tended to produce locally good solutions. Researchers are actively developing similar globally optimal solutions for other problem classes in vehicle routing and scheduling (see Desrosiers et al., 1995 and FISHER, 1995 for recent surveys of the field). Of course, the value of a globally optimal solution depends, as always, on the quality of the data. These references all refer to global optimization in the context of providing a mathematically optimal solution to an individual subproblem  $f_t(x_t)$ . For the remainder of this paper, all references to global optimization refer to our ability to find the optimal solution to  $f_t(x_t)$  as op-

posed to a heuristic or suboptimal solution (as might be used in harder problems such as vehicle routing).

Real-time problems offer a special challenge to the optimization community because it is difficult to get quality data in a timely way. For example, humans possess a certain amount of “head knowledge,” which is information that they have acquired by telephone, conversations, visual inspection, and experience that is not in the computer. In addition, there is information that no one knows. The most significant form of uncertain information is the demands of the customers, but we would also include weather delays, breakdowns, and failure of drivers to perform a given task.

We believe these factors are fundamental to any application. The result is that the recommendations made by a model are not always implemented. We refer to this effect as user noncompliance, which measures the degree to which users actually implement the recommendations of a model. User noncompliance often reflects information the user has that the system does not. The presence of user noncompliance implies that globally optimal solutions are, in fact, not optimal at all. Humans reflect this property intuitively by making decisions that are good in a local sense (both spatial and temporal) but which may not, in theory, properly take into account their impact on other parts of the system (in space or time). For this reason, software developers have argued that global optimization models can outperform humans, overcoming the limitations humans have in dealing with large problems.

We can quickly illustrate the impact of uncertain data on solution quality using a simple illustration shown in Figure 3. Here, we show driver A covering load 1, driver B covering load 2, and driver C covering load 3. Driver C would prefer to cover load 1, but if this were done, driver A would not have very attractive options. A few minutes later, a new load is called in that is better for driver A, allowing driver C to switch to load 1 after all. At this point, we learn that driver B would rather cover load 3 instead of load 2.

Given the same example, it would not be uncommon for a human to assign driver C to load 1 and driver B to load 3, leaving both driver A and load 2 unassigned. Because we assumed that driver A did not really have any attractive assignments, the planner might keep driver A unassigned while hoping for something better. This would be an instance of using a greedy solution now to achieve an overall better solution later.

In practice, humans making decisions on an operational basis spend relatively little time trying to find the best solution for a given situation. Instead,

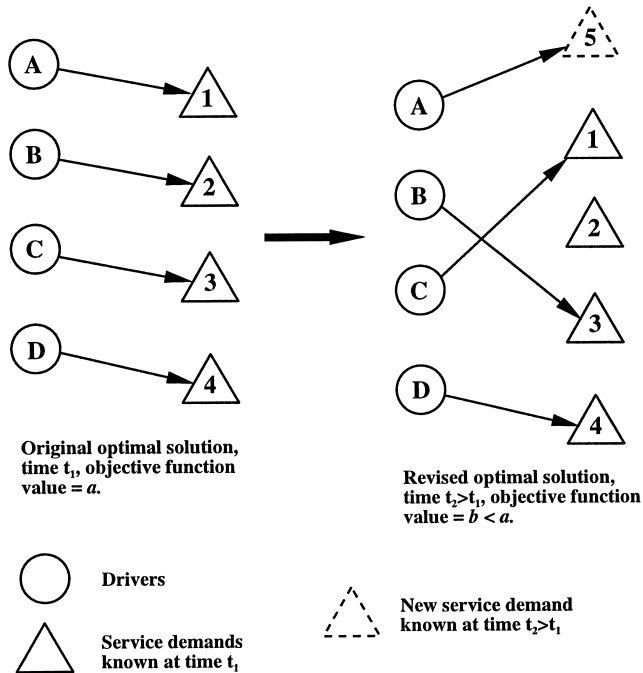


Fig. 3. Revised optimal assignment in light of new service demand information.

they spend most of their time simply verifying that the data they are looking at is, in fact, correct. Given the inherent uncertainty in the data (which may have nothing to do with forecasts of future customer activities), it is perhaps not surprising that they limit their searches for good solutions to small, verifiable subsets of data. Below, we show that such a strategy can actually outperform a globally optimal solution.

## 5. SIMULATOR ARCHITECTURE AND DEVELOPMENT

WE HAVE DEVELOPED a specialized simulator architecture to test hypotheses regarding the behavior of different dispatching strategies in a dynamic environment. A primary goal of our architecture was to take advantage of a production load-matching system that has been developed and implemented for a large truckload motor carrier. For this reason, we adopted a modular architecture consisting of three components: a network optimization module, a fleet simulator/dispatcher module, and a demand-generator module. The interaction between these three modules is represented graphically in Figure 4. Each of the three modules runs independently, coordinated through the use of a common clock object (just as a real system would operate). Of course, the simulation runs at a rate faster than real time.

The simulator architecture allows a wide variety of dispatching situations to be simulated. When the

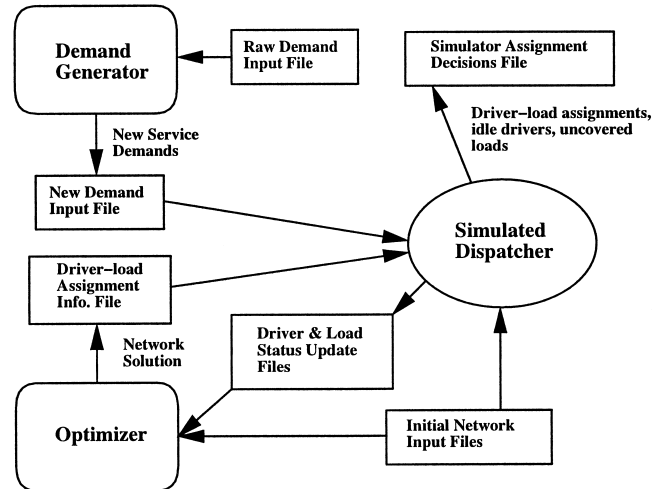


Fig. 4. The multimodular fleet simulation system. The diagram shows the main modules and communication pathways between them.

dispatcher determines that a driver or load needs to be matched, it sends a query to the optimizer asking for information about possible assignments. The optimizer, in simulated real time, sends over a small file containing a number of possible loads that a driver may be assigned to, along with the full set of costs and duals. Depending on parameter settings, the dispatcher may implement exactly what the optimizer recommends, or we may allow it to make other choices.

As the simulation progresses, two types of events can occur: drivers become available for assignment to available loads, and new loads come into the network for assignment. A schematic of the driver assignment-availability process is shown in Figure 5. The letter labels in the diagram refer to the various modules used at each step in this process. As a driver becomes available (in simulation time) for assignment, the simulator module (A) requests from the optimization module (B) driver-load arc-costing information for all loads with which the driver can be feasibly paired. This costing information is of the type required for the costing functions described in Section 3 and includes travel distances, availability times, and load characteristics. In addition, the most recent set of dual variables from the network algorithm is supplied from which the arc reduced costs may be calculated. The optimizer also supplies the flow on each arc, because more than one arc may have zero reduced cost (in such an event, the simulator will always choose the arc to which the optimizer assigned flow). From this information, a dispatching function inside the simulator module (C) makes a driver-load assignment decision (which may include no assignment at all), and updates all



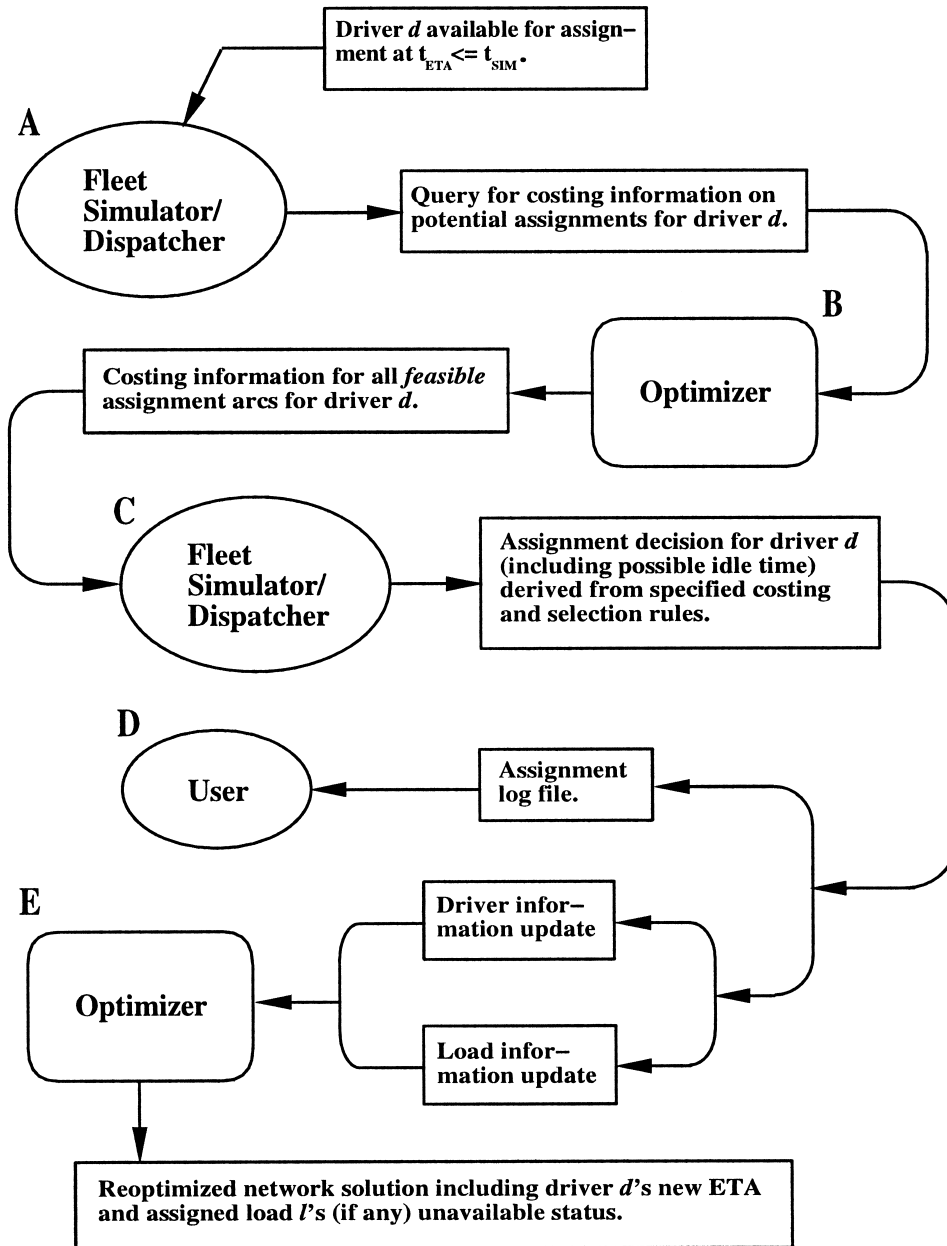


Fig. 5. Sequence of multimodular simulator actions occurring when a driver becomes available for assignment.

relevant driver, equipment, and load information, such as future driver time of availability, and load availability status. The assignment selection, information regarding any currently uncovered (rejected) loads, and any other desired information is then written to a log file for later analysis by the user (D). Driver and load status update information are written to flat files for input by the optimizer module (E), which then supplies a reoptimization of the network.

This driver assignment process is completely general in that it can be used with any optimization algorithm, or even with no algorithm at all, so long

as the appropriate cost and assignment information is given to the simulator module. The assignment decision-making logic is also quite flexible in that the user can specify exactly how and what costing information is to be used. This, for example, allows the user to specify that the simulated dispatcher exactly implement the solution of the optimization algorithm being used and then compare the resulting solution to that obtained by using some arbitrarily defined costing function. Such a function might be used to represent a human dispatcher's internal utility function.

## 6. SIMULATOR MODULES AND PROCEDURES

IN THIS SECTION, we describe some relevant details specific to the three modules that comprise the fleet simulation architecture: the optimizer, the simulator module (or interchangeably, the fleet simulator/dispatcher module), and the demand generator. We also discuss the use of the dispatcher selection probability parameter,  $p^d$ , and the implementation of random travel times in the simulation architecture.

### 6.1 Optimizer

The driver assignment optimizer consists of a network generator and a network simplex code for solving the matching problem. The code is a production dispatch system that is in use at several carriers. The cost functions that determine the penalties for violated delivery windows or for not returning a driver to his home on time are the same costs that are used in production. For this reason, we have treated these costs as real costs, because they have been chosen to accurately capture the tradeoff between meeting these goals and running additional empty miles (which is a hard, well-defined cost).

### 6.2 Demand Generator

The demand generator uses an actual historical demand file, from which it draws a random sample. As the simulation progresses, each load is read in, and a random number is drawn to determine whether it is to be included in this run. If it is, then the generator waits until it is the right time, comparing the actual call-in time of the load to the simulated wall-clock time. When it is the right time, the generator sends the load over to the dispatch simulator.

The use of a real dataset means that we have an accurate picture of actual call-in processes, and real data for the attributes of the loads, including pickup and delivery windows. By using only a sample of these loads, we were able to run repeated simulations using different random number seeds, allowing us to obtain repeated observations of a particular optimization process. In our work, we used a 30% sample of the load, reflecting the fact that our fleet size was approximately 30% of the size of the actual fleet used by the carrier. When we run repeated simulations with the same set of parameters, then we are simply choosing different samples of loads.

### 6.3 Fleet Simulator/Dispatcher

The simulator module performs two main functions. The first is the actual simulation of a truck-load fleet, and the second is the simulation of a human dispatching operation. The main steps in

these functions are presented in Figure 6. The following notation is used in this figure.

- $t_{st}$  = the simulated start time (in seconds) of the simulation run
- $t_{end}$  = the simulated end time (in seconds) of the simulation run
- $t_{cur}$  = the simulated current time (in seconds) of the simulation run
- $d$  = a given driver
- $\mathcal{D}_t$  = the set of all drivers in the network at time  $t$
- $\mathcal{D}_t^a$  = the set of all drivers available for assignment at time  $t$ , ( $\mathcal{D}_t^a \subset \mathcal{D}_t$ ).

The costing of potential assignments described in this figure is done using a modified reduced cost formula. This formula uses a dual variable discounting factor to vary the degree of dual variable utilization in assignment decisions. The formula used is as follows:

$$\bar{c}_{dl}(t, \alpha) = c_{dl}(t) + \alpha(v_{lt} - v_{dt}), \quad (5)$$

where

- $\bar{c}_{dl}(t, \alpha)$  = the adjusted reduced cost for assigning driver  $d$  to load  $l$  at time  $t$
- $c_{dl}(t)$  = the arc cost for assigning driver  $d$  to load  $l$  at time  $t$
- $v_{dt}$  = the dual variable for the node for driver  $d$  at time  $t$
- $v_{lt}$  = the dual variable for the node for load  $l$  at time  $t$
- $\alpha$  = the dual variable discount factor.

Potential assignments for a given driver are ranked for selection purposes from lowest to highest cost using  $\bar{c}_{dl}(t, \alpha)$ . For full global optimization, we set  $\alpha = 1.0$ , which gives us the equivalent network simplex reduced cost. For full greedy optimization, we set  $\alpha = 0.0$ , which gives a ranking according to arc costs. Values of  $\alpha$  in the interval  $(0, 1)$  are used to represent intermediate strategies.

### 6.4 The Dispatch Function

As discussed above, the presence of user noncompliance with the optimization system's recommendations is a major factor in the implementation of optimization decision-support systems. To simulate this process, we define

$$u_{dl}(w) = \begin{cases} 1 & \text{if the user accepts the recommendation} \\ & x_{dl} = 1 \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

**Initialization**

- Initialize internal clock object, status of each driver and initial set of loads.

Set  $t_{cur} = t_{st}$

WHILE  $t_{cur} < t_{end}$ {

  Get\_New\_Loads

- Check for new load information flat file from demand generator, and input and process any new load information.

  FOR  $\forall d \in \mathcal{D}_t^a$  DO {

    Get\_Current\_Available\_Assignments

- Get the cost information for all feasible load assignments for driver  $d$  from the optimizer via flat file swapping.

    Make\_Assignment\_Decision

- Calculate the assignment costs for all feasible driver-load pairings for the given driver. Costing possibilities include:
  - Arc cost calculated from optimizer's cost function or some other specified cost function
  - Reduced cost calculated from optimizer's cost function
  - Adjusted reduced cost (modify dual variables with some factor)
- Rank assignments for the driver by cost.
- WHILE (no assignment made and number of considerations less than  $n$ , a user-defined parameter){
  - Take the best remaining cost-wise pairing with user-defined probability  $p^d$ .
  - If no assignment made (with probability  $(1 - p^d)$ ), remove the previously considered pairing from consideration. }
- If no selection has been made, idle the driver for some specified time period.

  } /\* end of FOR loop \*/

Fleet\_Management

- Updated status of each load is sent to the optimizer.
- Updated status of each driver is sent to the optimizer.
- Assignment information is written to a log file for use by the user.
- Update  $t_{cur}$

} /\* end of main WHILE loop \*/

**Fig. 6.** Main fleet simulator/dispatcher module procedure.

$p^d$  = the probability that a dispatcher will  
 accept a potential driver to load assignment  
 =  $P[u_{dl}(w) = 1]$ .

$$x_{dl}^{\pi}(\alpha, w) = \begin{cases} 1 & \text{if } l = \underset{v \in \mathcal{L}_t}{\operatorname{argmin}} \{\bar{c}_{dl}(\alpha) | u_{dl}(w) = 1\} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

For a given realization  $w$ , we can determine which load a driver will be assigned to, allowing us to define the dispatch function for a driver  $d \in \mathcal{D}_t$  by

Equation 6, then, defines a class of dispatch functions parameterized by  $\alpha$ . If  $\alpha = 1$ , then we obtain the optimal solution of  $f_t^*$ ; if  $\alpha = 0$ , then we obtain a

greedy solution.  $0 < \alpha < 1$  produces a range of intermediate solutions. Of course, many other types of decision functions are also possible. For example, we could use a stable marriage algorithm (see, for example, GUSFIELD and IRVING, 1989) or a local search heuristic. Our function provides an elegant knob that takes us smoothly from greedy local solutions to optimal solutions.

For example, the first assignment will be selected by the simulated dispatcher function with probability  $p^d$  and rejected with probability  $(1 - p^d)$ . If it is rejected, the second-best assignment will then be selected with probability  $p^d$  and rejected with probability  $(1 - p^d)$ . This process continues until either an assignment selection is made, or the number of possible assignments rejected equals a specified user parameter  $n$ . Should  $n$  potential loads be rejected (with probability  $(1 - p^d)^n$ ), the given driver is assigned to remain idle for a specified time period,  $t_{\text{idle}}$ . For our experiments, we set  $n = 5$  and  $t_{\text{idle}} = 60$  minutes of simulation time.

A problem that exists when using this logic is the potential for the selection of excessively expensive assignment arcs that would never actually be selected by a real human dispatcher. For example, when using a value of  $p^d = 0.4$ , the chance of selecting the fourth-best cost-wise assignment is about 9%, and the fifth-best is about 5%. Because, in reality, there are often not five realistic assignment arcs for a given driver, these low-ranked arcs may have very large arc costs and should be considered infeasible from a common-sense standpoint. The inclusion of even a small number of such arcs in the simulated dispatcher's solution can skew objective function values in an unrealistic fashion. To prevent selection of such arcs, a cost cap of \$500 was placed on arcs considered feasible for assignment.

## 6.5 Objective Function

We may now define the total net contribution (or profit) incurred from running a simulation as

$$F(\alpha, p^d, \rho) = E^{p^d} \left\{ \sum_{t=t_{\text{st}}}^{t_{\text{end}}} \sum_{d \in \mathcal{D}_t^o} \sum_{l \in \mathcal{L}_t} -c_{dl}(\rho) x_{dl}^{\pi}(\alpha, w) \right\}, \quad (7)$$

where we choose now to maximize profits over a simulation (instead of minimizing costs at a given point in time). The expectation is over all random events, including the likelihood of user acceptance, given by the parameter  $p^d$  (hence the explicit representation of  $p^d$  in the expectation). We approximate the expectation by running several simulations using different random number seeds. Each run of the

simulation is a function of  $\alpha$  and  $p^d$ . In addition, our simulations considered variations of other problem characteristics, notably the presence of randomness in travel times and the dynamics of the process by which customer demands become known to the system.

## 6.6 Random Travel Times

In transportation applications, one of the most important sources of noise arises in travel times. In the load matching problem, travel times are important because they impact our ability to estimate when a driver will become available in the future, and whether this driver could be assigned to a load with a pickup window. To analyze the impact of errors in travel times, we first define

- $d_{\text{od}}$  = the distance in miles from load origin to destination
- $t_{\text{eta}}$  = the expected travel time in hours  
=  $60d_{\text{od}}/r$
- $r$  = the average rate of speed (typically 40 mph).

We calculate a realization of an actual travel using

$$t_{\text{actual}} = U(t_{\text{min}}, t_{\text{max}})$$

= the real travel time in hours,

where  $U(a, b)$  is a random variable uniformly distributed over the interval  $[a, b]$ . The bounds of our distribution are determined using

$$t_{\text{min}} = \text{the minimum possible travel time,}$$

$$= t_{\text{eta}} - \beta w_s \sqrt{t_{\text{eta}}}$$

$$t_{\text{max}} = \text{the maximum possible travel time,}$$

$$= t_{\text{eta}} + \beta w_e \sqrt{t_{\text{eta}}},$$

where

- $w_s$  = the number of standard errors ( $\sqrt{t_{\text{eta}}}$ ) prior to the expected time to include in the random variable window,
- $w_e$  = the number of standard errors ( $\sqrt{t_{\text{eta}}}$ ) after the expected time to include in the random variable window,
- $\beta$  = noise intensity parameter ranging from 0.0 to 1.0. If  $\beta = 0$ , then there is no noise and we have deterministic travel times.

During the simulated dispatcher's costing of driver-load pairs, real travel times for transit between load origin and destination are obtained by the above formulation and stored in the simulator module's data structures. When the simulator module selects a pairing for assignment, it sends an updated ETA

for the driver to the optimizer using  $t_{\text{eta}}$ . Thus, the optimizer is planning future assignments using  $t_{\text{eta}}$ , whereas the simulator maintains the actual arrival time  $t_{\text{actual}}$ .

In the event that  $t_{\text{actual}} < t_{\text{eta}}$ , the driver will arrive at the load destination and become available for a new assignment earlier than expected. At this time, the simulated dispatcher queries the optimizer module to obtain costing information for feasible loads for the given driver. This costing information will be based on  $t_{\text{eta}}$ . The simulated dispatcher then recosts potential assignment arcs from the optimizer based on the early arrival, and makes an assignment selection using the procedure described in Section 6.3.

Alternatively, if  $t_{\text{actual}} > t_{\text{eta}}$ , the driver will arrive at the load destination and be available for a new assignment later than expected. When the simulation clock reaches and then passes time  $t_{\text{eta}}$ , the optimizer continually resets  $t_{\text{eta}}$  to be the current time.

Randomness in travel times can have the effect of encouraging greedy solutions. Consider the situation where drivers 1 and 2 are available now, and driver 3 will become available later in the afternoon. Driver 1 is 40 miles away from a load, driver 2 is 90 miles away, and driver 3, when he arrives, would only be 20 miles away. Assume that driver 3's estimated time of arrival is 3 P.M., and the load must be picked up before 5 P.M. A network model would choose driver 3 to cover the load based on cost and service considerations. In this case, we might assign driver 1 to another load. Later, we might discover that the ETA for driver 3 is inaccurate, and that he will be delayed several hours. In this case, we have no choice but to assign driver 2 to the load, at a much higher cost.

## 7. EXPERIMENTAL DESIGN

OUR PRIMARY INTEREST is in determining the value of optimal (myopic) solutions over local solutions under different sources and degrees of uncertainty. We measure the degree to which a decision is global versus local through the dual discount parameter  $\alpha$ . We consider three sources of randomness: a) randomness in the dynamic arrival of loads to the system; b) randomness in the travel times, captured as errors in our estimates of driver ETA's; and c) randomness in the implementation of recommendations by the user, which we refer to as the user noncompliance problem. With each source of randomness, we focus on finding the value of  $\alpha$  that minimizes the simulation cost function. In the process, we also

obtain estimates of the cost of each source of uncertainty.

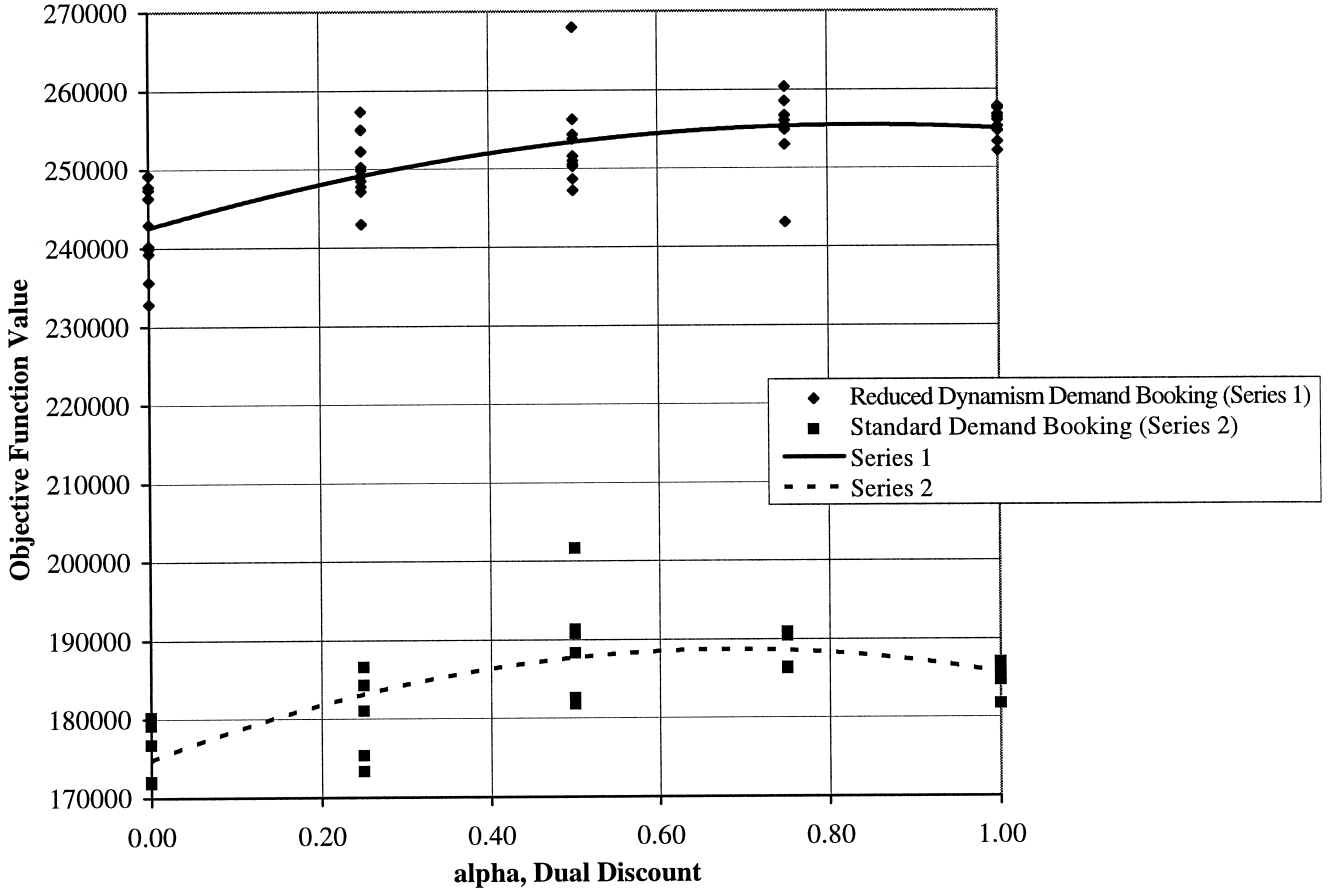
We measure the value of global over local solutions through the dual utilization factor  $\alpha$ . Thus, our experiments will focus on determining the optimal value of  $\alpha$  under different degrees and sources of randomness.

The experimental data used for our runs was obtained from a major truckload motor carrier. A simulated fleet consisting of 400 drivers was constructed by randomly picking from a data set consisting of the carrier's full staff of about 1500 drivers. An initial set of 100 loads was constructed for input into the system. Subsequent to these initial 100 loads, the demand generator was set to provide approximately 225 loads per day of simulation time. Both the initial and subsequently generated load sets were constructed by sampling from a set of actual historical service demands called into the carrier over a one-week period. Both the driver and load sets were distributed throughout the continental U.S. Each run simulated seven days of actual dispatching, run at a speed of 60 times real time, resulting in a run time of 2.8 hours per run. For each combination of parameter values simulated, with the exception of the advance demand booking experiment discussed in Section 8.1, we conducted five simulation runs. For the advance demand booking experiment, 10 runs were conducted for each parameter value studied. This was done to clearly delineate solution behavior for this special case.

For each run, we statistically estimated a second-order polynomial relating total simulation profits (given by Eq. 7) to the dual discount factor  $\alpha$ . We retained this second-order specification throughout the analysis.

Our experimental design is based on running simulations with different values of  $\alpha$  and  $p^d$ , along with other sources of randomness, and drawing conclusions regarding the appropriate decision function (as specified by the parameter  $\alpha$  in Eq. 6). Thus, we need to think about what it means to compare values of  $F(\alpha, p^d, \rho)$  in Eq. 7 for different values of  $p^d$  and  $\alpha$ . Three issues need to be discussed. First is the role of the utility parameter vector  $\rho$ , and what it means to optimize over a mixture of soft and hard costs. Second is the issue of comparing model runs for different values of the user acceptance parameter  $p^d$ . Finally, we discuss the important issue of comparing model runs for different values of  $\alpha$ .

The utility vector  $\rho$  controls the relative importance of different dimensions of the cost function. As we discussed above, our costs include both quantifiable transportation costs, and other soft costs related to customer service and driver management.



**Fig. 7.** Objective function versus dispatcher dual discount factor with advance demand booking and perfect user compliance ( $p^d = 1.0$ ).

Our research addresses the challenge of finding a decision function  $x^\pi$  that maximizes  $F(\alpha, p^d, \rho)$  within the class of decision functions defined by Eq. 6, parameterized by  $\alpha$ . At this point, we take as given the utility function parameterized by the vector  $\rho$ . Thus, a dollar of lateness counts the same as a dollar from traveling empty. In the context of our research, this is the correct way to run the simulation. The choice of the vector  $\rho$  is set by management (in our case, this is the vector running in production at a major motor carrier). It is not our job to question whether we have the correct value of  $\rho$ , which has been chosen to balance the importance of issues such as on-time service and routing drivers through their home with operating costs. Our simulations cover a week of dispatching; tradeoffs of empty miles and service, in contrast, are learned over years of watching customers switch to competitors because of poor service. Our challenge, then, is to maximize  $F(\alpha, p^d, \rho)$  for a given choice of  $\rho$ . For this reason, we have made no effort to run simulations with different values of  $\rho$ .

The second issue concerns comparing model runs

using different values of the user acceptance probability,  $p^d$ . This question raises a very subtle issue with regard to interpreting the objective function  $F(\alpha, p^d, \rho)$ . Although we have a well-defined objective function, the issue of user compliance brings into focus the difference between the objective function we are maximizing, and the one the users are maximizing. Assume that we wanted to compare  $F(\alpha, p_1^d, \rho)$  to  $F(\alpha, p_2^d, \rho)$  where  $p_1^d > p_2^d$ . We might wish to estimate the increase in profits if the user acceptance level increased. In fact, this comparison is not meaningful, and we cannot use our results to estimate the value of higher user compliance. To see this, consider a run with  $p_1^d = 1$  versus  $p_2^d = 0.7$ . At  $p_1^d = 1$ , we are accepting all the recommendations of the model. In general, these recommendations are the highest profit options and will produce the highest profits over a simulation. If  $p_2^d = 0.7$ , then we are, in effect, saying that 30% of the time, a top-ranked recommendation is not acceptable. Thus, the supposedly high profit loads that appear at the top of the list for a driver are, in fact, low profit (measured in terms of our utility function) for reasons

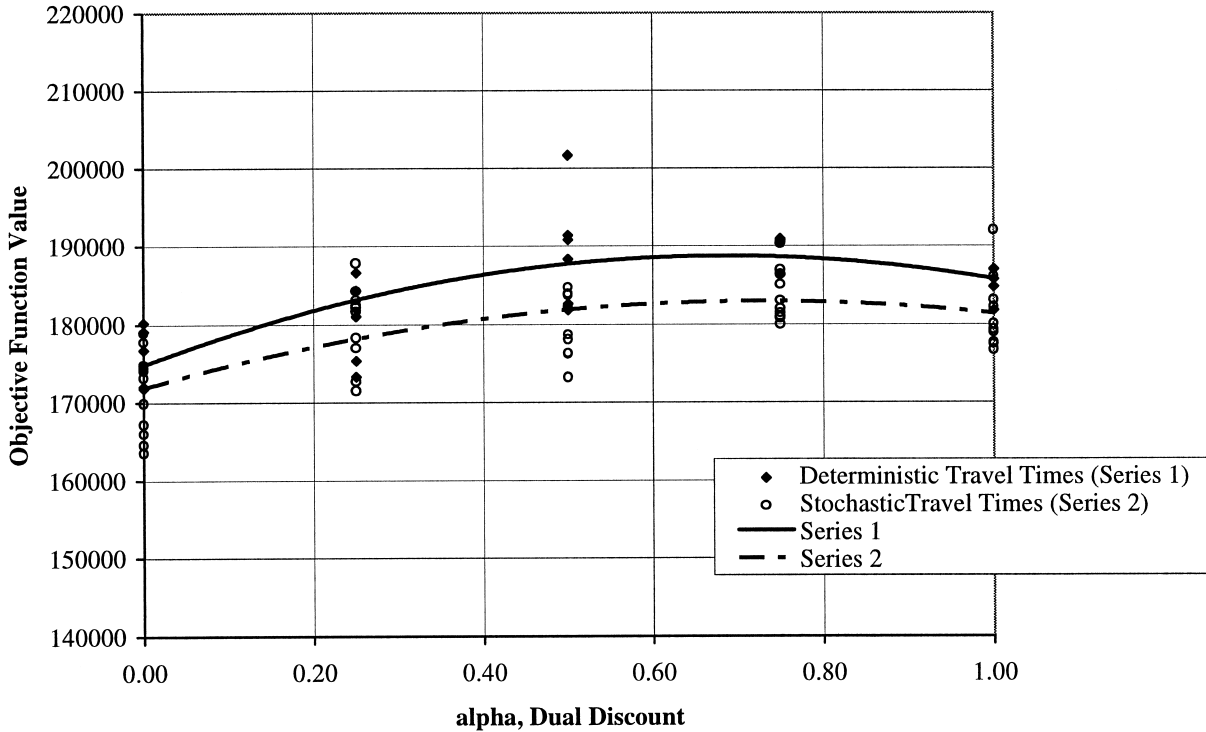


Fig. 8. Objective function versus dispatcher dual discount factor, deterministic and stochastic travel times ( $p^d = 1.0$ ).

that are not captured by the model. It would be tempting to use our simulations to estimate the value of always using the model versus only sometimes using the model. We feel that our simulations do not allow us to draw any conclusions along these lines.

Finally, we address the third and most important issue of comparing values of  $F(\alpha, p^d, \rho)$  for different values of  $\alpha$ . Here, we are using the same user acceptance probability  $p^d$ . For a fixed value of  $p^d$  we will, for any given value of  $\alpha$ , be searching, on average, the same distance down any list of loads offered to a driver. As long as  $p^d$  is held fixed, we are capturing the same degree of dissatisfaction with the higher ranked options for a driver. When we change  $\alpha$  while holding  $p^d$  fixed, we are, in effect, solving the same problem with different algorithms.

The parameters  $\rho$  and  $p^d$ , then, represent characteristics of the problem.  $\alpha$ , in contrast, is a characteristic of the algorithm. We may compare  $F(\alpha, p^d, \rho)$  directly for different values of  $\alpha$ , and we may also compare the optimal value of  $\alpha$  for different values of  $p^d$  and  $\rho$ , but we may not compare  $F(\alpha, p^d, \rho)$  directly for different values of  $p^d$  or  $\rho$ . Because a major focus of our research is the impact of user noncompliance, it is interesting to estimate the best value of  $\alpha$  for different values of  $p^d$ . We did not feel that it would be very interesting to ask the same question for different parameter vectors  $\rho$ .

## 8. EXPERIMENTAL RESULTS

WE NOW PRESENT the results of our simulations. First, Section 8.1 describes the results of our experiments that focus on the effects of randomness in the booking process. It is in these runs that we produce results that show that, if the demands are known in advance, and travel times are deterministic, and user compliance is 100%, then the optimal myopic solution ( $\alpha = 1$ ) produces the best results. This is an important piece of validation.

Next, Section 8.2 considers the impact of uncertain travel times. Finally, Section 8.3 presents the experiments that summarize most carefully the value of greedy, optimal, and intermediate solutions in the presence of varying levels of user compliance.

### 8.1 Effect of Dynamic Service Demand Booking

Our first set of experiments included a set of runs with the lowest level of uncertainty among all our experiments. In this run, we assumed all loads to be called in during the day were known at 7 A.M. on that day (but we do not know about tomorrow's loads). We assumed deterministic travel times and perfect user compliance. We then ran simulations with different values of  $\alpha$ , expecting to find that the best value of  $\alpha$  was very close to 1.0. For this set of

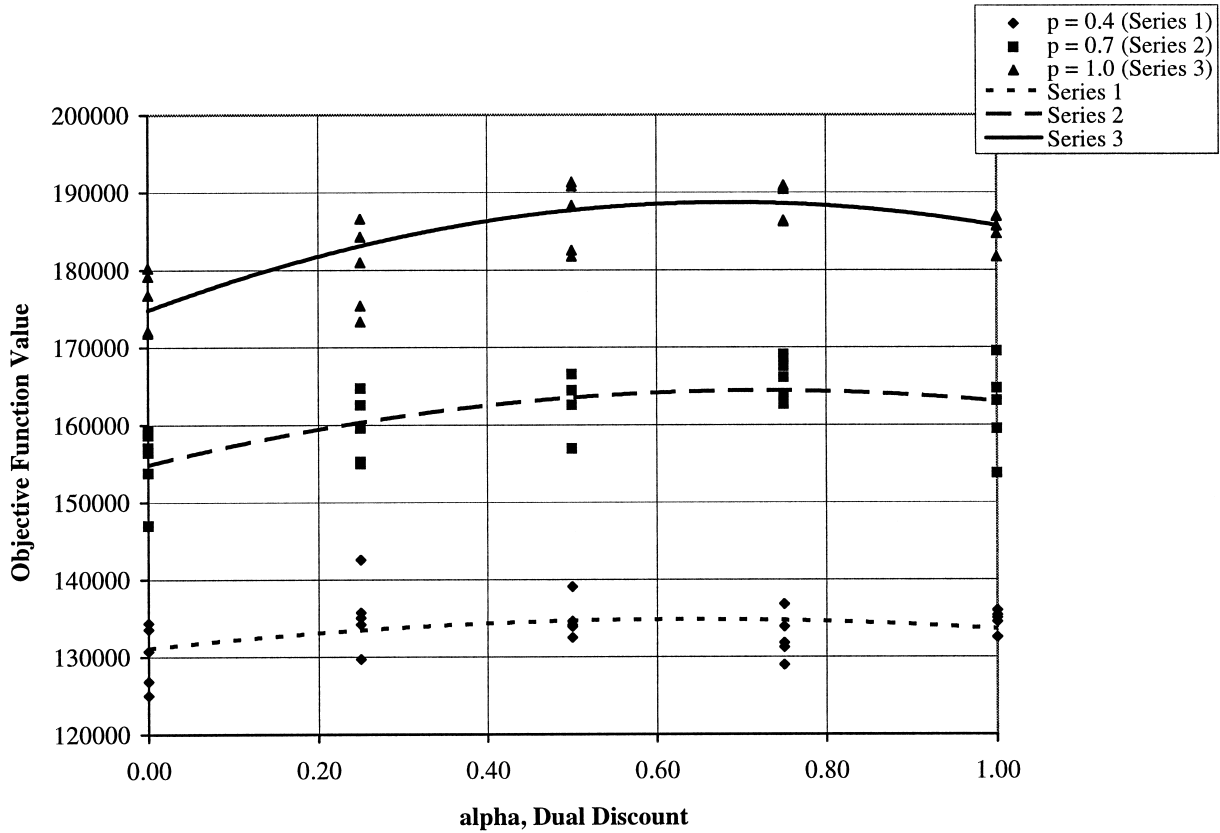


Fig. 9. Objective function versus dispatcher dual discount factor, various probabilities of selection.

experiments, we performed 10 samples for each value of  $\alpha$  that was tested.

The results are shown in Figure 7, which compares total profits versus  $\alpha$  for two sets of data: the first assumes all the daily bookings are known at the beginning of the day, and the second retains the original booking profile of loads. It is not surprising that we get higher profits when all the demands are known at the beginning of the day. Most significant is that, when demands are known in advance, the curve steadily increases with  $\alpha$ , leveling off for values of  $\alpha$  over approximately 0.60. For this run, we could not reject the hypothesis that the optimum dual discount factor was equal to 1.0, as we would expect. When we simulate dynamic booking (the lower curve), performance noticeably dropped for values of  $\alpha$  over 0.80, with an apparent optimum close to 0.75. This is our first evidence that, in the presence of random customer booking, the global optimum solution does not give the best overall results.

## 8.2 Effect of Network Stochasticity

The second set of experiments we performed examined the effect of stochasticity in fleet operations

in the form of random travel times between load origin and destination locations. The random travel times were implemented as described in Section 6.6. The objective function values for these runs were compared to those obtained using only expected travel times.

For these experimental runs, we set  $p^d = 1.0$ . The noise intensity parameter,  $\beta$ , was set to 1.0. The time window start and end parameters,  $w_s$  and  $w_e$ , respectively, were both set to 1.5. This means that a 600-mile move that takes an average of 16 hours might require anything from 10 to as much as 22 hours. While drivers in a real-world situation tend to run late more often than early, we wanted the expectation of the random travel time to equal the expected travel time. This was done to avoid influencing the objective function value by changing the relative number of driver-load assignments made during a simulation. Had the window been set to always make travel times at least as long as the expected time, for example, the fact that drivers would always be at best on-time and usually late would mean that fewer assignments could be made in the course of a simulation.

The results are shown in Figure 8. These show



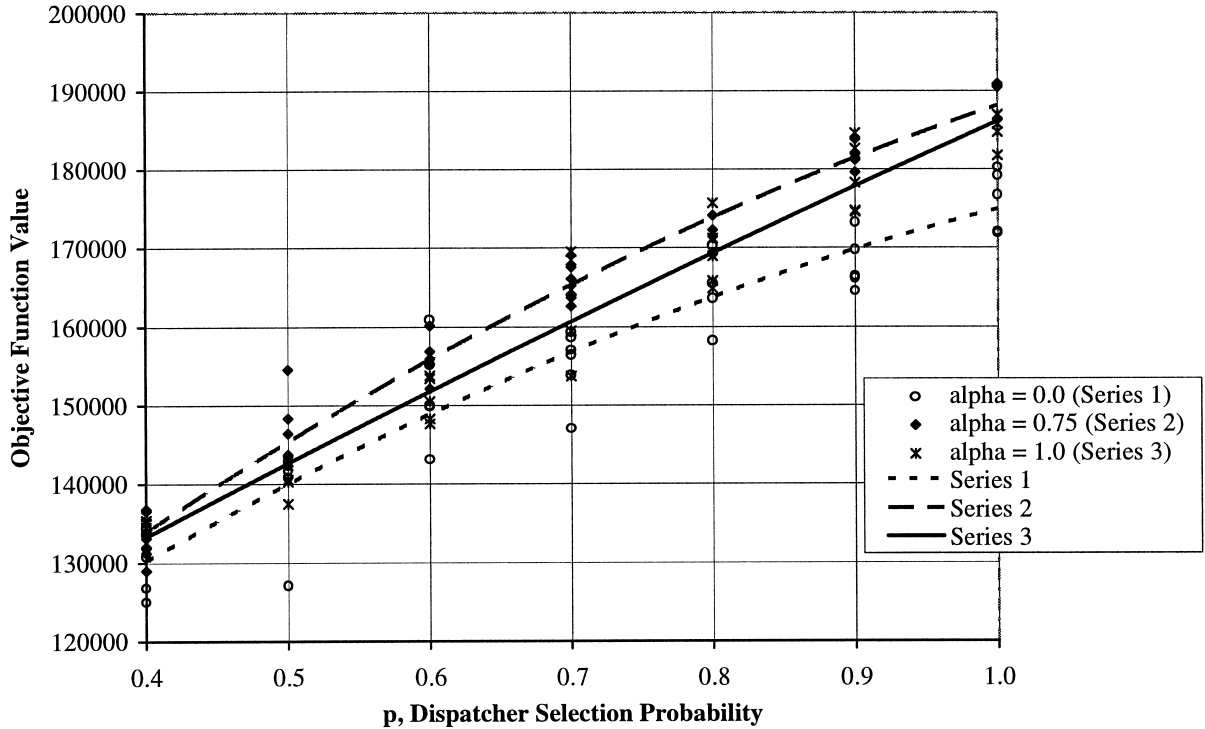


Fig. 10. Objective function versus dispatcher selection probability, various dual discounts.

that randomness in travel times has a relatively modest (but significant to a motor carrier) impact on overall profits. We note that the impact of  $\alpha$  is diminished in the presence of random travel times. More importantly for our research, the shape of the curve as a function of  $\alpha$  appears not to have an impact on the optimum dual discount factor. Of course, both runs were done in the process of dynamic booking, so we are, in effect, observing that the randomness in customer bookings is more important than randomness in travel times. The optimum value of  $\alpha$  still appears to be around 0.75.

### 8.3 Effect of User Noncompliance

Our last set of experiments looked at the effect of user noncompliance on system performance. We report these results in two ways. First, Figure 9 summarizes system profits as a function of  $\alpha$  for different values of  $p^d$ . Our first conclusion from these runs is that the optimum value of  $\alpha$  is close to 0.75 for all three values of  $p^d$ . Our second and more significant conclusion is that, for the lowest levels of user compliance,  $p^d = 0.4$ , the dual discount has relatively little effect on overall profits. Said differently, the value in listing loads sorted in the order from best to worst (measured with respect to  $c_{dlt}(\alpha)$ ) is significantly reduced. As user compliance rises, the value of a proper listing of loads increases, although the

globally optimal solutions appear to slightly underperform solutions using  $\alpha = 0.75$ .

A different perspective on the results was obtained by plotting total profits as a function of different values of  $p^d$ , using  $\alpha = 0, 0.75$ , and  $1.0$ , shown in Figure 10. The results show that, as the level of user compliance drops, the value of a globally optimal solution over a greedy solution drops dramatically. In the truckload motor carrier industry, it is not unusual to see values of  $p^d = 0.6$  or even lower. A value of  $p^d = 0.7$  is considered very high. These results show that, for these levels of user compliance, greedy solutions work almost as well as globally optimal solutions. Furthermore, using  $\alpha = 0.75$  gives the best results over all levels of user compliance, although, for the lowest levels, it really does not matter what you do. Reassuringly, if user compliance is high (over 90%) then the value of global network information (either with  $\alpha = 1.0$  or  $0.75$ ) is quite high, and demonstrates the usefulness of optimization models.

## 9. CONCLUSIONS

THE EXPERIMENTS in this paper investigate the impact of different forms of uncertainty on the value of optimal (myopic) solutions in a particular class of routing and scheduling problems. The results show

that uncertainty does reduce the value of globally optimal solutions, bringing into question whether rigorously optimal solutions are useful in a dynamic setting.

Our research did not compare myopic and dynamic models, but rather looked only at models that consider only information that is known (or reasonably well known, in the case of random travel times) at a given point in time. This said, even a myopic model plans for decisions in the future, as long as it is based on known information. In our case, we would consider the global optimization of all drivers to all loads, including drivers that would not arrive until later in the day or the next day. Our dual discount, then, implicitly was putting a weight of 1.0 on the decision we were about to make, and a reduced weight on decisions that we might make at a later point in time (although it might only be a few seconds later).

We are unaware of any other research that has raised the issue of user compliance, and measured its impact on the choice of algorithm. Even static models are subject to user editing and modification prior to implementation. Given such postoptimality tampering, it is clear that the original solution is no longer optimal, because it assumes that all the decisions in the solution will be implemented. We would argue that such considerations should be built into the design and analysis of algorithms for all classes of routing and scheduling problems. The result may be problems that are actually easy to solve, producing both faster run times and more robust solutions.

#### ACKNOWLEDGMENT

THIS RESEARCH WAS SUPPORTED in part by grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research. The authors would also like to gratefully acknowledge the careful reading by an anonymous referee who helped to highlight important issues in this study.

#### REFERENCES

- S. ALBERS, "Competitive Online Algorithms," *OPTIMA: Math. Prog. Soc. Newsl.* **54**, 1–8 (1997).
- D. BERTSIMAS AND G. V. RYZIN, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane," *Opns. Res.* **39**, 601–615 (1991).
- D. J. BERTSIMAS AND D. SIMCHI-LEVI, "A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty," *Opns. Res.* **44**, 286–304 (1996).
- D. J. BERTSIMAS AND G. VAN RYZIN, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane," *Opns. Res.* **39**, 601–615 (1991).
- D. J. BERTSIMAS, P. CHERVI, AND M. PETERSON, "Computational Approaches to Stochastic Vehicle Routing," *Transp. Sci.* **29**, 342–352 (1995).
- R. CHEUNG AND W. B. POWELL, "An Algorithm for Multistage Dynamic Networks with Random Arc Capacities, with an Application to Dynamic Fleet Management," *Opns. Res.* **44**, 951–963 (1996).
- J. DESROSIERS, M. SOLOMON, AND F. SOUMIS, "Time Constrained Routing and Scheduling," in *Handbook in Operations Research and Management Science*, C. Monma, T. Magnanti, and M. Ball (eds.), Volume on *Networks*, 35–139, North Holland, Amsterdam, 1995.
- M. DROR, G. LAPORTE, AND P. TRUDEAU, "Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks," *Transp. Sci.* **23**, 166–176 (1989).
- M. FISHER, "Vehicle Routing," in *Handbook in Operations Research and Management Science*, C. Monma, T. Magnanti and M. Ball (eds.), Volume on *Networks*, 1–33, North Holland, Amsterdam, 1995.
- L. FRANTZESKAKIS AND W. POWELL, "A Successive Linear Approximation Procedure for Stochastic Dynamic Vehicle Allocation Problems," *Transp. Sci.* **24**, 40–57 (1990).
- R. GRAHAM, "Bounds for Certain Multiprocessor Anomalies," *Bell Syst. Tech. J.* **45**, 1563–1581 (1966).
- D. GUSFIELD AND R. IRVING, *The Stable Marriage Algorithm: Structure and Algorithms*, MIT Press, Boston, MA, 1989.
- W. JORDAN AND M. A. TURNQUIST, "A Stochastic Dynamic Network Model for Railroad Car Distribution," *Transp. Sci.* **17**, 123–145 (1983).
- G. LAPORTE AND F. LOUVEAUX, "Formulations and Bounds for the Stochastic Capacitated Vehicle Routing Problem with Uncertain Supplies," in *Economic Decision-Making: Games, Econometrics and Optimization*, J. Gabzewicz, J. Richard, and L. Wolsey (eds.), North Holland, Amsterdam, 1990.
- M. PINEDO, *Scheduling: Theory, Algorithm, and System*, Prentice Hall, Englewood Cliffs, NJ, 1995.
- W. B. POWELL, "A Stochastic Model of the Dynamic Vehicle Allocation Problem," *Transp. Sci.* **20**, 117–129 (1986).
- W. B. POWELL, "An Operational Planning Model for the Dynamic Vehicle Allocation Problem with Uncertain Demands," *Transp. Res.* **21B**, 217–232 (1987).
- W. B. POWELL, "A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem," in *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad (eds.), 249–292, North Holland, New York, 1988.
- W. B. POWELL, "A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers," *Transp. Sci.* **30**, 195–219 (1996).
- H. PSARAFITIS, "Dynamic Vehicle Routing Problems," in *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad (eds.) 223–248, North Holland, Amsterdam, 1988.

- P. SCHRIJVER, *Supporting Fleet Management by Mobile Communications*, P. R. Schrijver, The Netherlands, 1993.
- D. D. SLEATOR AND R. E. TARJAN, "Amortized Efficiency of List Update and Paging Rules," *Com. ACM* **28**, 202–208 (1985).
- W. STEWART AND B. GOLDEN, "Stochastic Vehicle Routing: A Comprehensive Approach," *Eur. J. Opns. Res.* **14**, 371–385 (1983).
- P. TRUDEAU AND M. DROR, "Stochastic Inventory Routing: Route Design with Stockouts and Route Failures," *Transp. Sci.* **26**, 171–184 (1992).
- W. WHITE, "Dynamic Transshipment Networks: An Algorithm and Its Application to the Distribution of Empty Containers," *Networks*, **2**, 211–236 (1972).

(Received: August 1997; revisions received: July 1998, September 1998; accepted: October 1998)