A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers

WARREN B. POWELL

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

The dynamic assignment problem arises in a number of application areas in transportation and logistics. Taxi drivers have to be assigned to pick up passengers, police have to be assigned to emergencies, and truck drivers have to pick up and carry loads of freight. All of these problems are characterized by demands that arrive continuously and randomly throughout the day, and require a dispatcher to assign a driver to handle a specific demand. We use as our motivating application the load matching problem that arises in long-haul truckload trucking, where we have to assign drivers to loads on a real-time basis. A hybrid model is presented that handles the detailed assignment of drivers to loads, as well as handling forecasts of future loads. Numerical experiments demonstrate that our stochastic, dynamic model outperforms standard myopic models that are widely used in practice.

Consider the problem of dynamically assigning drivers for a truckload motor carrier to handle loads that arise randomly over time. At any instant in time, we have a set $\mathcal{D}$ of drivers to be assigned to a set $\mathcal{L}$ of loads, where the size of $\mathcal{L}$ may be greater or smaller than $\mathcal{D}$. The situation is illustrated in Figure 1 which shows drivers (representing units of capacity) and a series of loads to be covered. The problem is to decide what driver to assign to each load. Loads are characterized by a) start and ending time window, b) origin and destination (the origin and destination of each task may be the same), c) duration (which might include travel time if the origin and destination are different) and d) a vector of required or desirable attributes for the driver being assigned to handle the load. Drivers, on the other hand, are characterized by a) time of availability (which may be in the future, if the driver is currently utilized on a task, or the past, if the driver became available earlier), b) location at time of ETA, and c) a vector of attributes which might include hours of service, desired time off, training (which determines which tasks he can cover), equipment available (special type of tractor or trailer, special tools), and home domicile.

The load matching problem for truckload motor carriers is an instance of a resource allocation problem where a complex resource (a driver) needs to be assigned to tasks (loads) that arise randomly over time. While demands are random, they do arise in a somewhat predictable way. In particular, the distribution of demands over time and space is inhomogeneous. Furthermore, if a demand cannot be serviced within a reasonable period of time, it is lost to the system, with the associated penalties that accompany poor service (loss of revenue, loss of goodwill).

The need to reposition resources in anticipation of future demands arises in problems like taxi service or truckload trucking (where a driver is responsible for moving a full load from one point to another), where the origin and destination of the task is different. In such problems, the flow inbound to a particular region or area is often not equal to the flow out of a region, creating over time imbalances in the allocation of capacity over space. In some instances, these imbalances can be handled in real-time using a process known as *checkerboarding*, which is illustrated in Figure 2. In this figure, region A has one truck and no loads, while region B has one truck and two loads. If we have assigned drivers greedily to loads, it is likely that we would have ended up with a solution where there would have been no drivers to cover the second load in region B. However, by per-

Transportation Science
Vol. 30, No. 3, August 1996

0041-1655/96/3003-0195 $01.25

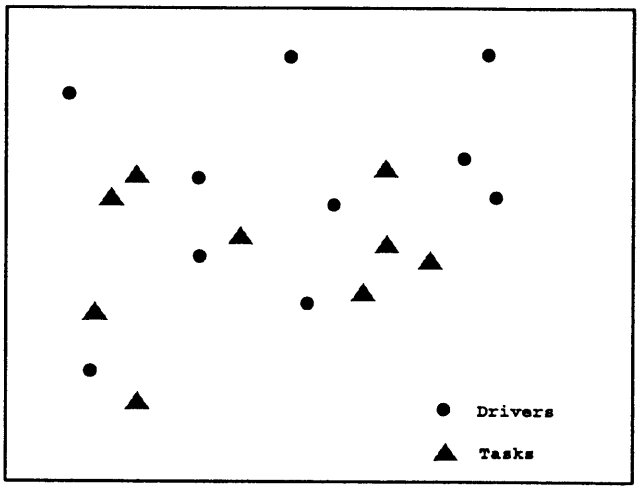

Fig. 1. Spatial illustration of a dynamic assignment problem.

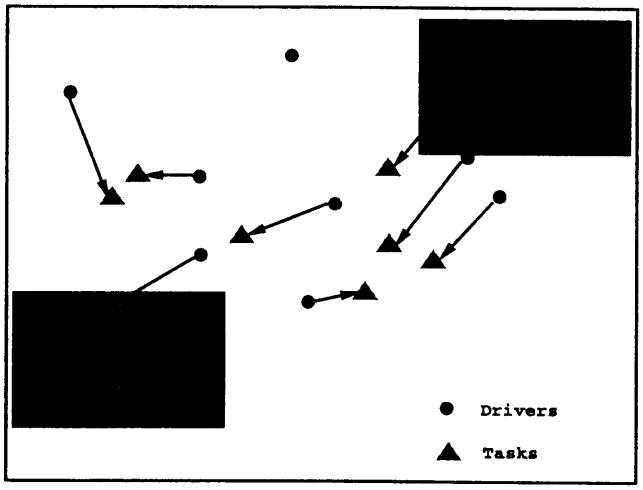

Fig. 2. Illustration of checkerboarding in an assignment problem.

forming a global assignment, we are able to effectively shift drivers from region A to region B, often by assigning drivers to loads other than the closest load. This solution assumes, however, that all the drivers can move the longer distances and still pick up their loads on time. In some instances, it is necessary to move excess capacity before all the demands are known.

This paper accomplishes the following tasks:

1. We describe basic truckload motor carrier operations, and summarize the key dimensions of the dynamic fleet management problems for truckload motor carriers.
2. We review alternative models for the dynamic assignment problem, and discuss their strengths and weaknesses.
3. We develop a stochastic model of the dynamic booking process for truckload motor carriers, and show how it can be integrated into a stochastic network model for planning vehicle movements.
4. We develop a methodology for evaluating and testing dynamic fleet management models in a continuous time setting using rolling horizon simulations. This methodology is implemented in a system called MIDAS, which is described and used to test both the speed of the model and the quality of the resulting solution.

The primary methodological contributions of this paper include:

1. A new model is developed for solving (approximately) the stochastic, dynamic assignment problem in continuous time and space. The resulting model combines actual and forecasted demands in an integrated way that allows the model to *react* to real demands at the same time that it *anticipates* forecasted demands. We believe this model is the first model proposed for the stochastic, dynamic assignment problem in a continuous time, continuous space setting.
2. The hybrid model can be solved as a pure network, which allows it to be used in a real-time setting. We show that the model is computationally fast enough to handle the largest problems that may arise in actual applications. The model also possesses the flexibility to accommodate a variety of real-world issues.
3. Extensive simulation experiments demonstrate that the model outperforms myopic models which are widely used in practice. Our experiments show that this effect is most pronounced over relatively long periods of time (several weeks of dispatching).

Finally, using the fleet simulator, we are able to make substantive contributions to our understanding of the economics of fleet operations:

1. By simulating fleets of different size, we were able to estimate the economies of density for truckload motor carriers. The results demonstrate that carriers with fleets of over 1,000 trucks have a significant cost advantage over carriers with fleets under 500 trucks. This is the first quantitative estimate of the value of increased density for the truckload industry reported in the research literature.
2. By simulating the effects of different demand booking profiles, we were able to estimate the cost of uncertainty in demand forecasts, and the value of having shippers provide advance notice regarding demands. We provide experimental ev-

idence to support the hypothesis that the value of this advance information is minimal.

Although the model is developed for truckload motor carriers, we believe some of the ideas expressed here can be applied to other dynamic fleet management problems. The model should also be of interest to researchers in stochastic programming since this is the framework we use to handle uncertainties in forecasted demands. Also, since the model is an approximation, we feel that other researchers may be able to contribute insights which may further improve the model.

We are not aware of any papers which directly address the real-time dispatching problem of truckload motor carriers. BROWN and GRAVES (1981) present an integer linear programming formulation of a real-time routing and scheduling problem for petroleum tank trucks. The model develops truck tours for known (deterministic) customer demands (the problem they consider, however, is much more complex than our problem). BELL ET AL. (1983) describes a set-partitioning formulation for a similar problem (routing and scheduling of tanker trucks in real time). GAVISH (1981) describes an optimization-based, hierarchical model for real-time routing and scheduling. All of these papers present optimal algorithms or near optimal heuristics (actually, even the optimal methods produced suboptimal results due to heuristic pruning of the solution space) for use in a real-time environment. At the same time, the underlying models are all essentially static and deterministic, running on periodic snapshots of data, and do not incorporate forecasted demands.

Considerably more attention has been given to handling forecasted demands in the literature on the dynamic vehicle allocation problem. Early papers treated forecasted demands as deterministic (WHITE (1972), MAGNANTI and SIMPSON (1978)) and focused on the development of specialized algorithms (WHITE and BOMBERAULT (1969), ARONSON and CHEN (1986)). More recently, a series of papers have considered models which explicitly treat uncertainty in demand forecasts. JORDAN and TURNQUIST (1983) first introduced stochastic demands using a distribution formulation, and represented stochastic inventories of vehicles using normal distributions. POWELL (1986) extended this model to the full dynamic vehicle allocation problem (which modeled stochastic loaded movements). Later, POWELL (1988) showed how this problem could be modeled as a multistage dynamic network with random arc capacities. Methods for approximating the expected recourse function for this problem have since been developed (FRANTZESKAKIS and POWELL (1990), POWELL and CHEUNG (1994a,b), CHEUNG and POWELL (1995)) and shown to outperform deterministic models. CRAINIC ET AL. (1993) proposes a stochastic dynamic model for the allocation of empty containers, again using the distribution form introduced by Jordan and Turnquist (repositioning empty containers without tracking loaded movements). They propose, but do not test, a stochastic linearization approach for incorporating forecasted demands.

A separate but related literature has evolved around the dynamic traveling repairman problem. Like the dynamic vehicle allocation problem, the DTRP has the elements of a foundation problem for dynamic routing. A series of papers (BERTSIMAS and VAN RYZIN (1991, 1993) and BERTSIMAS and HOWELL (1993) have provided a probabilistic analysis of relatively simple heuristics for solving the dynamic traveling repairman problem. Thus, the underlying models used in these papers are essentially myopic in nature.

A limited literature has evolved around stochastic formulations for a stochastic version of the classical vehicle routing problem (see DROR ET AL. (1989), DROR and TRUDEAU (1986), LAPORTE and LOUVEAUX (1990), LAPORTE ET AL. (1992), STEWART and GOLDEN (1983), and TRUDEAU and DROR (1992). However, all of these papers focus primarily on a two-stage formulation where vehicle routes are designed prior to knowing customer demands.

Section 1 provides a description of truckload motor carrier operations, including an overview of the dynamic fleet management problem, and an introduction to basic driver dispatching. The following sections describe a series models for the dynamic assignment problem. Section 2 presents a static network assignment model for handling driver assignment. Section 3 describes a deterministic, dynamic network model, followed in Section 4 by a stochastic formulation of the same model. Section 5 reviews methods for approximating the expected recourse function, and Section 6 shows how a piecewise linear separable approximation yields a pure network for the first stage problem. Finally, a hybrid model is introduced in Section 7 which combines the strengths of all three models. Section 8 describes the experimental environment used to run the tests, covering the generation of demands, the nature of the datasets used, and the running of the experiments. Section 9 then presents the results of experiments to answer a series of questions: What is the value of a dynamic model over a myopic one? What is the cost of uncertainty? And what are the economies of density in the truckload industry?

1. DYNAMIC FLEET MANAGEMENT FOR TRUCKLOAD MOTOR CARRIERS

TRUCKLOAD MOTOR CARRIERS represent a relatively simple operation in transportation. Basically, a shipper calls a carrier and requests that a driver come by and pick up a load. The load may be in the form of a trailer that is already loaded with freight, or it might represent a consignment of freight that must be loaded into a trailer. In the first case, the carrier must send a driver with a tractor (the driver might also bring a trailer that will be dropped off at the shippers trailer pool); in the second, the driver must bring an empty trailer that will then be filled with freight.

Once the trailer is loaded, it is the job of the carrier to move the load from origin to destination. In most cases, this simply involves having the driver drive the load to the destination over the road network. In some instances, the driver might bring the trailer back to a terminal and turn it over to another driver who will handle the movement. Many loads must be delivered quickly and therefore must be handled by a single driver (or a team of two drivers, which allows the trailer to move without stopping for rest). Some loads might have several days of slack built into the schedule, allowing or even forcing the carrier to store the trailer at a yard (this creates both problems and opportunities for the carrier).

After the load is delivered, the driver becomes available to be assigned to a new demand. It is the responsibility of the carrier to find work for the driver, who is typically paid by the mile. If the load terminated in a region with low demand (relative to the supply of trailers) the carrier must decide between holding the driver in the region until a demand does arise, or repositioning him empty to another region which is more promising.

Elements of the fleet management problem can be divided into supply (or capacity) management (serving the customer) and demand management (controlling the demands placed on the carrier). The supply management problem includes:

- Determining what driver to assign to handle a demand.
- Repositioning excess capacity from one region to the next.
- Determining how a load is to be handled once it is picked up. A driver may simply move the load directly from origin to destination. Alternatively, the carrier may be able to hold the load for a period of time at a location, and assign a new driver to move the load to the final destination. In the extreme, a carrier could move the load using several drivers over a *relay network* where each driver simply moves the load a single leg (this allows the carrier to keep drivers close to home).
- Managing the flows and inventories of trailers required to satisfy shipper demands.

An important dimension of the capacity management problem is driver management. The choice of what driver to assign to a load, repositioning empty drivers, and the routing and scheduling of the driver while moving a load, must take into consideration factors such as:

- Driver work rules—Federal laws, and in some cases local work rules, limit the number of hours a driver can work at any one time. The most important rules are: a) the driver cannot be on the road for more than 10 hours during a single work shift, b) the driver cannot be on duty (driving plus other on-duty activities) for more than 15 hours before an eight hour rest period, and c) the driver cannot work more than 70 hours (time on duty) in any eight-day stretch. Since a driver can be on duty up to 15 hours in a day, this last constraint is often binding (a driver that has worked 65 hours in the last seven days can only work 5 hours on the eighth day).
- Driver pay—Drivers are only paid when they work, and therefore the carrier must be careful to balance the workload across drivers (this is especially important during low demand periods).
- Returning drivers home—Drivers in truckload trucking may easily spend anywhere from one to four weeks away from home, although most carriers try to return a driver home at least every two weeks. Assigning drivers to loads that bring the driver home, then, is an important criterion in the choice of load assignment.
- Driver skills—A load may require special skills, such as experience handling hazardous materials or oversize loads, or expertise with customs for cross-border loads.
- Driver quality—Some shippers expect or require drivers with good customer-relation skills.

Demand management is an important side of the fleet management problem. Demands may arise either because the shipper calls the carrier (of course, this is the most common mechanism) or through a call to the shipper initiated by the carrier. Shippers can further be divided into three broad groups: a) primary shippers, where the carrier is effectively obligated to move any load tendered by the shipper, b) secondary shippers, where the carrier is obligated

to move loads only in certain traffic lanes (for example, the carrier may only take the shipper's northeast freight), and c) tertiary shippers, where the carrier may accept or reject any piece of freight.

Under these conditions, the carrier must manage two important aspects of the fleet management problem, which include:

- Load acceptance/rejection—For secondary and tertiary shippers, a carrier may accept or reject certain loads based on real-time capacity availability or system balance considerations.
- Load solicitation—The carrier may wish to aggressively solicit freight out of specific regions or in specific lanes to correct short term balance problems.

Load acceptance/rejection and load solicitation represent two important tools in a carrier's arsenal to influence the demands on the carrier. They require that a carrier be able to quickly evaluate the profitability of "spot" loads (loads tendered by secondary and tertiary shippers), anticipate areas of surplus or deficit, and trade off the profitability of current versus future demands (especially when the current demand is from a tertiary shipper, while the future demands might be from a primary account).

2. A DETERMINISTIC ASSIGNMENT MODEL

SOME OF THE MOST important elements of the problem of assigning drivers to loads can be captured by a simple assignment model. Consider a snapshot of available drivers and loads as depicted in Fig. 1. Let

$\mathcal{L}$ = set of loads,
$\mathcal{D}$ = set of drivers,
c^a_{dl} = cost of assigning driver d to load l,
c^r_l = cost (penalty) from not assigning any drivers to load l (cost of a load rejection),
c^h_d = cost of holding driver d, and not assigning him to any load,
$x^a_{dl} = \begin{cases} 1 & \text{if driver } d \text{ is assigned to load } l \\ 0 & \text{otherwise,} \end{cases}$
y_l = 1 if load l is refused (no driver is assigned to pick it up),
x^h_d = 1 if driver d is not assigned to any load.

The driver assignment problem can then be stated simply as:

$$(DAP) \quad \min_{x^a, x^h, x^r} \sum_{d \in \mathcal{D}} \sum_{l \in \mathcal{L}} c^a_{dl} x^a_{dl} + \sum_{d \in \mathcal{D}} c^h_d x^h_d + \sum_{l \in \mathcal{L}} c^r_l y_l \tag{1}$$

subject to:

$$\sum_{d \in \mathcal{D}} x^a_{dl} + y_l = 1 \quad l \in \mathcal{L}$$

$$\sum_{l \in \mathcal{L}} x^a_{dl} + x^h_d = 1 \quad d \in \mathcal{D} \tag{2}$$

$$x^a_{dl}, y_l, x^h_d \geq 0 \quad l \in \mathcal{L}, d \in \mathcal{D}$$

This, of course, is a simple assignment model with dummy supply and demand nodes to handle situations with excess drivers or excess loads. For a given set of costs, the problem of finding the optimal assignment of drivers is easily handled using an efficient network algorithm.

Variations of this basic model are now marketed commercially by software vendors to truckload motor carriers. While in its infancy (only a handful of carriers have actually implemented optimization models as of 1994), the widespread availability and use of satellite tracking and two-way mobile communications between driver and dispatcher have increased the demand for computer-assisted dispatch systems. Of particular value in the dynamic world of truckload trucking is the speed with which simple assignment models can be solved. In truckload trucking, a medium-sized carrier will have between 300 and 1000 drivers, and only two carriers have more than 5000 drivers. These problems can be solved from scratch on a modern workstation in a few seconds for medium sized carriers. Furthermore, in a real-time setting, these models can be optimized each time the data changes (a new load is called in, a driver is dispatched, a pickup or delivery appointment changes) in a fraction of this time. In fact, most of the time required to fully respond to a change in the data is consumed by reading and writing files.

The real challenge in developing the model DAP is estimating the cost coefficients. In an elementary model, we could use:

c^a_{dl} = cost of moving empty from a driver's current location d to the pickup point of the load l. This is normally the distance times a fixed cost per mile,
c^h_d = 0, if we do not wish to penalize holding, or we might use a fixed cost per hour that a driver has been waiting (even more realistic, we could use a nonlinear function to penalize making drivers sit a long time),
c^r_l = management specified "penalty" for not assigning any driver to a load (equal, perhaps, to the cost of the longest "acceptable" empty movement).

Specific, technical limitations of this cost structure include:

1. If there are not enough drivers to handle a particular load now, there may be drivers becoming available in the future that can cover the load. Thus, it is important not to set c_l^r too high.
2. Depending on the location of a driver and the number of drivers available relative to demands, the cost of holding a driver may actually be quite high, reflecting the fact that the driver will likely have to wait a long time before being assigned a load.

The real strength of the model is its flexibility which allows it to handle a wide variety of practical concerns by modifying the basic cost structure with a set of penalties and bonuses (negative costs). For example, it is possible to put an increasingly high penalty for not assigning a driver to a load so that drivers that have been waiting longer will be favored over drivers that just became available (the cost tradeoff must be set by management). Other issues that can be accommodated using similar methods include:

- Assigning drivers to loads that will help return them to their home domicile.
- Favoring drivers with more seniority or better work records. Also, placing good drivers on longer loads (which are more profitable for the driver).
- Favoring loads for special customers.
- Ensuring that driver teams (teams of two drivers that move continuously, alternating driving) are assigned to longer loads.
- Assigning drivers with tractors requiring maintenance to loads that bring the tractor near a maintenance facility.
- Assigning drivers with special skills (e.g., experience handling customers paperwork) to loads with special needs (loads headed to Canada or Mexico).

Of course, the determination of these bonuses and penalties is ad hoc, but it does provide an important degree of management participation.

In summary, the strengths of the static driver assignment model include:

1. It is easy to develop and explain.
2. It can be optimized in real-time using simple, robust network optimization codes.
3. It can be easily adapted to handle a wide variety of real-world issues.

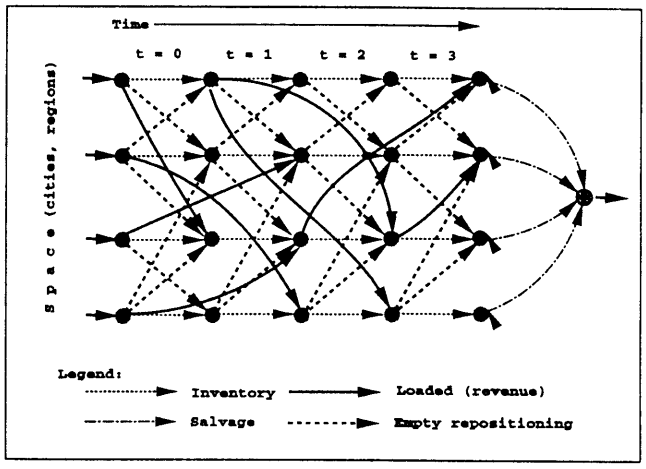

Fig. 3. Deterministic, dynamic network for fleet management.

The model does, however, suffer from several limitations as a result of the static formulation. Weaknesses include:

1. The model is unable to recommend repositioning drivers to neighboring regions in anticipation of forecasted loads.
2. The model is unable to recommend which loads should be accepted or rejected based on the profitability of the destination of the load.
3. The model can not be used to provide recommendations of where the carrier should solicit new freight to fill anticipated capacity in the future.

To be sure, all of these issues can be dealt with in an ad hoc way. However, we can consider more advanced models which explicitly handle forecasted demands and future inventories of trucks.

3. A DETERMINISTIC, DYNAMIC MODEL

WE CAN DEVELOP a dynamic model which incorporates both current and future activities. In our assignment model, we can model the exact location of each individual driver and load, using, for example, a five-digit zip code or even a street address. However, it is difficult to accurately forecast loads at this level, and as a result it is common to aggregate space into larger regions. In addition, we will aggregate time into days, and develop demand forecasts from region i to region j at time t. The resulting model is depicted in Figure 3, which uses four types of arcs:

- revenue (or demand, or loaded) arcs, which represent the task of moving loaded from a region,
- empty arcs, which represent the task of moving available capacity empty,
- holding (or inventory) arcs, which represent

holding capacity in the same region from one time period to the next,
- salvage arcs, which bring flow from each region at the end of the planning horizon into the supersink.

Supplies of vehicles enter nodes over the first few time periods, representing inventories of vehicles that either are available or are becoming available. In the basic model, all of these supplies leave through a supersink.

We start by defining:

$\mathcal{C}$ = set of cities or regions (aggregations of space), where we will use i and j to index this set,
τ_{ij} = travel time from i to j ($\tau_{ii} = 1$),
T = length of the planning horizon,
x_{ijt} = flow of loaded vehicles from i to j, departing in period t (and arriving in period $t + \tau_{ij}$),
y_{ijt} = flow of empty vehicles from i to j, departing in period t,
y_{iit} = vehicles held in region i from time period t to $t + 1$,
D_{ijt} = market demand for loads from i to j in period t,
c_{ijt} = cost of moving empty from i to j in period t,
r_{ijt} = net revenue derived from a loaded move in time period t,
R_{it} = capacity entering the network for the first time in region i at time t.

This model can be stated mathematically in two forms: a *simultaneous* form, and a *recursive* form. The simultaneous form of the problem is given by:

$$(DVA)\ \max \sum_{t=0}^{T} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} [r_{ijt} x_{ijt} - c_{ijt} y_{ijt}] \qquad (3)$$

subject to, for $t = 0, \ldots, T$:

$$\begin{aligned} \sum_{j \in \mathcal{C}} (x_{ijt} + y_{ijt}) - \sum_{k \in \mathcal{C}} (x_{ki,t-\tau_{ki}} + y_{ki,t-\tau_{ki}}) &= R_{it} \quad \forall i \in \mathcal{C} \\ x_{ijt} &\le D_{ijt} \quad \forall i, j \in \mathcal{C} \\ x_{ijt}, y_{ijt} &\ge 0 \quad \forall i, j \in \mathcal{C} \end{aligned} \qquad (4)$$

In this model, we assume that T is chosen to be a *near forecast horizon* (ARONSON and CHEN, 1985), meaning that it is long enough to suggest that the optimal first period provide good results when implementing solutions (x_0, y_0) on a rolling horizon basis.

An alternative to the simultaneous form of the model is the recursive form. First, for notational simplicity, assume all travel times are one period. That is, $\tau_{ij} = 1$, $i, j \in \mathcal{C}$. Now let:

S_{it} = total flow moving *through* region i at time t

$$= \sum_{k \in \mathcal{C}} (x_{ki,t-1} + y_{ki,t-1}) \qquad (5)$$

Next, we define recursively, for $t = 1, \ldots, T$:

$$Q_t(S_t) = \max_{x_t, y_t, S_{t+1}} r_t x_t - c_t y_t + Q_{t+1}(S_{t+1}) \qquad (6)$$

subject to:

$$\sum_j (x_{ijt} + y_{ijt}) = S_{it} \quad i \in \mathcal{C}$$

$$\sum_i (x_{ijt} + y_{ijt}) - S_{t+1} = 0 \quad i \in \mathcal{C}$$

$$x_{ijt} \le D_{ijt} \quad i, j \in \mathcal{C}$$

$$x_{ijt}, y_{ijt} \ge 0 \quad i, j \in \mathcal{C}$$

The recursive form of the basic model is important because it sets the foundation for the stochastic model presented later. POWELL ET AL. (1993) shows that for nonlinear cost functions, the recursive forms allow the development of much faster algorithms. Below, the recursive form is necessary to handle uncertainties in forecasted demands. For the basic linear model, however, the only methods that have been specialized for linear, dynamic networks are the inductive algorithms presented in WHITE and BOMBERAULT (1969) and ARONSON and CHEN (1986), which start with a one period problem and solve progressively longer horizons.

Specific advantages of this model over the assignment model include:

1. It captures both actual and forecasted demands over the planning horizon.
2. The model can recommend repositioning excess capacity from one region to the next.
3. The model can recommend which loads to accept or reject when there is too little capacity.

Disadvantages of the model include:

1. Deterministic loads which have already been booked are not differentiated from forecasted loads. The model can recommend moving a load that has been forecasted but has not really materialized.
2. Truncation errors can be significant in a deterministic model, forcing the use of a long planning horizon. The result can be a surprisingly large dynamic network.
3. Forecasted demands are generally noninteger, producing fractional solutions. Heuristic round-

qing of the demands (or the optimal fractional flows) can introduce significant errors.

4. Spatial aggregation into regions simplifies demand forecasting, but eliminates the ability to recognize specific characteristics of drivers and loads.

4. A STOCHASTIC, DYNAMIC MODEL

SOME OF THE WEAKNESSES of the deterministic formulation of the dynamic vehicle allocation problem can be corrected by introducing a stochastic formulation that explicitly handles uncertainties in demand forecasts. Randomness in forecasted demands can be formulated using a dynamic network with random arc capacities. This model was first proposed by POWELL (1988). We review this basic model, starting with the simpler formulation which assumes that travel times between each pair of cities is exactly one time period. This assumption is also used in POWELL (1988), as well as in FRANTZESKAKIS and POWELL (1990), which proposes a method for solving the stochastic program. These earlier papers also assume that all demands in the first period are known, and that all demands in future time periods are forecasted. In contrast, we suggest a method that not only handles multiple period travel times, but also a mixture of known and forecasted demands in future time periods. The distinction between known and forecasted loads (in the future) is important for several reasons. First, deterministic loads in the future can be handled more accurately than stochastic loads (if we lump deterministic loads to be served in the future with stochastic loads, our solution quality will not be as high). Second, we wish to investigate problems where we vary the percentage of demands that is known in the future. If we do not handle correctly the optimization of loads in the future, we will underestimate the value of future information.

4.1. Single Time Period Travel Times

To begin, we assume that we have a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with elementary outcome $\omega = \{\omega_1, \omega_2, \ldots, \omega_T\}$. An outcome ω_t represents a realization of all market demands D_t in time period t. We assume that decisions (x_t, y_t) are made in time period t after the realization of demands in period t, and before the realization of demands in later time periods. In this case, we refer to time period t as a *stage*. The *history* of the process, $\mathcal{H}_t$, is given by

$$\mathcal{H}_t = \{(x_0, y_0), (\omega_1, x_1, y_1), (\omega_2, x_2, y_2), \ldots, (\omega_{t-1}, x_{t-1}, y_{t-1})\}$$

The history of the system, for our purposes, can be summarized by a state variable $\mathbf{S}_t$.

Let

D_{ijt} = random market demand from i to j in period t.

$\hat{D}_{ij0}$ = actual (known) market demand from i to j in period 0.

x_{ij0} = flow of loaded vehicles in stage 0,

y_{ij0} = flow of empty vehicles in stage 0,

$\mathbf{S}_t$ = $\mathbf{S}_t(\mathcal{H}_t)$ = state of the system at the beginning of stage t, which depends on the history of the process up to stage t,

$x_t(\mathbf{S}_t, \omega_t)$, $y_t(\mathbf{S}_t, \omega_t)$ = flows of (loaded, empty) vehicles in stage t, which depends on both the state of the system $\mathbf{S}_t$ and the random outcomes in stage t, ω_t.

It is important in our notation to express the fact that our decision variables are conditionally dependent on both the state of the system at the beginning of the stage (or alternatively, the history of the process up to that stage) as well as random outcomes in that stage.

Under the assumption that $\tau_{ij} = 1$, $\mathbf{S}_t$ is defined by equation (5). For the case with unit travel times, the basic stochastic model for dynamic fleet management can be written as the following multistage stochastic program:

$$\max_{x_0, y_0, S_1} \{r_0 x_0 - c_0 y_0 + E_{\omega_1}[Q_1(\mathbf{S}_1, \omega_1)]\} \tag{7}$$

subject to:

$$\sum_j (x_{ij0} + y_{ij0}) = R_{i0} \quad \forall i \in \mathcal{C} \tag{8}$$

$$x_{ijt} \leq \hat{D}_{ij0} \quad \forall i, j \in \mathcal{C} \tag{9}$$

$$x_{ij0}, y_{ij0} \geq 0 \quad \forall i, j \in \mathcal{C} \tag{10}$$

$$\sum_i (x_{ij0} + y_{ij0}) - S_{j1} = 0 \quad \forall i, j \in \mathcal{C} \tag{11}$$

$Q_1(\mathbf{S}_1, \omega_1)$ is the *conditional recourse function*, defined recursively using:

$$Q_t(\mathbf{S}_t, \omega_t) = \max\{r_t x_t(\mathbf{S}_t, \omega_t) - c_t y_t(\mathbf{S}_t, \omega_t) + E_{\omega_{t+1}}[Q_{t+1}(\mathbf{S}_{t+1}, \omega_{t+1})]\} \tag{12}$$

subject to:

$$\sum_j (x_{ijt}(\mathbf{S}_t, \omega_t) + y_{ijt}(\mathbf{S}_t, \omega_t)) = S_{it} \quad \forall i \in \mathcal{C}$$

$$\sum_i (x_{ijt}(\mathbf{S}_t, \omega_t) + y_{ijt}(\mathbf{S}_t, \omega_t)) - S_{j,t+1} = 0 \quad \forall j \in \mathcal{C} \tag{13}$$

$$x_{ijt}(\mathbf{S}_t, \omega_t) \leq D_{ijt}(\omega_t) \quad \forall i, j \in \mathcal{C}$$

$$x_{ijt}, y_{ijt} \geq 0 \quad \forall i, j \in \mathcal{C}$$

Let

$$\bar{Q}_t(\mathbf{S}_t) = E_{\omega_t}\{Q_t(\mathbf{S}_t, \omega_t)\}$$

be the *expected recourse function*. $\bar{Q}_t(\mathbf{S}_t)$ is a convex function of $\mathbf{S}_t$. If we could determine this function explicitly, we could substitute it back into equation (7) and determine the optimal first-period decisions.

4.2. Multiperiod Travel Times

The previous section benefited from the simplification afforded by the assumption that all travel times between cities require one period. In practice, of course, travel times between cities might take on integer values $\tau_{ij} = 1, 2, \ldots, \bar{\tau}$. One way to handle multiperiod travel times is to define an extended state variable:

$S_{jt}(k)$ = flow inbound to region j at time period k that will not actually arrive to region j until time period $t + k$

$N_j(k) = \{i | \tau_{ij} = k\}$

Now our state variable is augmented to include not only the inventory of vehicles in each city, but also the number that are inbound but k periods away. Now the equations for single period travel times become:

$$\max_{x_0, y_0, S_1}\{r_0 x_0 - c_0 y_0 + E_{\omega_1}[Q_1(\mathbf{S}_1, \omega_1)]\} \quad (14)$$

subject to:

$$\begin{aligned} \sum_j (x_{ij0} + y_{ij0}) &= R_{i0} \quad \forall i \in \mathcal{C} \\ x_{ijt} &\le \hat{D}_{ij0} \quad \forall i, j \in \mathcal{C} \\ x_{ij0}, y_{ij0} &\ge 0 \quad \forall i, j \in \mathcal{C} \\ \sum_{i \in N_j(k)} (x_{ij0} + y_{ij0}) + S_{j0}(k) - S_{j1}(k-1) &= 0 \quad \forall j \in \mathcal{C} \end{aligned} \quad (15)$$

The conditional recourse function is defined similarly:

$$\begin{aligned} Q_t(\mathbf{S}_t, \omega_t) = \max\{&r_t x_t(\mathbf{S}_t, \omega_t) - c_t y_t(\mathbf{S}_t, \omega_t) \\ &+ E_{\omega_{t+1}}[Q_{t+1}(\mathbf{S}_{t+1}, \omega_{t+1})]\} \end{aligned}$$

subject to:

$$\sum_j x_{ijt}(\mathbf{S}_t, \omega_t) + y_{ijt}(\mathbf{S}_t, \omega_t) = S_{it}(0) \quad \forall i \in \mathcal{C}$$

$$\sum_i (x_{ijt}(\mathbf{S}_t, \omega_t) + y_{ijt}(\mathbf{S}_t, \omega_t)) + S_{j,t}(k) - S_{j,t+1}(k-1) = 0 \quad \forall j \in \mathcal{C} \quad (16)$$

$$x_{ijt}(\mathbf{S}_t, \omega_t) \le D_{ijt}(\omega_t) \quad \forall i, j \in \mathcal{C}$$

$$x_{ijt}, y_{it} \ge 0 \quad \forall i, j \in \mathcal{C}$$

This rather straightforward extension to multiperiod travel times is technically correct, but as we show below, causes us problems in the development of tractable approximations.

5. APPROXIMATING THE RECOURSE FUNCTION

STOCHASTIC PROGRAMS, and especially multistage stochastic programs, are in practice computationally intractable, requiring instead the development of approximations. In this section, we review methods for approximating the expected recourse function, focusing on techniques for replacing the recourse function with a nonlinear (or piecewise linear), separable approximation. We begin with scenario aggregation, which is a general purpose method widely used in the field of stochastic programming. Against this "brute force" approach are a number of other methods that do more to take advantage of the specific structure of our problem.

Scenario Aggregation

Scenario aggregation involves solving the so-called *equivalent deterministic* function (WETS (1974)). Assume that from our sample space that we choose L *scenarios* denoted by $(\omega^1, \omega^2, \ldots, \omega^s, \ldots, \omega^L)$, and assume that p^s is the probability of scenario ω^s. This involves solving:

$$\max \sum_{s=1}^{L} p^s \left\{ \sum_{t=0}^{T} r_t x_t(\tilde{\omega}^s) - c_t y_t(\tilde{\omega}^s) \right\} \quad (17)$$

subject to, for $t = 1, 2, \ldots, T$:

$$\sum_{j \in \mathcal{C}} (x_{ij1}(\tilde{\omega}) + y_{ij1}(\tilde{\omega})) = S_{i1} + R_{i1} \quad \forall i \in \mathcal{C} \quad (18)$$

$$\sum_{j \in \mathcal{C}} [x_{ijt}(\tilde{\omega}) + y_{ijt}(\tilde{\omega})] - \sum_{k \in \mathcal{C}} (x_{k_{i,t-1}}(\tilde{\omega}) + y_{ki,t-1}(\tilde{\omega})) = R_{it} \quad \forall i \in \mathcal{C} \quad (19)$$

$$x_{ijt}(\tilde{\omega}) \le D_{ijt}(\tilde{\omega}) \quad \forall i, j \in \mathcal{C} \quad (20)$$

$$x_{ij0}(\omega^s) - x_{ij0} = 0 \quad s = 1, \ldots, L, \forall i, j \in \mathcal{C} \quad (21)$$

$$y_{ij0}(\omega^s) - y_{ij0} = 0 \quad s = 1, \ldots, L, \forall i, j \in \mathcal{C} \quad (22)$$

Equations (21) and (22) are called *nonanticipativity* constraints. Without them, the problem would decompose by scenario $\tilde{\omega}$, which means that we would

have more than one answer to implement in the first stage.

This formulation works for two-stage problems only. It is possible to develop a multistage formulation by defining a scenario $\tilde{\omega}_t$ for each stage t. Then, we would require nonanticipativity constraints for each stage. Typically, the resulting linear program explodes in size. More common in practice is to simply approximate all future stages as being a single stage. In effect, we will allow decisions in period t to "anticipate" the outcomes in period $t + 1$, for periods $t \geq 1$. However, even this approximate formulation can produce intractably large linear programs. For example, assume we have a stochastic network with 1,000 nodes and 10,000 arcs, of which 1,000 have random upper bounds. If we model this problem with, say, 50 scenarios, it becomes a linear program with 600,000 constraints (one for each node and capacitated arc, times the number of scenarios, plus a nonanticipativity constraint for each arc and each scenario), and 500,000 variables (one for each arc and each scenario).

The advantage of the equivalent deterministic formulation, which is a standard approach used within the stochastic programming community, is that it allows for the random generation of correlated demands, and explicitly represents the full optimization problem under each scenario. Thus, the approach allows for the simultaneous optimization of current and future actions. The major limitation is that this approach typically produces a very large scale linear program, forcing us to use a relatively small number of scenarios to capture future outcomes. In addition, the method does little to take advantage of the underlying structure of most problems that arise in transportation.

Stochastic Gradient Methods

Stochastic gradient methods (generally referred to as stochastic quasigradient, or SQG, methods) replace the expected recourse function with sample gradients. At each iteration, we would solve problems of the form:

$$\max_{x_0, y_0, S_1} r_0 x_0 - c_0 y_0 + (\bar{g}^k)^T S_1$$

subject to equations (8)–(11). $\bar{g}^k$ is an estimate of the gradient of the recourse function calculated by:

$$\bar{g}^k = (1 - \gamma)\bar{g}^{k-1} + \gamma g(\omega^k)$$

where γ is a smoothing constant and $g(\omega^k)$ is a subgradient of $Q(S, \omega^k)$ defined in equation (12). For our problem, $Q(S, \omega^k)$ is a pure network, so $g(\omega^k)$ is typically just the dual of the flow conservation constraints (11).

Stochastic quasigradient methods were developed by ERMOLIEV (1983) using a projection operator to handle constrained problems. GUPAL and BAJENOV (1972) demonstrate convergence for the constrained stochastic linearization, which requires the smoothing step for the gradient estimate (that is, it requires $\gamma < 1$).

Response Surface Methods

BEALE ET AL. (1980) suggest approximating expected recourse functions using response surface methods. For example, we might replace $\bar{Q}(S)$ with:

$$\hat{Q}(S) = a_0 + \sum_{i \in \mathcal{C}} (a_{i1} S_i + a_{i2} S_i^2)$$

Of course, we could include cross products and higher order terms, but the basic idea is to replace the recourse function with a statistically estimated function. If we choose to use a separable function, then the first stage problem can be solved as a nonlinear network. If the dimensionality of $\mathbf{S}$ is high (which is typical) then a large number of samples may be required to adequately estimate the coefficients.

Simple Recourse

The complexity of problem (10) is the demand constraint (13) with the random right hand side. Assume now that we replace this constraint with the following:

$$x_{ijt} + x_{ijt}^+(\omega) - x_{ijt}^-(\omega) \leq D_{ijt}(\omega)$$

where $x^+(\omega)$ and $x^-(\omega)$ are recourse variables. We assume that the recourse variables are chosen after a realization $D(\omega)$ is made. If $x_{ijt} > D_{ijt}(\omega)$, then $x_{ijt}^-(\omega) = x_{ijt} - D_{ijt}(\omega)$ can be interpreted as trucks moving empty from i to j. If $x_{ijt} > D_{ijt}(\omega)$, then $x_{ijt}^+(\omega) = D_{ijt}(\omega) - x_{ijt}$ is the lost demand. The key is that x is chosen before we know ω while the recourse variables $x^+(\omega)$ and $x^-(\omega)$ are chosen after we know ω.

Depending on the distribution of $D(\omega)$, this change converts the original stochastic program to a nonlinear network problem (see POWELL (1988) and POWELL and FRANTZESKAKIS (1992)), which can be solved using standard algorithms. The limitation, of course, is that the approximation of the expected recourse function can be quite crude.

Successive Linear Approximation Procedure (SLAP)

The SLAP algorithm (FRANTZESKAKIS and POWELL, 1990) is designed for multistage, dynamic networks with random arc capacities, where each stage is comprised of a bipartite graph. Assume the re-

course function for stage $t + 1$ is replaced with a linear approximation:

$$Q^L_{t+1}(S_{t+1}) = \sum_{i \in \mathscr{C}} v_{i,t+1} S_{i,t+1}$$

where v_t is an approximation of $\nabla Q_t(S_t)$. Then the recourse problem for stage t decomposes by region:

$$\hat{Q}_t(S_t) = \sum_{i \in \mathscr{C}} \hat{Q}_{it}(S_{it})$$

where

$$\hat{Q}_{it}(S_{it}) = E_{\omega_t} \max_{x_t, S_{t+1}} \sum_{j \in \mathscr{C}} (r_{ij} x_{ijt} - c_{ij} y_{ijt} + v_{j,t+1} S_{j,t+1}) \tag{23}$$

subject to:

$$\sum_{i \in \mathscr{C}} (x_{ijt} + y_{ijt}) - S_{j,t+1} = 0 \quad j \in \mathscr{C},\ t = 1, \ldots, T \tag{24}$$

$$x_{ijt} < D_{ijt}(\omega_t) \quad i, j \in \mathscr{C},\ t = 1, \ldots, T$$

The optimal solution of equation (23) can be found by letting w_{in} be the value of the n_{th} *option* out of region i. An option can be moving loaded or empty to region j, with value $r_{ij} + v_{j,t+1}$ or $-c_{ij} + v_{j,t+1}$, respectively. If the options are ranked so that $w_{i1} \geq w_{i2} \geq \ldots \geq w_{iN}$, then the optimal solution is to allocate as much flow to the highest ranked options until all S_{it} units of flow have been assigned. POWELL (1987) (see also FRANTZESKAKIS and POWELL, 1990) shows how the expectation in (23) can be easily found, producing a piecewise linear, convex function of the form:

$$\hat{Q}_{it}(s) = \sum_{k=1}^{s} q_{it}(k) \tag{25}$$

where $q_{it}(k)$ is the expected value of the k^{th} unit of flow in region i at time t.

Successive Convex Approximation Procedure (SCAM)

CHEUNG and POWELL (1995) introduce an approximation for the expected recourse function that takes the same form as equation (25) but which does not require the linearization approximation given in equation (5). Instead, $Q_{t+1}(S_{t+1})$ is replaced with a piecewise linear, convex (but separable) approximation. Using results from POWELL and CHEUNG (1994a,b)), a separable, piecewise linear approximation of $Q_t(S_t)$ is developed that is more accurate that than produced by the SLAP procedure. The reader is referred to these papers for details.

Hybrid Stochastic Gradient Methods

Finally, it is possible to combine stochastic linearization methods with a nonlinear approximation. CULIOLI and COHEN (1990) suggest solving problems which at iteration k would look like:

$$\min_x c^T x + K(x) + (a_k g^k - \nabla K(x^k))^T x$$

where x is our vector of first-stage decision variables, a_k is a stepsize (such as $a_k = 1/k$), g^k is a sample gradient at iteration k, and $K(x)$ is a strictly convex function (for example, we might use $K(x) = \|x\|^2$). The function $K(x)$ helps stabilize the process from one iteration to the next, thereby correcting a major deficiency of pure stochastic linearization.

Building on this idea, CHEUNG and POWELL (1994) propose solving problems of the form:

$$\min_{x_0, S_1} c^T x_0 + \hat{Q}^k(S_1) + a_k(g^k - \hat{q}^k)^T S_1$$

where:

g^k = a stochastic subgradient of $\bar{Q}(S)$ at S^k that is, $g^k \in \partial Q(S^k, \omega^{k+1})$;
$\hat{Q}^k(S)$ = a convex approximation of $\bar{Q}(S)$;
$\hat{q}^k$ = a subgradient of $\hat{Q}^k(S_1^k)$, that is $\hat{q}^k \in \partial \hat{Q}^k(S_1^k)$.

The function $\hat{Q}^k(S_1^k)$ is updated iteratively using:

$$\hat{Q}^{k+1}(S_1) = \hat{Q}^k(S_1) + a_k(g^k - \hat{q}^k)^T S_1;$$

This method assumes that we start with an initial approximation $\hat{Q}^0(S)$. This approximation could be as simple as $\hat{Q}^0(S) = |S|^2$, or it could be produced using one of the other methods described above. It is shown in CHEUNG and POWELL (1994) that the sample information can greatly improve the quality of the solution in a two-stage problem. Furthermore, since we are basically adding a linear correction term to the original approximation, we retain properties of separability if these exist in the original approximation $\hat{Q}^0(S)$.

Remarks

Each of these methods offers specific strengths and weaknesses, and testing each one is a project in itself. Scenario aggregation provides a mechanism for capturing complex correlations in the data. This is usually done by sampling realizations from past history. However, the resulting model is typically a very large linear program, which becomes intractably large if we try to capture multistage effects. Integer solutions are extremely difficult to obtain.

At the other extreme are stochastic gradient methods which replace the recourse function with a linear approximation. While we are not aware of

results for solving multistage problems, these methods have rigorous proofs of convergence for two stage applications (by contrast, scenario methods provide an optimal solution only for the specific scenarios represented, not the entire sample space). Stochastic gradient methods are also very easy to solve, and produce integer solutions naturally. However, they are known to have notoriously slow rates of convergence.

Response surface methods have a certain appeal, combining the flexibility of sampling with nonlinear approximations, but we are not aware of any testing or refinement of these methods for this problem class. It is an approach worth pursuing, but out of the scope of this paper.

Simple recourse provides simple nonlinear approximations which are easy to solve, but the approximations are very poor in quality, and the resulting models do not easily yield integer solutions.

The SLAP and SCAM approximations, which are more sophisticated versions of the approximation first introduced in POWELL (1987), both represent attempts to replace the expected recourse function with a separable, piecewise linear approximation. Current evidence suggests that the SCAM approximation will outperform SLAP (which in turn, should outperform the approximation in POWELL (1987)), but this is purely an experimental question. The primary limitation of all these approximations is that they are not convergent—they are purely approximations.

The hybrid method described in CHEUNG and POWELL (1994) offers the promise of combining a good approximation with an adjustment term that provides a rigorous proof of convergence. The attractiveness of this approach is that it achieves this goal without destroying the underlying network structure of the problem. The result certainly has theoretical appeal, but it remains to be seen whether it contributes substantially to the accuracy of the approximations.

6. A PURE NETWORK APPROXIMATION

IF WE CHOOSE to approximate the expected recourse function using the equivalent deterministic framework, we face the daunting task of solving a potentially very large linear program. A more attractive alternative is to use one of the separable (convex) approximations. Starting with the version of the problem with single period travel times, equation (7) would become:

$$\max_{x_0,y_0,S_1}\left\{ r_0x_0 - c_0y_0 + \sum_{j\in\mathcal{C}} \hat{Q}_{j1}(S_{j1}) \right\}$$

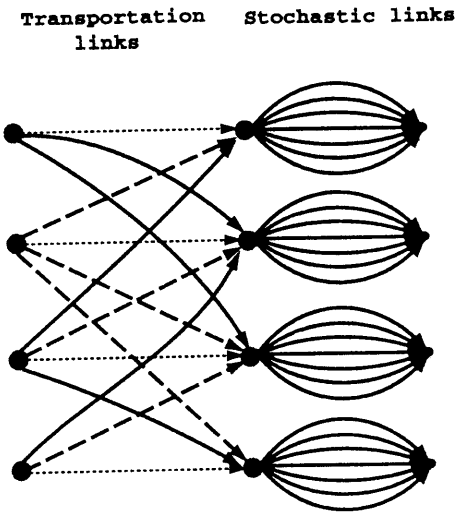

Fig. 4. Equivalent network formulation of stochastic program with single-period travel times.

If the random variables in the expected recourse function are discrete, then $\hat{Q}(\mathbf{S}_t)$ should be a piecewise linear, convex function. In this case, (26) is equivalent to the network shown in Figure 4, where the approximate recourse function is represented by a cluster of "recourse links" which capture the expected marginal contribution of each unit of flow into a region in a time period. Known demands (in the first time period) are represented as links moving between the origin and destination of the demand, with a positive contribution and an upper bound of one (or equal to the number of loads moving between the same origin and destination).

In addition to assuming one period travel times, this formulation also assumes that there is no "known" information in the future. In practice, real problems not only have multiperiod travel times, but a portion of the demands that have to be satisfied in the future will be known now. Of course, we can simply lump these into the forecasted demands, but in doing so, we lose the opportunity to work with this additional information.

If travel times span multiple time periods, then we would solve:

$$\max_{x_0,y_0,S_1}\left\{ r_0x_0 - c_0y_0 + \sum_{t=1}^{\bar{\tau}} \sum_{j\in\mathcal{C}} \hat{Q}_{jt}(S'_{j1}) \right\} \qquad (26)$$

(27) can be formulated as a pure network as illustrated in Figure 5.

This section has outlined an approach for solving the first stage problem of managing vehicle inventories by approximating the expected recourse function as a separable, piecewise linear function. The network model shown in Figure 5 was first intro-

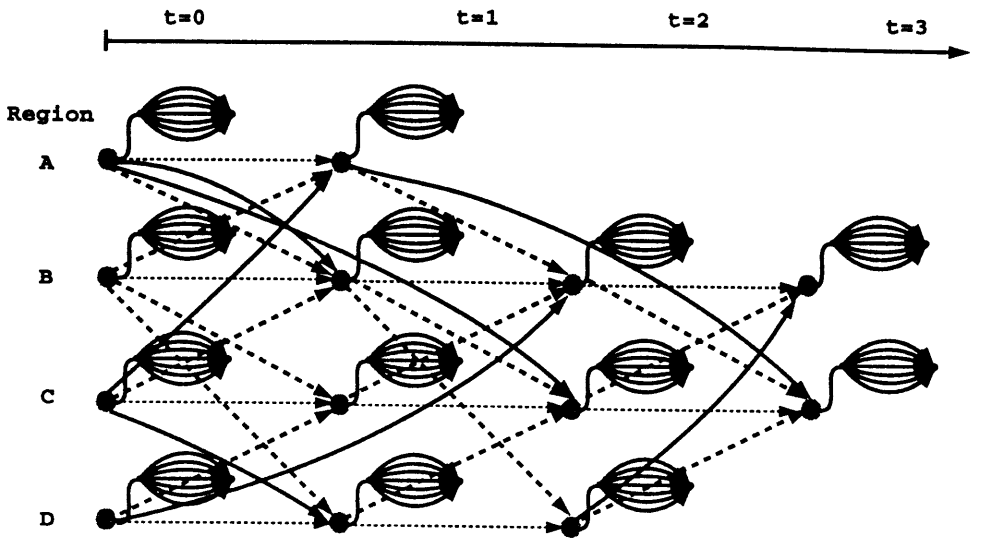

Fig. 5. Equivalent network formulation of stochastic program with multi-period travel times.

duced in POWELL (1987), and was the basis for the LOADMAP model first implemented at North American Van Lines (POWELL ET AL., 1988).

7. A HYBRID MODEL FOR THE DYNAMIC ASSIGNMENT PROBLEM

WE HAVE NOW reviewed two broad classes of models for the real-time dynamic problem. The first uses a static assignment model that assigns specific drivers to specific loads, while the second uses a dynamic network that accounts for forecasted demands, but requires temporal and spatial aggregation to handle forecasting of future activities. We have also described both deterministic and stochastic versions of the dynamic network. Clearly, each model offers both valuable strengths as well as significant weaknesses.

It is possible to produce a model that combines the best features of both models, illustrated in Figure 6. The network has two components. The first is the *assignment network*, which includes nodes for each individual driver and load, and arcs representing the assignment of drivers to loads. The second part is the *forecast network*, which works at an aggregate level and includes all forecasted demands as well as loads that have already been called in but are not to be picked up until some time into the future. Loads which are available to be picked up immediately or in the near future are represented by an origin (pick-up) node in the assignment network, and a destination node in the forecast network. The only other set of links that move between the assignment network and the forecast network are empty repositioning arcs which join a driver to a specific region and time period.

We assume for the modeling of the forecast network that time is aggregated into specific time periods. In truckload trucking, a typical time period is one day, since most loads must be picked up on a specific day (with varying time windows within the day). The model, however, may be solved in real-time over the course of a day as new information on drivers and loads becomes available. For this reason, the first time period in the forecast network (which we refer to as time period 0) specifically covers demands which are forecasted to be called in during the remainder of the day, while all future time periods refer to complete days (of course, time periods need not be of a particular size—time period 3 could cover two days if we wish). For example, when the model is run at 10 am, the first time period will include demands that are forecasted to be called in for pickup later that day. These forecasts will have to be updated periodically over the course of the day.

Nodes in the assignment network (drivers and loads) do not necessarily represent activities in the first time period. Rather, we associate with each node a time of availability. For example, driver $\mathcal{D}_i$ might be enroute to Chicago, and is forecasted to become available at hour 14 (later in the same day). Refer to this time as (0, 14). We could still represent this driver as a node in the assignment network, but we would have to restrict the links that can be generated out of this node. For example, this driver could only be assigned to loads that could be picked up after time (0, 14), accounting for driving time from the destination of the load to the origin of the next load.

Now consider a load that is available at time (1,

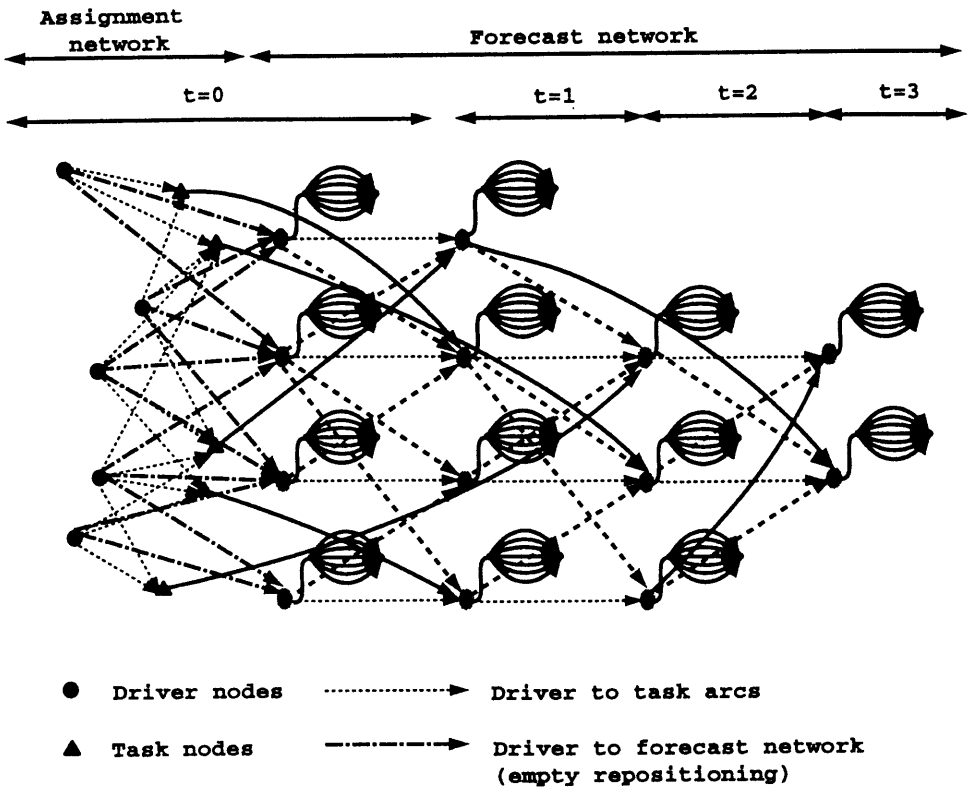

Fig. 6. Hybrid network for solving combined assignment and fleet management problem.

15) (that is, hour 15 on day 1) in region i. We can represent the origin of this load as a node in the assignment network, or as a link emanating from the node for region i on day 1 in the forecast network. The value of modeling the load as a node in the assignment network is that it allows us to capture all the characteristics of the load. By contrast, if we model the load as originating in the forecast network, then it is possible the load could be covered by a driver moving, say, from region k to region i on day 1, which could then be assigned to handle the second load. In this way, the model can assign one driver to handle two loads, although the unique characteristics of the driver and load are lost in the forecast network.

When a driver is assigned to an activity in the forecast network, there are essentially four possible options. The driver may be assigned to a node in the current time period or a future one. In addition, a driver may be assigned to a node representing the region where he is currently located, or a different region. The actions to be taken for each option are summarized as follows:

Assignment to first time period:

—Assignment to "own" region:

1. The truck should sit and do nothing. Its most profitable option is to wait for a load to be called in later in the day.

—Assignment to a different region:

2. The truck should reposition empty to the region in anticipation of demands that may be called in later in the day. If the truck sits where it is, it may be too far away from the load when it is called in, thereby resulting in lost revenue.

Assignment to a future time period:

—Assignment to "own" region:

3. Again, the truck should sit and do nothing, but if the model assigns him to a future time period, then it is unlikely that the driver will be used any time soon. This represents an opportunity for using telemarketing to perform load solicitation.

—Assignment to a different region:

4. The driver should reposition empty (typically moving overnight) to another region. Since the driver is not expected to be needed until some time in the future, it may not be necessary for the truck to start moving empty immediately.

The model can be stated mathematically by combining the assignment model (equation 1), with the approximate recourse function (equation 26):

$$(DAP)\ \min \sum_{d\in\mathcal{D}} \sum_{l\in\mathcal{L}} c^a_{dl}x^a_{dl} + \sum_{d\in\mathcal{A}} c^h_d x^h_d + \sum_{l\in\mathcal{L}} c^r_l y_l + \sum_{t=1}^{\tau} \sum_{j\in\mathcal{C}} \hat{Q}_{jt}(S_{jt}) \quad (27)$$

subject to:

$$\sum_{d\in\mathcal{D}} x^a_{dl} + y_l = 1 \quad (28)$$

$$\sum_{l\in\mathcal{L}} x^a_{dl} + x^h_d = 1 \quad (29)$$

$$x^a_{dl},\ y_l,\ x^h_d \geq 0 \quad (30)$$

$$S_{jt} = \sum_{i\in\mathcal{C}} [x_{ij,t-\tau_{ij}} + y_{ij,t-\tau_{ij}}] \quad (31)$$

Specific features of this hybrid model include:

- Drivers and loads can be modeled with the same level of detail possible in a static assignment model.
- The model can not only recommend which driver should be assigned to a particular load, but also whether a driver should be "held" in a region (in anticipation of future loads) or repositioned empty to a neighboring region (again, in anticipation of future loads).
- Known ("booked") loads and forecasted loads are each handled in a natural way. The model avoids the problem of deterministic models where known and forecasted loads are indistinguishable to the model.
- The model is a pure network, and can be optimized extremely quickly.
- The model returns integer solutions.

The integration of known and forecasted demands in a single model in this way is, we believe, new. Of particular value is the combination of the assignment network, which provides a high level of detail, and the forecast network, which handles forecasted activities at a lower level of detail.

Of course, the model also possesses limitations, including:

- The model can assign a driver to a single load, but cannot develop tours for specific drivers that cover multiple loads. The model can assign a driver to more than one load, but loses the identity of the driver past the first load.
- Forecasted loads require spatial and temporal aggregation, which can produce errors in decisions to reposition equipment.
- The separable approximation of the expected recourse function can ignore interactions between neighboring regions. In some cases, vehicles arriving to a region i_1 might be routinely repositioned empty to region i_2 which might be quite close by. The result can be significant distortions in the recourse functions for both regions.

At this point, we have a model with tremendous potential for solving the real-time driver assignment problem for truckload motor carriers. The question that arises is: What is the value of including an approximation of the expected recourse function, and how do errors introduced by approximating the recourse function impact the actual dispatch decisions? Lacking theoretical answers, we have to address this question experimentally.

8. EXPERIMENTAL TESTING

THE HYBRID NETWORK illustrated in Figure 6 represents an important extension of the standard assignment model that is most widely used in practice. It raises a series of experimental questions:

- Does the hybrid, stochastic, dynamic network model outperform a myopic, assignment model working alone?
- Can the resulting network model be optimized in real time for problems of realistic size?
- Does the model accurately capture the dynamics of real-time dispatching?

These questions address the validity of the model and whether it can be solved in real-time. The first question is of particular interest, since the standard approach in practice is to use a myopic assignment model to assign drivers to loads. The practical argument in favor of this simple model is that carriers do not need to reposition drivers empty in anticipation of future demands. Instead, they will simply wait until the customers call in, and then take advantage of the large size of the fleet to cover loads.

If the third question can be answered affirmatively, then not only have we developed the foundation for a successful real-time dispatching system, we also have a simulation model that can be used to address more substantive questions. For example:

- What is the cost of uncertainty in truckload dispatching?
- What are the economies of density that can be derived from larger operations?

A stochastic, dynamic model is particularly valuable in estimating the value of knowing more demands in advance. If we tried to answer this question with a simple myopic model, we could overestimate the value of reducing uncertainty simply because the myopic model might perform artificially badly in an environment where very little is known in advance (but where demands can be forecasted).

We investigate these questions using a simulation package called MIDAS (Micro Dispatch and Simulation). We begin with Section 8.1 by describing the process by which demands are generated. Next, Section 8.2 describes the specific optimization model used in MIDAS. Then, Section 8.3 describes the simulation model itself. Finally, Section 8.4 outlines the development of the datasets used for the simulations and summarizes initial testing that guided the running of the experiments.

8.1. The Booking Process

For this paper, we use a relatively simple model of the order booking process. We represent time by (d, h) where d represents a day, and h represents a time of day (for example, $0 \leq h \leq 23$). An order is characterized by call-in time, pickup window, origin and destination. If we let $\mathbf{d}_\ell$ represent the ℓ^{th} order, we can write:

$$\mathbf{d}_\ell = [(\mathbf{d}, \mathbf{h})_c, (\mathbf{d}, \mathbf{h}_1, \mathbf{h}_2)_p, \mathbf{i}, \mathbf{j}]_\ell$$

where:

$(d, h)_c$ = the day and hour that an order is first *called in*, which is when it becomes known to the carrier,

$(d, h_1, h_2)_p$ = the day, and time window within the day (represented by the interval (h_1, h_2)) during which the load must be picked up,

i, j = origin and destination locations of the load.

Let:

λ_s = aggregate rate at which demands are called in over the week over the entire system

$d_b = d_p - d_c$ = the number of days in advance that a load is "booked."

λ_s is assumed known. We further assume that calls arrive to the system according to a time-varying Poisson process. An order $\mathbf{d}_\ell$ is determined by the outcome of four independent random variables: the call-in time $(d, t)_c$, the difference between the pickup day and the call-in day $(d_p - d_c)$, the origin and the destination. From historical data, we assume that the following statistics can be generated:

$p_c(d_w)$ = fraction of demands called in on day of week d_w, $1 \leq d_w \leq 7$,

$p_h(h)$ = fraction of demands called in during hour h, $0 \leq h \leq 23$,

$p_b(d_b|d_w)$ = fraction of demands called in d_b days in advance ($0 \leq d_b \leq d_b^{\max}$), given that the order was originally called in on day of week d_w,

$p_o(i)$ = fraction of demands which originate in city i,

$p_d(j|i)$ = fraction of demands which terminate in city j given that they originated in i.

We make the assumption that if $d_b = 0$, then $(h_1, h_2) = (h_c, t_e)$, where t_e is defined as the "end" of the working day. In other words, if the call-in day is the same as the pickup day, then the pickup window spans from the time the order is called in until the end of the day (which we might define as 6 pm). On the other hand, if $d_b > 0$, then $(h_1, h_2) = (t_m, t_e)$, where t_m is the beginning of the working day (perhaps 7 am).

All of these statistics are easily obtained from historical data files. A key statistic is $p_b(d_b|d_c)$ which is referred to as the *booking profile*. This gives the amount of time which a demand is known in advance of the actual pickup. This distribution implies that a certain percentage of demands are already known by the beginning of the day. The process of randomly generating demands for the system is now fairly straightforward. Consider a particular point in time $t = (d, h)$. Let $d(t)$ be the day of week for time t, and let $h(t)$ be the hour of day for time t. At time t, calls are coming into the system at rate $\lambda(t) = \lambda_s p_c(d(t)) p_h(h(t))$. Using methods for simulating time-varying Poisson processes, we can easily obtain the call-in time for a demand. Once the call-in time is known, we can find the pickup day from the distribution $p_b(d_b|d_c = d(t))$. Finally, the origin of the demand can be determined from the distribution $p_o(i)$, and given the origin, we can sample a destination from $p_d(i|j)$.

This simple booking model captures a number of important qualities of an actual demand process. An important dimension of the process is the booking profile which tells us how much of the demand is known in advance. In addition, we capture time of day, and day of week effects, as well as a realistic spatial pattern. All of the parameters driving this model are easily estimated for a real company using commonly available historical data. At the same time, the model ignores other effects, some of which are easily incorporated, and others being more dif-

ficult. Among the easier fixes are using distributions that are dependent on the origin (time of day and day of week distributions may vary geographically) or even the destination of the load. The booking profile depends on the day of the week (this is particularly important for weekends) but not the time of day (the likelihood that a load should be picked up the next day increases with time of day). Also, the underlying model is Poisson, and does not account for more complex patterns of demands.

8.2. The Optimization Model

The optimization model used within MIDAS is the network model displayed in Figure 6, as originally described in CAPE (1987). The expected recourse function was calculated using the methods described in POWELL (1987). This is not the most advanced technology, as it ignores subsequent developments in FRANTZESKAKIS and POWELL (1990) and CHEUNG and POWELL (1995). The testing of different calculations of the expected recourse function is very difficult from a software development perspective, and the testing and comparison of alternative technologies for approximating the expected recourse function is beyond the scope of this paper.

The network model was generated from scratch at the beginning of the simulation, and at midnight for each subsequent day. This step is time consuming, but we took advantage of the low demand rate after midnight to perform the reoptimization (we did not "stop the clock" to regenerate and optimize the network). Once generated, the model was then updated and reoptimized over the course of the day, using the previous basis as a starting point. Once a driver is actually dispatched on a load, the model would estimate the next time of availability for the driver, and regenerate the driver node, and links to any load nodes (taking into account the time of availability of the driver, which might be several days in the future). Each time a new load was "called in" the model would generate a new load node in the assignment network, and a link from this node to the appropriate node in the forecast network. Then, we would generate links from nearby drivers into the load node.

The network was optimized with a primal network simplex code, developed by the author. A "big M" start was used for the initial optimization. Each time a driver was dispatched on a load, a high cost would be put on the corresponding assignment arc. The model would flag this arc to be dropped, and it would be eliminated from the data structures as soon as the system detected that the link was no longer in the basis (at this point, both the loaded movement arc, as well as all assignment arcs into the load node, would be dropped from the network).

8.3. The MIDAS Simulator

MIDAS is a system that simulates the real-time process of customers calling in orders (truckload movements) and the dispatching of drivers to those loads. The system uses as input a set of regions (used for the forecast network), forecasted region to region demands (by day of week), travel times and distances, hour of day booking profile, and the distribution of days between the time an order is called in, and the earliest time it can be picked up. The user fixes the number of drivers the system should use, and a preprocessor generates a file of initial driver positions (location, and the earliest time they will be available). Loads are also randomly generated in advance and stored in a file.

With these datasets in place, we are now ready to run a full simulation using MIDAS. MIDAS runs a simulation at a rate that is a specified factor times an actual clock. We did not, for example, allow MIDAS the time to optimize the problem with the clock turned off. This approach allowed us to determine whether the optimization could actually keep up with a problem. For example, assume that loads are called in at a rate of 600 per week. This rate is equivalent to an average hourly rate of 3.5 loads per hour. However, this could easily produce peak demand rates of up to 30 loads per hour. If we run the simulation at, say, 50 times normal speed, then we are testing the ability of the simulation to actually handle as much as 1500 loads per hour.

The running of the simulation was relatively simple. An internal loop cycled continuously, constantly checking the "real" time. An event list of drivers becoming available, and the call-in time of the next load, was maintained. Whenever the time of an event matched the real clock time, the system would call the optimization model to determine what action should be taken. As each action was taken, or as new events occurred (such as a new load being "called in") we would update the network model and reoptimize.

Whenever an action was taken, statistics would be computed and a running summary was presented to the user. In addition, a file giving cumulative statistics each hour was produced. Statistics recorded included: total profits, total empty miles, total loaded miles, total loads moved, and total refused loads.

We developed the model with a detailed graphical user interface which allowed us to see each individual dispatch. This display was valuable in the early stages of the development both for model debugging, as well as refining the model. For example, one

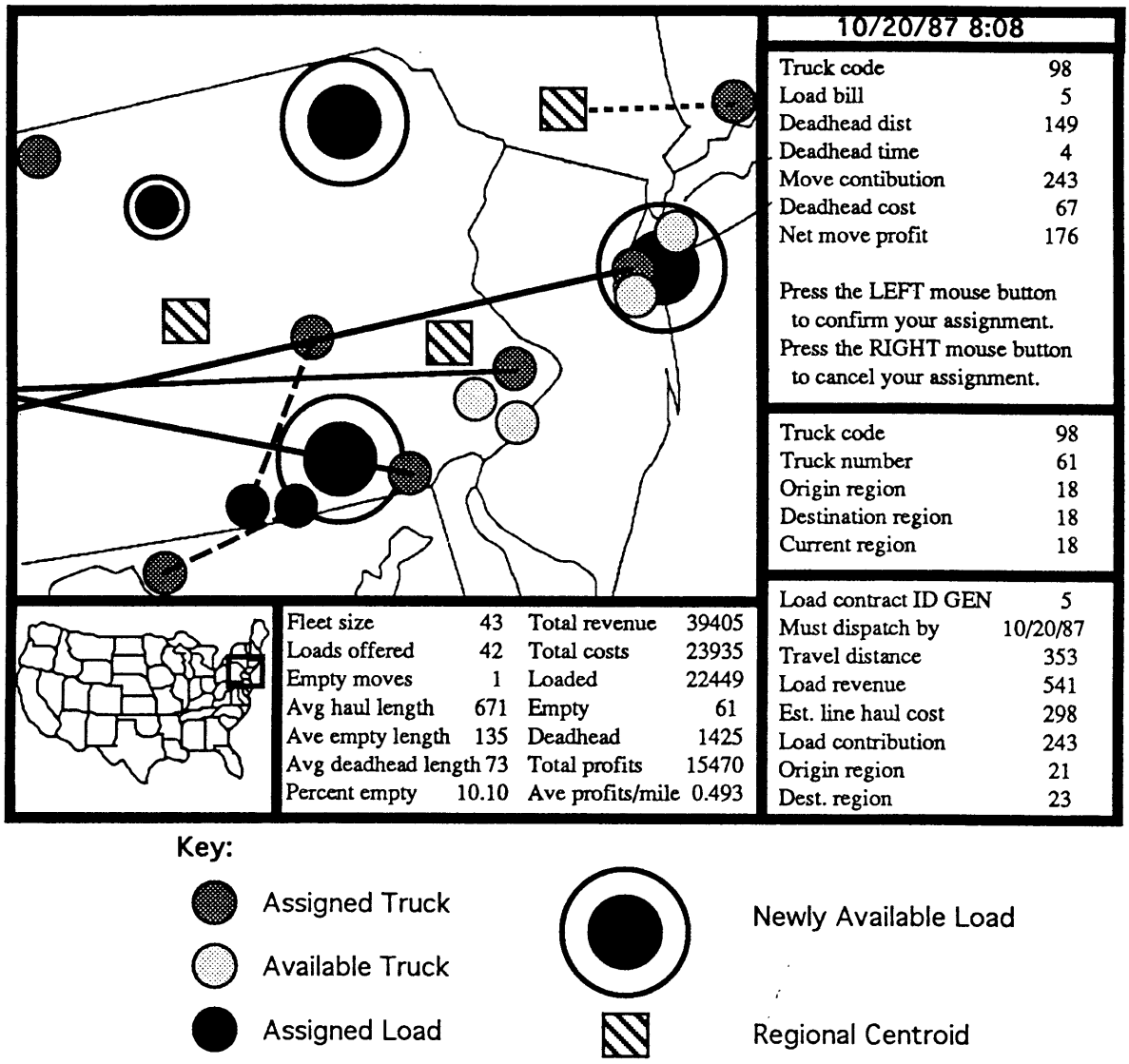

Fig. 7. Illustration of interactive screen for MIDAS simulator.

anomaly that was discovered through the graphical interface was the tendency of the model to move trucks empty back and forth between certain nearby regions, since this empty cost was less than the driver layover cost (the result was higher profits, but also higher empty miles).

The graphical interface is depicted in Figure 7. In the lower left hand corner is a map of the United States with a small rectangle. By adjusting the size and location of the rectangle, the user could control the contents of the main panel. In the main panel, the system would show loads waiting to be picked up, drivers waiting and unassigned, drivers moving loaded, drivers moving empty to pick up a load, and drivers moving empty as a repositioning move to a neighboring region. At the bottom of the screen is a system summary report with a variety of statistics that are updated as the simulation progresses. Finally, along the right hand column, three panels are provided which allow the user either to override the recommendation of the optimization model, or to perform all the optimizations manually. This feature allowed us to run simulations comparing the performance of the optimization model to that of manual dispatchers (students).

8.4. Experimental Design

The data for our runs was derived from a major truckload motor carrier. The carrier had a fleet of approximately 1,100 drivers serving 2,600 loads per week between 60 regions. Using an historical dataset that covered approximately one month of actual

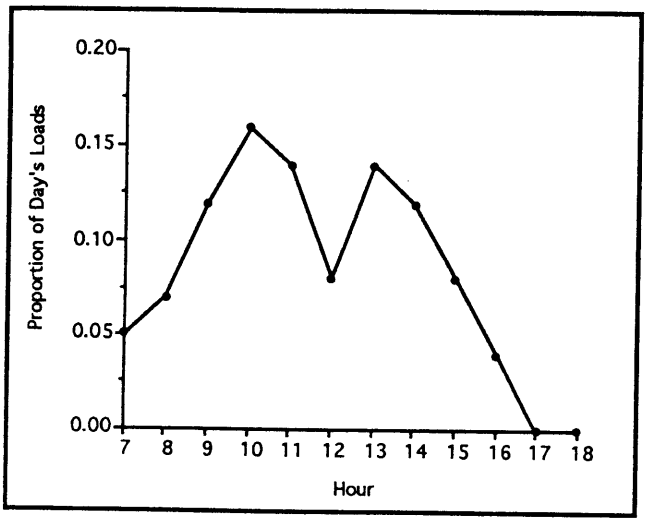

Fig. 8. Hour-of-day call-in distribution.

loads, we were able to construct the distribution of loads by origin; the conditional distribution of loads by destination; the distribution of loads called in by hour and by day of week; and the booking profile. The hour of day distribution is given in Figure 8.

The booking profile, which gives the distribution of loads called in 0, 1, 2, . . . , days in advance, was replaced with a simpler distribution for the purposes of parametric testing. The data showed that approximately 60 percent of the loads were called in for same day pickup, and roughly 40 percent were for pickup on the next working day. Therefore, we used as a booking profile the data given in Table I. In subsequent experiments, we replaced the percent of loads called in for pickup tomorrow, assumed in this table to be 0.4, with a parameter p_b which was allowed to vary from zero to one. If the call-in day was Friday, Saturday or Sunday, we assumed that the fraction called in for same-day pickup was $1 - p_b$, while the rest of the distribution was assumed to be proportional to the values in Table I.

Finally, Table II gives the distribution of demands called in by day of week. For this dataset, Mondays and Fridays were the heaviest days for loads called in. Of course, a proportion of the Friday loads were called in for Monday pickup, making Monday and Tuesday the heaviest days for picking up loads.

TABLE II

Day of Week Call-in Distribution

Day	Percent
Monday	21.2
Tuesday	18.7
Wednesday	18.5
Thursday	16.8
Friday	20.7
Saturday	3.2
Sunday	1.0

Using this data, we randomly generated a set of drivers and loads. We decided to scale the system to a 200 driver fleet. We chose a system demand rate equal to 600 loads per week. The ratio of three loads per driver per week is higher than is achieved in practice, where numbers in the range of 2 to 2.5 are more typical (the ratio depends on the average length of a load). Our choice of a higher load to driver ratio was intended to capture refused loads, which are never present in historical datasets which only show the loads that were actually carried.

A number of runs were conducted to determine model warm-up and to evaluate the statistical stability of key output estimates (such as operating contribution per mile). We initially expected that, given our care in generating realistic initial conditions, that we would not need to account for a warm-up period, and, given the number of events being simulated, that a one or two week simulation would suffice. However, one key aspect of the model we are testing is its ability to reposition empty equipment from regions with excess inbound demand to regions with excess outbound demand. Figure 9 shows the number of drivers available in each region at the end of each week, over a four week run. From this data, we concluded that the distribution of inventories had not stabilized even at the end of a

TABLE I

Day of Week Booking Profile

Call-in Day	Pickup Day						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Monday	0.6	0.4	0.0	0.0	0.0	0.0	0.0
Tuesday	0.0	0.6	0.4	0.0	0.0	0.0	0.0
Wednesday	0.0	0.0	0.6	0.4	0.0	0.0	0.0
Thursday	0.0	0.0	0.0	0.6	0.4	0.0	0.0
Friday	0.3	0.0	0.0	0.0	0.6	0.1	0.0
Saturday	0.3	0.0	0.0	0.0	0.0	0.6	0.1
Sunday	0.4	0.0	0.0	0.0	0.0	0.0	0.6

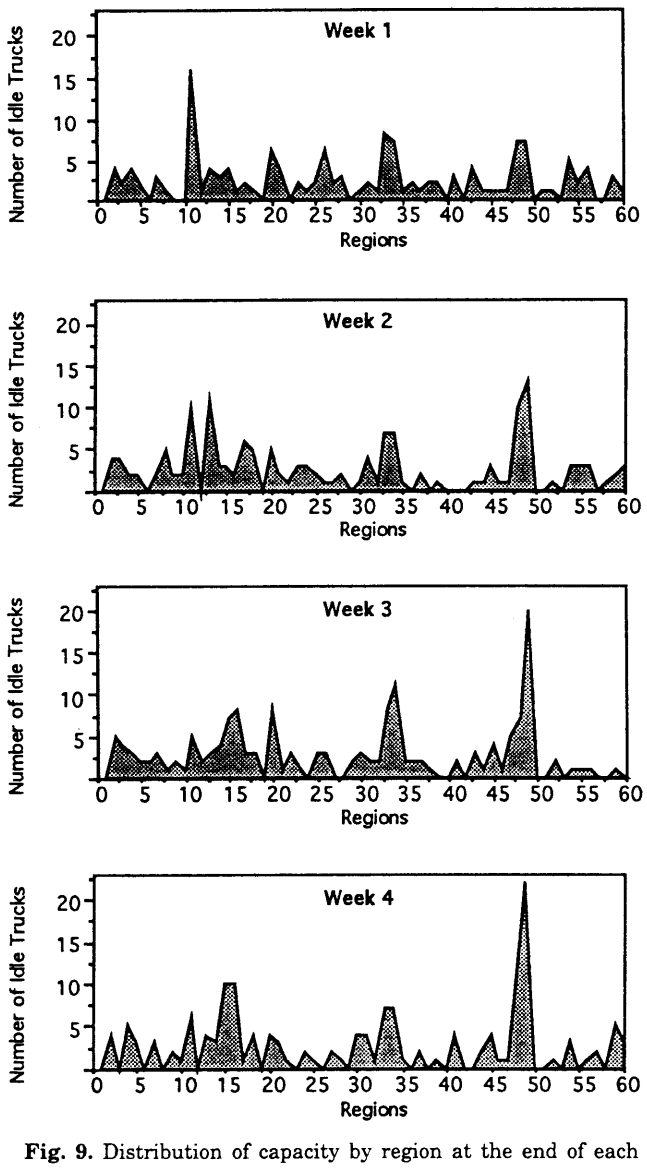

Fig. 9. Distribution of capacity by region at the end of each week in the simulation.

full week of simulated dispatching, despite the care given to generating reasonable initial distributions. However, weeks 2, 3 and 4 do seem fairly similar. For all subsequent analyses, we discarded statistics derived from the first week, and averaged the last three weeks.

We next conducted a series of runs to develop a sense of the statistical stability of a simulation run. Table III summarizes the results of four separate datasets, each using a different set of drivers and loads. For each run, we show the loads per driver per week, the contribution (in dollars) per driver per week, the percent of miles run empty, and the percent of loads that were "refused." This dataset is too small to provide accurate estimates of the distribution or even the standard deviation of these statistics, but it does provide an indication of the degree of variability in each statistic.

TABLE III
Results of Repeated Runs Showing Statistical Variability

Data Set	Loads/Driver/ Week	Contribution/ Truck/Week	Percent Empty Miles	Percent Loads Refused
P1	2.05	$176.3	8.57	31
P2	1.95	$178.3	8.63	35
P3	1.86	$163.8	8.89	38
P4	1.92	$157.4	8.68	36

These runs were all conducted using an internal simulation clock that ran 120 times real time. To further speed the simulations, we used a speedup factor of 400 between 7 pm and 7 am, when there was relatively little activity. At this speed, our 200 driver fleet with 600 loads per week had the effective size, in terms of algorithmic demands, of a fleet with 24,000 drivers and 72,000 loads per week. By contrast, the largest truckload fleets have approximately 8,000 drivers handling approximately 20,000 loads per week. However, we were not able to run a 1,000 driver fleet at this speed. (The runs were made on a Silicon Graphics IRIS workstation.) To keep execution times reasonable, we implemented a strategy where dispatching is performed once every 15 simulated minutes. Thus, new customer requests are batched up and dispatched once every 15 minutes (at a speedup of 120, this is equivalent to running the optimization model once every 7.5 seconds in real time). Comparisons with a pure continuous time application of the optimization model (where we optimize each time a customer calls in) showed that the results were virtually the same. From these results, we can already conclude that the model can handle problems well in excess of those encountered in practice.

9. EXPERIMENTAL RESULTS

THREE EXPERIMENTAL QUESTIONS were addressed. The first, and most important methodologically, is the value of a dynamic model over a more traditional myopic model. Section 9.1 shows that the dynamic model does in fact outperform a myopic model, and provides an estimate of the benefit. Then, Section 9.2 estimates the value of future information by simulating problems where loads are known farther in advance. Finally, Section 9.3 addresses the substantive question of the economies of density, and provides an estimate of the relative productivity of small versus large fleets.

9.1. The Value of a Dynamic Model

We begin our investigations by addressing the question of the value of the stochastic, dynamic model.

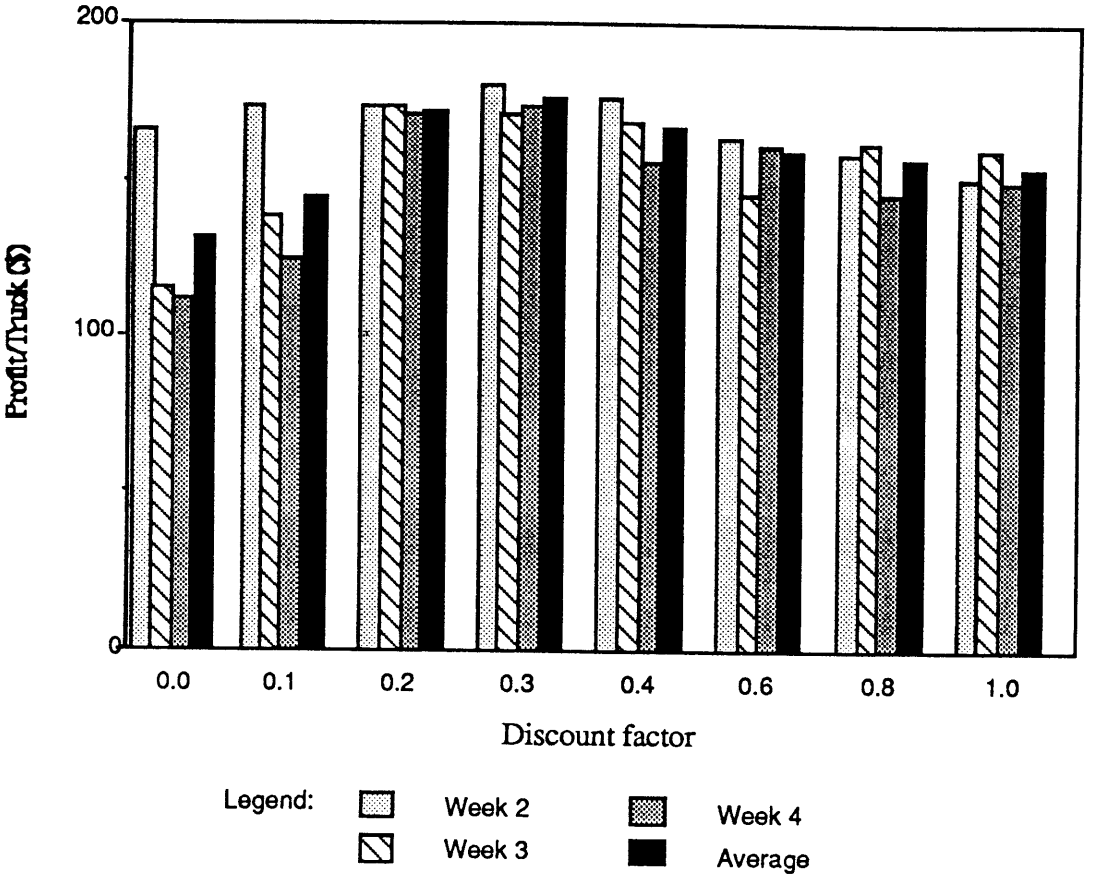

Fig. 10. Effect of discount factor on expected recourse function over the length of the simulation.

Other researchers have considered the value of a stochastic, dynamic model over a deterministic, dynamic model (FRANTZESKAKIS and POWELL (1990), CHEUNG and POWELL (1995)). This prior research was restricted to discrete time (using a unit of one day) and discrete space (where capacity in one region was not allowed to handle demands in another region in the same day). Here, we test the value of a stochastic, dynamic model in the context of a continuous time, continuous space problem. As described earlier, a discrete time, discrete space approximation was used for the forecast network. However, the assignment model observed no such boundaries. Furthermore, MIDAS simulates real-time dispatching, thereby testing the model in both a continuous time, continuous space setting.

We tested the value of the stochastic, dynamic model by introducing a "discount factor" α in front of the expected recourse function $\bar{Q}(S)$. If $\alpha = 0$, then the model is equivalent to a myopic assignment model, introduced in Section 2. If $\alpha = 1$, then we have the full hybrid model presented in Section 7. We also tested intermediate values of α.

The results of these experiments are shown in Figure 10. For each value of α from 0 to 1, we show the total system profits for each of weeks 2, 3 and 4, as well as the average over all three weeks. From this single diagram, we can make the following observations:

- A stochastic, dynamic model (with $\alpha > 0$) significantly outperforms a myopic assignment model when measured in terms of the contribution per truck.
- At the same time, overall performance is optimum with $\alpha = 0.3$, which outperforms the fully dynamic model ($\alpha = 1$).
- The performance of the system with $\alpha = 0$ is fairly good in week 2 of the simulation, but degrades substantially in weeks 3 and 4. Thus, the value of a dynamic model is noticeable only over a fairly long simulation period. If we excluded week 2, and treated weeks 3 and 4 as more representative of steady state, the value of the dynamic model would be even more pronounced. By contrast, week 2 is not that different from weeks 3 and 4 for values of α greater than 0.2.

TABLE IV

Effect of Advance Booking of Demands to Reduce Uncertainty

Proportion of Demands Known in Advance	Loads/ Truck/ Week	Contribution/ Truck/Week	Percentage of Miles Empty	Percentage of Loads Refused
0.0	1.97	$166.2	9.06	34
0.2	2.01	$172.9	8.94	33
0.4	1.97	$171.9	8.51	34
0.6	2.06	$175.6	9.08	31
0.8	1.94	$184.2	8.07	35
1.0	1.90	$189.7	7.63	36

TABLE V

Economies of Density for Truckload Motor Carriers

Fleet Size and Demand Rate	Loads/ Truck/ Week	Contribution/ Truck/Week	Percentage of Miles Empty	Percentage of Loads Refused
100T, 300L	1.41	$123.9	8.69	53
200T, 600L	1.97	$171.9	8.51	34
300T, 900L	2.13	$193.6	8.36	28
400T, 1200L	2.26	$213.4	7.75	25
1000T, 3000L	2.55	$280.2	5.51	17

9.2. The Value of Future Information

We next considered the value of future information. This question was studied by varying the prebook parameter in Table I. In this base table, the prebook parameter was set to $p_b = 0.4$. We varied this from 0.0 to 1.0, ranging from no advance information to the situation where all demands are known a day in advance. As we varied p_b, we handled the weekends by spreading the distribution of loads picked up on each day in the future by using the same proportions in Table I. For example, loads called in on Friday show 10 percent to be picked up on Saturday, and 30 percent to be picked up the following Monday. If we use $p_b = 0.2$ instead of 0.4, we would simply scale these percentages down by 50 percent.

The results of varying p_b from 0 to 1.0 are shown in Table IV. For each run, we used the exact same set of drivers and loads, but simply generated new pickup dates for each load. This approach minimized the statistical variability between the runs. The results show a steady improvement in profits as the proportion of advance information increases. Interestingly, there was not a significant change in the percent of loads refused (which actually seemed to increase slightly) but there was a marked decline in empty miles, and approximately a 15 percent improvement in overall profitability between the two extremes.

One application of this result is calculating the benefits of a price discount to encourage advance booking. The question is, are the savings of advance information large enough to warrant a price discount, which in itself would have to be large enough to encourage a change in behavior on the part of the shipper. From the results in Table IV, it seems unlikely that the economics are strong enough to warrant the use of significant price discounts to encourage shippers to call in their demands early. From a base of 40 percent advance booking, increasing the advance booking to 80 percent would improve average profitability of each driver by approximately 8 percent. This improvement would be offset by a decrease in price of approximately one percent. Thus, a price discount to encourage shippers to call in their orders one day in advance would have to be less than one percent to be attractive to the carrier.

9.3. Economies of Density

Our last set of experiments addressed the substantive question of economies of density in the motor carrier industry. We created five different datasets, with 100, 200, 300, 400 and 1,000 drivers, respectively. For each dataset, we held the ratio between the number of loads per week and the number of drivers to exactly three. Each simulation was run for four weeks.

The results are shown in Table V. First, it is useful to note that most carriers will average between 2.0 and 2.5 loads per driver per week. Also, an unoptimized carrier (that is, using manual dispatching) will exhibit empty mile rates of around 6 to 8 percent for the largest carriers, up to 10 to 12 percent for medium sized carriers (more specialized carriers might run as much as 20 percent empty). Considering that our model optimizes empty miles, without regard to other factors (such as getting drivers home) which would increase overall empty miles, the numbers in Table V seem quite realistic, with the exception of those for the 100 truck fleet. In practice, a 100 truck fleet (and even 200 trucks) would serve only a regional customer base, whereas our simulator generates demands nationally (heavily weighted around the midwest, east and south). Therefore, we should discount the statistics for our smallest fleet size.

The most important column is the contribution per truck per week, which shows substantial improvements in overall profitability from the increase in density. Typically, a truck will average around 2,000 miles per week. Comparing the 200 truck fleet to the 1,000 truck fleet shows over an improvement in overall contribution of $100 per week, which translates to improved profits of around $0.05 per mile. With transportation rates around $1 per mile, such an advantage would provide a carrier with a substantial cost advantage.

Today, there are perhaps approximately two dozen carriers with fleets of 1,000 drivers or more, with two megacarriers with fleets approaching 10,000 drivers. While we did not simulate fleets this size, it is likely that these largest carriers also enjoy a modest cost advantage of around $0.01 per mile due purely to economies of density. However, in an industry with tight profit margins, such a difference is indeed significant.

10. CONCLUSIONS

THIS PAPER HAS provided both an introduction to the load matching problem for truckload motor carriers, and an overview of a variety of different modeling approaches. The paper represents one of the earliest attempts to perform dynamic routing and scheduling of drivers using a model that takes explicit account of forecasted demands. The experimental results in this paper, on a single dataset derived from an actual carrier, show that the dynamic model does in fact outperform the more standard myopic one.

This paper is, at the same time, only a first, small step in what is likely to become an active field of research. While these initial results are encouraging, *many* questions remain. Methodologically, this paper has described several ways for approximating the expected recourse function, but only the most primitive was actually tested. Additional testing is needed to evaluate the newer approximations that have been developed (such as FRANTZESKAKIS and POWELL (1990) and CHEUNG and POWELL (1995)). We also need to refine and test the hybrid method suggested in CHEUNG and POWELL (1994). This method has tremendous conceptual appeal, but experimentation is needed for the complete evaluation.

An area that sparks considerable debate within the research community is the need for stochastic models. FRANTZESKAKIS and POWELL (1990) and CHEUNG and POWELL (1995) show that stochastic approximations will outperform deterministic models, but other researchers will no doubt argue that a better deterministic approximation might work as well, or better, than the stochastic model. Deterministic models do not need the separability approximation that we have used in the development of our stochastic models. A significant problem with our separable approximations is our need to treat a location i at time t independently of location i at time $t + 1$. Assuming separability across time limits our ability to use smaller time steps. Deterministic models do not have this problem. Only time will tell which approximation works out the best.

There is an open question of how all of these methods work with different data sets. Real problems vary in terms of the width of the time windows, which in turn impacts the need to anticipate future demands. If there is a single, important factor in the testing of stochastic models in a dynamic setting, it is the need to anticipate demands before they happen. Clearly, taxi drivers must anticipate their demands. Railroads, on the other hand, do not face the same service demands. Truckload carriers fall in between. As we tested our models, we found that consistently, the area where they performed poorly was their tendency to overanticipate demands. The introduction of the discount factor α represented a simple, heuristic mechanism to dampen the effect of the recourse function, which reduced some of the unnecessary empties.

Another question of this research is our ability to approximate the history of the process using a relatively simple state variable, giving the number of drivers available at each location at each point in time in the future. Thus, we care about how many drivers will be in Chicago on Wednesday, but we do not care how many of these drivers need to get home, how many have run out of hours, how many have tractors needing maintenance, and so on. In our simulations, we ignored the need to get drivers home, but did capture their duty hours. This research suggests that aggregate state variables may be useful, but again, more research is needed.

There are, of course, many other questions that arise. Considering the relative simplicity of the problem, it is astonishing, and exciting, to see the number of research questions that arise. Having developed the Midas simulator in the 1980's, we are now embarking on the developing of an entirely new simulation library that will become a test bed for investigating some of these questions.

ACKNOWLEDGMENT

THIS RESEARCH WAS supported in part by grant DDM-9102134 from the National Science Foundation, and by grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research. Also, this research could not have taken place without the active participation of a number of graduate students and undergraduates at Princeton University. Ken Nickerson '84 wrote the original FORTRAN code that solved the stochastic network model presented in POWELL ET AL. (1988) which was the basis for the implementation presented in POWELL ET AL. (1988). David Cape '87 modified this code (CAPE, 1987) to handle the combined assignment and forecast networks which allowed us to handle the dispatching of individual drivers and loads. Linos Frantzeskakis '89 developed the first formal math-

ematical model, and contributed substantially to the debugging of the initial code. Thomas Kurian '88 provided a graphical interface, and developed the simulation model which simulated the use of the dispatch model in real-time. Brendan O'Conner '90 (1990) designed and implemented the interactive screen given in Figure 7 and conducted an extensive set of simulations that improved the performance of the model. Mary Ellen Noyes '93 (1993) performed all of the experimental results reported in Section 8. Finally, I would like to gratefully acknowledge Woody Richardson and especially Mike Crowe at Schneider National who introduced me to this problem. It was Mike Crowe who originally asked me to quantify the value of uncertainty for truckload motor carriers. After 12 years, I can finally say: not much.

REFERENCES

ARONSON, J. AND B. CHEN, "Decision Horizon Results for an Infinite Horizon, Production Planning Network Model," Technical report 85-or-3, Department of Operations Research and Engineering Management, Southern Methodist University 1985.

ARONSON, J. AND B. CHEN, "A Forward Network Simplex Algorithm for Solving Multiperiod Network Flow Problems," *Naval Research Logistics Quarterly* **33(3)**:445–467 (1986).

BEALE, E., J. FOREST, AND C. TAYLOR, "Multi-time Period Stochastic Programming," In M. Dempster (ed), *Stochastic Programming,* Academic Press, 1980.

BELL, W., L. DALBERTO, M. FISHER, A. GREENFIELD, R. JAIKUMAR, P. KEDIA, R. MACK, AND P. PRUTZMAN, "Improving the Distribution of Industrial Gases with an Online Computerized Routing and Scheduling Optimizer," *Interfaces* **13,** 4–23 (1983).

BERTSIMAS, D. J. AND L. HOWELL, "Further Results on the Probabilistic Traveling Salesman Problem," *Eur. J. Opns. Res.* **65,** 68–95 (1993).

BERTSIMAS, D. J. AND G. VAN RYZIN, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane," *Operations Research* **39,** 601–615 (1991).

BERTSIMAS, D. J. AND G. VAN RYZIN, "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidean Plane: The Multiple-Server, Capacitated Vehicle Case," *Operations Research* **41,** 60–76 (1993).

BROWN, G. AND G. GRAVES, "Real-Time Dispatch of Petroleum Tank Trucks," *Management Science* **27,** 19–32 (1981).

CAPE, D., "Micromap: A Dynamic Dispatch and Planning Model for Truckload Motor Carriers," Senior thesis, Department of Civil Engineering and Operations Research, Princeton University, 1987.

CHEUNG, R.-M. AND W. POWELL, "The Convergence of Hybrid Stochastic Gradient Methods in Stochastic Programming, with an Application to Dynamic Networks," Working paper-sor-94-02, Princeton University, 1994.

CHEUNG, R.-M. AND W. POWELL, "An Algorithm for Multistage Dynamic Networks with Random Arc Capacities, with an Application to Dynamic Fleet Management," *Operations Research* (1995).

CRAINIC, T., M. GENDREAU AND P. DEJAX, "Dynamic Stochastic Models for the Allocation of Empty Containers," *Operations Research* **41,** 102–126 (1993).

CULIOLI, J.-C. AND G. COHEN, "Decomposition/Coordination Algorithms in Stochastic Optimization," *SIAM Journal of Control and Optimization* **28,** 1372–1403 (1990).

DROR, M., G. LAPORTE, AND P. TRUDEAU, "Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks," *Transportation Science* **23,** 166–176 (1989).

DROR, M. AND P. TRUDEAU, "Stochastic Vehicle Routing with Modified Savings Algorithm," *European Journal of Operations Research* **23,** 228–235 (1986).

ERMOLIEV, Y., "Stochastic Quasigradient Methods and their Application to System Optimization," *Stochastics* **9,** 1–36 (1983).

FRANTZESKAKIS, L. AND W. POWELL, "A Successive Linear Approximation Procedure for Stochastic Dynamic Vehicle Allocation Problems," *Transportation Science* **24,** 40–57 (1990).

GAVISH, B., "A Decision Support System for Managing the Transportation Needs of a Large Corporation," *AIIE Transactions* 61–85 (1981).

GUPAL, A. M. AND L. G. BAZHENOV, "A Stochastic Method of Linearization," *Cybernetics,* 482–484 (1972).

JORDAN, W. AND M. TURNQUIST, "A Stochastic Dynamic Network Model for Railroad Car Distribution," *Transportation Science* **17,** 123–145 (1983).

LAPORTE, G. AND F. LOUVEAUX, "Formulations and Bounds for the Stochastic Capacitated Vehicle Routing Problem with Uncertain Supplies," In J. Gabzewicz, J. Richard, and L. Wolsey (eds), *Economic Decision-Making: Games, Econometrics and Optimization,* North Holland, Amsterdam, 1990.

LAPORTE, G., F. LOUVEAUX, AND H. MERCURE, "The Vehicle Routing Problem with Stochastic Travel Times," *Transportation Science* **26,** 161–170 (1992).

MAGNANTI, T. AND R. SIMPSON, "Transportation Network Analysis and Decomposition Methods," Report no. dot-tsc-rspd-78-6, U.S. Department of Transportation, 1978.

NOYES, M. E., "Validation and Testing of a Stochastic, Dynamic Fleet Management System," Master's thesis, Department of Civil Engineering and Operations Research, Princeton University, 1993.

O'CONNOR, B. M., "Real-time Dispatching for Truckload Motor Carriers: A Comparison of Graphics-Assisted Manual Dispatching and Optimization-based Dispatching," Senior thesis, Department of Civil Engineering and Operations Research, Princeton University, 1990.

POWELL, W., "A Stochastic Model of the Dynamic Vehicle Allocation Problem," *Transportation Science* **20,** 117–129 (1986).

POWELL, W., "An Operational Planning Model for the Dy-

namic Vehicle Allocation Problem with Uncertain Demands," *Transportation Research* **21B,** 217–232 (1987).

POWELL, W., "A Comparative Review of Alternative Algorithms for the Dynamic Vehicle Allocation Problem," *Vehicle Routing: Methods and Studies,* 249–292 (1988).

POWELL, W., E. BERKKAM AND I. LUSTIG, "On Algorithms for Nonlinear Dynamic Networks," In D. Du and P. Pardalos (eds), *Network Optimization Problems: Algorithms, Complexity and Applications*, World Scientific Press, New Jersey, 203–231 (1993).

POWELL, W. AND R. CHEUNG, "A Network Recourse Decomposition Method for Dynamic Networks with Random Arc Capacities," *Networks* **24,** 369–384 (1994a).

POWELL, W. AND R. CHEUNG, "Stochastic Programs over Trees with Random Arc Capacities," *Networks* **24,** 161–175 (1994b).

POWELL, W. AND L. FRANTZESKAKIS, "Restricted Recourse Strategies for Stochastic, Dynamic Networks," *Transportation Science* **28,** 3–23 (1992).

POWELL, W., Y. SHEFFI, K. NICKERSON, K. BUTTERBAUGH AND S. ATHERTON, "Maximizing Profits for North American Van Lines' Truckload Division: A New Framework for Pricing and Operations," *Interfaces* **18,** 21–41 (1988).

STEWART, W. AND B. GOLDEN, "Stochastic Vehicle Routing: A Comprehensive Approach," *Eur. J. Oper. Res.* **14,** 371–385 (1983).

TRUDEAU, P. AND M. DROR, "Stochastic Inventory Routing: Route Design with Stockouts and Route Failures," *Transportation Science* **26,** 171–184 (1992).

WETS, R., "Stochastic Programs with Fixed Resources: the Equivalent Deterministic Problem," *SIAM Review* **16,** 309–339 (1974).

WHITE, W., "Dynamic Transshipment Networks: An Algorithm and its Application to the Distribution of Empty Containers," *Networks* **2,** 211–236 (1972).

WHITE, W. AND A. BOMBERAULT, "A Network Algorithm for Empty Freight Car Allocation," *IBM Systems Journal* **8,** 147–171 (1969).

(Received, November 1994; revisions received: July 1995; accepted September 1995)