Interfaces

Vol. 40, No. 5, September–October 2010, pp. 342–352
ISSN 0092-2102 | EISSN 1526-551X | 10 | 4005 | 0342

INFORMS

DOI 10.1287/inte.1100.0510

Approximate Dynamic Programming Captures Fleet Operations for Schneider National

Hugo P. Simão, Abraham George, Warren B. Powell
Princeton University, Princeton, New Jersey 08544
{hpsimao@princeton.edu, abegeorge@gmail.com, powell@princeton.edu}

Ted Gifford, John Nienow, Jeff Day
Schneider National, Green Bay, Wisconsin 54306
{giffordt@schneider.com, nienowj@schneider.com, jeff.day@dac.us}

Schneider National needed a simulation model that would capture the dynamics of its fleet of over 6,000 long-haul drivers to determine where the company should hire new drivers, estimate the impact of changes in work rules, find the best way to manage Canadian drivers, and experiment with new ways to get drivers home. It needed a model that could perform as well as its experienced team of dispatchers and fleet managers. In developing our model, we had to simulate drivers and loads at a high level of detail, capturing both complex dynamics and multiple forms of uncertainty. We used approximate dynamic programming to produce realistic, high-quality decisions that capture the ability of dispatchers to anticipate the future impact of decisions. The resulting model closely calibrated against Schneider's historical performance, giving the company the confidence to base major policy decisions on studies performed using the model. These policy decisions helped Schneider to avoid costs of $30 million by identifying problems with a new driver-management policy, achieve annual savings of $5 million by identifying the best driver domiciles, reduce the number of late deliveries by more than 50 percent by analyzing service commitment policies, and save $3.8 million annually by reducing training expenses for new border-crossing regulations.

Key words: dynamic programming; transportation models; freight transportation; simulation.
History: Published online in *Articles in Advance* July 21, 2010.

Schneider National is one of the three largest truckload motor carriers in the United States. Over 6,000 of its 15,000 drivers participate in moving one-way truckloads, typically over distances ranging from several hundred miles to several thousand miles. These drivers often spend two weeks or more away from home, a job characteristic that contributes to driver turnover of 100 percent or more at most long-haul carriers. Schneider wanted a model to allow it to design business policies that would, among other objectives, help its drivers get home on time on a regular basis. To meet this need, Schneider contracted with CASTLE Laboratory at Princeton University to develop a model that would handle the high level of detail required for these studies and would also capture the intelligence of an experienced team of dispatchers.

Schneider needed a model that would help it answer several questions. What would be the impact of changes in federal regulations governing drivers? What would be the best way to manage drivers based in Canada? Where should new drivers be hired? How many teams (i.e., drivers who work in pairs that can operate 20 hours each day) should the company maintain? Could Schneider make commitments to drivers on when they will be given time at home?

To produce believable results, our model had to closely match actual fleet performance, corresponding with the decisions of a skilled group of dispatchers supported by state-of-the-art planning systems. To capture driver behavior in a realistic way, it had to model drivers using 15 separate attributes; to capture driver productivity, it had to represent all work rules. Our model also had to consider customer service requirements and other operational details, such as driver relays and the proper handling of geographically constrained drivers (e.g., Canadian and regional drivers).

Perhaps our biggest challenge was the requirement to design the model such that it could make real-time decisions that could anticipate their impact on the future. For example, if the model assigns a driver domiciled in one city to a load terminating in another city, will that driver be able to get home within a reasonable time? Should a team, which is best used on long loads because teams can move more miles per day, be sent to a location that primarily produces short loads? What if some of these short loads move to a location that produces longer loads?

Dispatchers clearly think about future impacts. Thus, it became clear that optimizing decisions at a point in time would not be sufficient; we had to optimize decisions *over time*. If we formulated the problem as a mathematical optimization problem, we would generate a linear program with literally hundreds of millions of rows (constraints) and billions of columns (i.e., decisions to assign drivers to loads over several weeks).

We knew that even if we could solve such a model, we would be ignoring the inherent uncertainty of truckload operations. The customer demands that would arise randomly over time were our most important sources of uncertainty, complicating the problem of getting drivers home. For example, we would like to send a Virginia-based driver to Chicago knowing that a load that would take that driver home would be waiting there; however, this is simply not how truckload operations work. Customers continually make new requests, and uncertainty in loading, unloading, and transit times makes it impossible to predict the availability of drivers and loads in the future.

We formulated the problem as a stochastic optimization model and used the modeling and algorithmic framework of approximate dynamic programming (ADP), a simulation-based type of optimization algorithm that uses iterative learning to optimize complex, stochastic problems. The extensive literature on ADP (including its sister communities, which we refer to by names such as reinforcement learning and neurodynamic programming) has largely ignored the challenges of high-dimensional decision variables. For this project, we had to develop methods to handle driver assignment problems with up to 60,000 variables in each period, millions of random variables (the demands), and a state variable with a virtually infinite number of dimensions.

To use a trite expression, necessity is the mother of invention; we knew this project would require us to develop a novel strategy to solve the problem. Our solution involved combining results from three PhD dissertations—Spivey (2001), Marar (2002), and George (2005)—to create a model that could handle the high level of detail while also producing decisions that balance current and future rewards. Equally important was the need to calibrate the model against historical performance, a dimension that the academic optimization community has largely ignored. Simão et al. (2009) give a detailed technical description of the model and the algorithm.

The Operational Problem

Truckload operations sound deceptively simple. A set of drivers is available and must be assigned to a set of loads. When a driver is assigned to a load, the driver moves to the shipper, picks up a full truckload of freight, and delivers it. When the truck is empty, the trucking company must find a new load for the driver to deliver.

In reality, truckload operations are far more complex. Drivers are described by a multidimensional attribute vector that can include many characteristics. These include current location (destination location for an en route driver), estimated time of arrival (if the driver is currently moving), domicile (driver's home location), driver type (team or single driver, driver using company-owned equipment, owner operator, and other characteristics that describe drivers who move primarily within a single region), days since last visit to home, next scheduled time or desired time at home, road hours (number of hours driven today), duty hours (number of hours the driver has been on duty today), and driver's duty hours over each of the previous seven days (to enforce the 70-duty-hours-in-eight-days rule). Loads also have a complex vector of attributes that capture pickup and delivery windows, nature of the pickup and delivery appointments, and the load priority.

When assigning a driver to a load, the dispatcher must consider factors such as the number of miles the truck must move empty to pick up the load, the driver's ability to deliver the load on time, the driver's nationality, the appropriateness of the load length for this driver type (e.g., teams are better used on long loads), the driver's ability to get home on time after delivering the load, and the productivity (miles per day) of this driver type. Dispatchers sometimes use complex strategies, such as swapping drivers between loads en route and relaying loads (dropping the loads at a location other than their destination), so that a more appropriate driver can complete the move.

Dispatchers want to minimize empty miles and move loads on time; however, they also must manage other goals. Driver turnover is a major consideration. Two objectives that are important to dispatchers are (1) getting drivers home on time (especially over a weekend) and (2) giving each driver a specific number of miles to move each week to ensure income for the driver. Schneider can impact the ability of dispatchers to get drivers home both through the choice of loads assigned to drivers (which requires thinking about the future) and by choosing where to hire drivers. Schneider was also interested in making advance commitments to drivers; however, the company wanted to ensure that it could meet its commitments at a reasonable cost.

The issue of maintaining driver productivity (measured by the average length of the loads to which a driver was assigned) was more complex. Teams must drive more miles because the work must generate income for each team member. Drivers who own their own tractors need more miles (but fewer miles than teams) because they must cover lease payments on their tractors. Single drivers using company-owned equipment need the fewest miles. Drivers who are assigned fewer miles than they normally expect may quit (and usually do), forcing the company to hire and train new drivers. However, there are no strict rules on the length of a particular load given to any driver. A team may be given a short load that repositions that team to a location that has long loads. In addition, moving a short load is better than moving nothing. What matters to drivers is the average number of miles they drive each week. If the model deviates significantly from historical averages, then it will be impossible to implement these policies without risking much higher driver turnover, something that we are simply not able to capture. For this reason, Schneider decided that the model had to maintain key driver metrics to produce a realistic simulation.

Getting drivers home on time and maintaining a targeted length of haul were two critical issues in our development of the model. However, we also had to consider on-time service, both when picking up and delivering a load. The model had to carefully observe regulations on the number of hours a driver could move each day; most notoriously, it had to consider the infamous 70-duty-hours-in-eight-days rule, which often limited a driver to substantially fewer than the allowed 14 hours on duty in one day.

To produce realistic results, we had to incorporate some of the more complex operating strategies used by the dispatchers, who struggled to meet numerous goals while keeping costs low. For example, we could not schedule a driver to move an empty truck 300 miles just to pick up a load that would take that driver near home. Instead, a dispatcher might assign driver A in Baltimore to move load 1 going to Dallas and driver B in Atlanta to move load 2 going to Chicago, recognizing that driver A needs to get home to Chicago. It might be possible to develop a schedule such that these drivers meet and swap loads, allowing driver A to finish moving the load to Chicago and get home on time with little or no additional empty-movement costs.

The Optimization Challenge

The optimization problem requires that we maximize the total revenue from moving loads (we do not require that all loads be covered) minus the cost of moving trucks to pick up and deliver loads. The model also includes a number of bonuses and penalties to encourage appropriate behaviors. These include penalties for early or late pickups and deliveries, bonuses for assigning Canada-based drivers to loads that return to Canada, bonuses for getting a driver home on time (in particular, on a weekend), and bonuses and penalties for using team drivers, for example, on longer loads. Of course, we cannot assign a driver to two loads at the same time, and each load can be covered by at most one driver. More subtly, in assigning drivers to loads, we are not allowed to use information that has not yet become known about future loads.

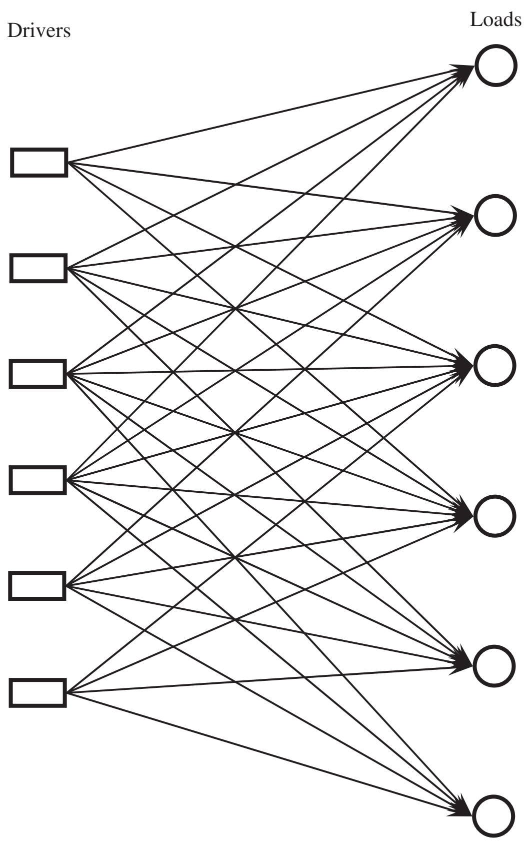

Figure 1: The diagram depicts an assignment model for assigning known drivers to known loads at a point in time.

Two optimization models are widely used in freight transportation. The first is a simple assignment problem in which resources (resources are truck drivers in our situation; they might be freight cars, locomotives, or planes in other situations) are assigned to tasks (loads of freight) (see Figure 1). This model represents every driver and load individually, making it possible to capture the attributes of each driver and load at a high level of detail at a point in time. However, at this point in time, the model cannot capture the impact of decisions about the future.

The second modeling strategy, and the most common way of modeling activities into the future, is to use a time-space network (see Figure 2). In this model, resources are represented by their location at a point in time. This model, which assumes that all resources at the same location are identical, is useful when modeling fleets of identical trailers; however, in

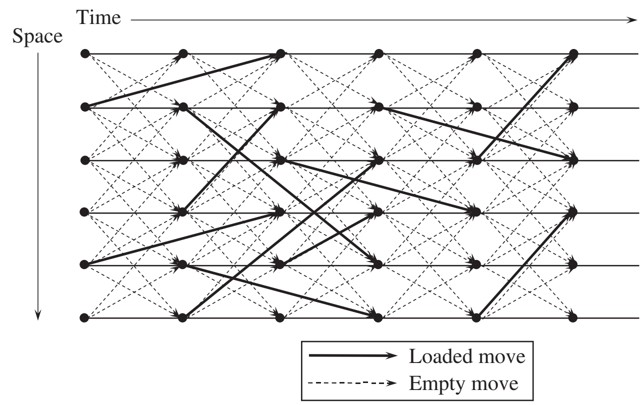

Figure 2: The diagram depicts a classic time-space network, capturing loaded movements (solid arcs) and empty movements (dashed arcs) between different locations over time.

practice, the equipment usually differs in size (e.g., older 45-foot trailers, the more common 48-foot trailers, and 53-foot trailers) and capability (e.g., refrigeration or special shock absorbers for more delicate freight).

We can handle different types of equipment if we use a multicommodity network-flow model (see Figure 3). Imagine that we are flowing each equipment type over its own time-space network, except where a loaded movement can be moved by more than one equipment type. These problems become hard to solve because we are only interested in integer solutions; i.e., we cannot assign half of a driver to a load. Modern solvers, such as CPLEX, have improved dramatically in their ability to handle large integer

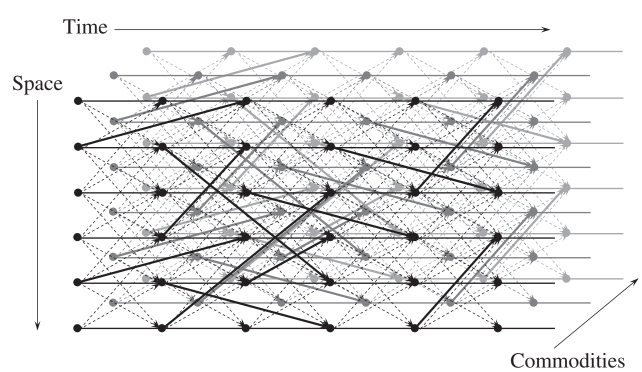

Figure 3: The diagram depicts a multicommodity flow problem defined over a time-space network in which we have represented three different types of equipment.

programming problems; however, problems such as these can have hundreds of thousands of integer variables. Despite this large size, we are still not even close to modeling the real problem.

If we discretize the country into 100 regions (which is common for truckload motor carriers), and if we have only one equipment type, then our time-space diagram (see Figure 2) would have 100 nodes per period. If we have five types of equipment, our multicommodity network would have 500 nodes per period. If we model individual drivers and capture only 100 locations, five driver types, 100 home domiciles, and up to 30 days since the last visit to home, we already have 1.5 million nodes per period. If we try to capture all the attributes, the number of possibilities is effectively infinite.

The complexity of the problem arises when we combine the detailed vector of attributes required to model a driver with the need to make decisions that anticipate the future impact of the decisions. Figure 4 illustrates an initial possible assignment of five drivers to five loads. Focus now on a single driver with attribute vector a_3. This driver might be assigned to each of the five loads, creating a new driver in the future; this new driver has attributes that depend on the initial attribute vector a_3, along with the characteristics of each load. To know if we should assign our driver to each of these five loads, we have to think

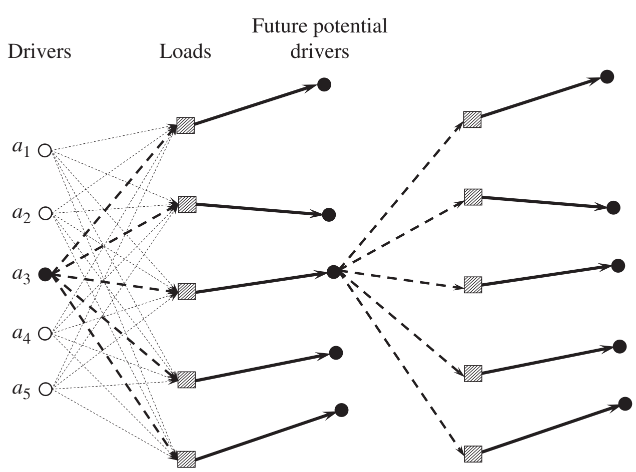

Figure 4: The diagram illustrates the growth in the number of potential drivers in the future when we consider possible assignments over time.

about what this particular driver might do after completing each of these five loads (and so on and so on). If we extend this logic over the course of a month in which a driver might be expected to handle 12 loads, we might create about 240 million ($5^{12} = 2.44 \times 10^8$) potential future drivers. Now multiply 240 million by the 6,000 drivers in our fleet (it is virtually impossible to have two drivers with identical attributes). The resulting network would have over 1.4 trillion nodes *per time period*.

This problem as stated is difficult enough; however, we now must also introduce the dimension of uncertainty. Customer demands, which come in on a rolling basis, are the most significant sources of uncertainty in truckload trucking. In addition, we must capture randomness in travel times. It is easy to claim that because of the uncertainty, we do not have to look into the future; however, our work demonstrates that to accurately capture the behavior of Schneider's experienced dispatchers, thinking about the future impact of a decision is critical. Furthermore, we must realize that we will need the marginal value of a driver over the entire simulation.

The presence of uncertainty guided us to an elegant and practical solution. Even small versions of this problem cannot be solved exactly using any standard stochastic optimization framework. For this reason, we turned to the modeling and algorithmic framework of ADP, which offers both a rigorous mathematical foundation and the property of being very intuitive. We start with the simple concept of solving a sequence of simple assignment problems (see Figure 1). After assigning drivers to loads, we simulate random travel times and new loads, advance the clock, and solve the problem again. Of course, if we do only this, we would be simulating only a simple myopic policy that provides none of the important qualities we need to solve this problem.

Instead, we solve a somewhat modified assignment problem. Rather than only assigning drivers to loads, we modify the assignment problem to capture an approximate value of drivers in the future. Figure 5 illustrates this concept. We start with the same assignment problem shown in Figure 1; however, we now add an approximation of the value of a driver after completing each load. Therefore, if we are considering assigning the driver with attribute vector a_3 to load 1,

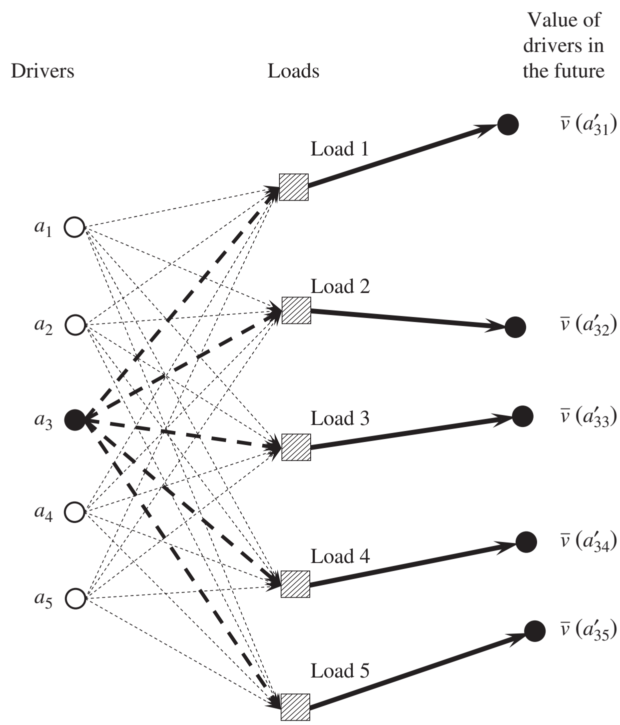

Figure 5: We show the driver assignment problem with the approximate value of drivers in the future.

we would obtain a driver with attribute vector a'_{31}, with an approximate value of $\bar{v}(a'_{31})$. We estimate the value $\bar{v}(a'_{31})$ by using the marginal value of drivers in the future.

This simple concept introduces some technical complications, which Simão et al. (2009) summarize in more detail. First, we might never have assigned the driver with attribute a_3 to load 1, which means that we were never allowed to observe a driver with attribute a'_{31} in the future. Instead, we view the problem of estimating the value $\bar{v}(a'_{31})$ as a statistical exercise in which we need an estimate $\bar{v}(a)$ for any attribute a. Although we can present this as a simple exercise in statistical estimation, estimating these values introduces enough issues to fill a doctoral dissertation and several research papers. For example, the value of a driver at one point in time depends on the value of drivers in the future, which are also approximations. In addition, a characteristic of our problem is that many drivers live in some parts of the country (e.g., Illinois and Georgia) and relatively few drivers live in other locations (e.g., South Dakota and Nevada). We require methods that allow us to take advantage of the large number of observations in the more active parts of the country while still handling the areas that have relatively few observations.

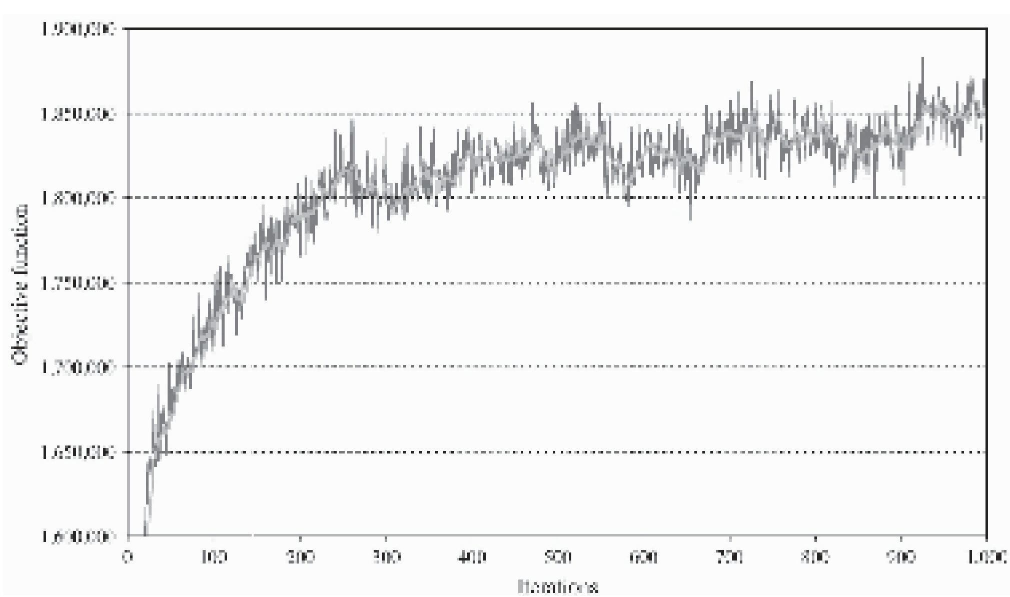

Figure 6: The improvement in the objective function illustrates optimizing behavior.

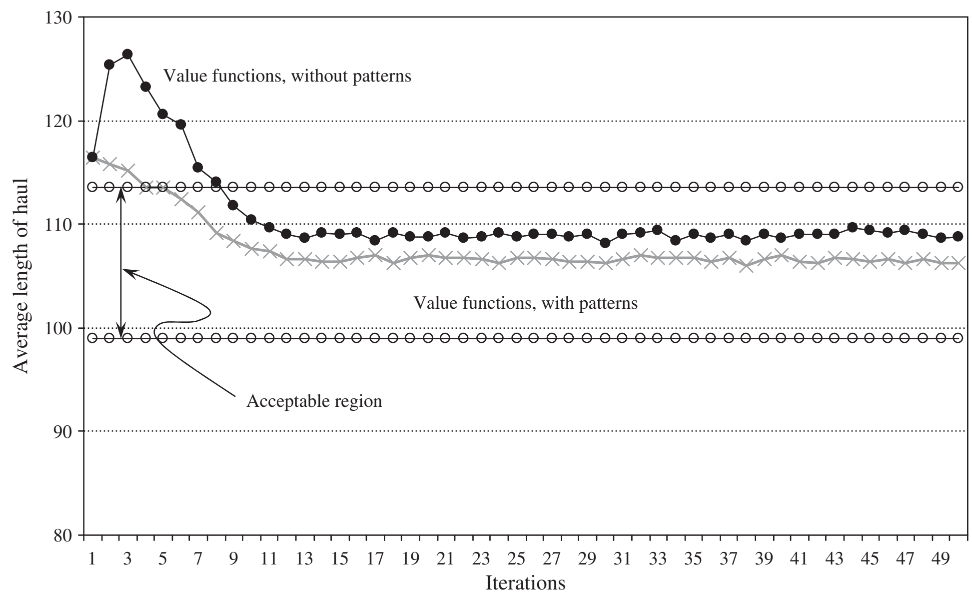

Figure 7: We show the effect of value functions and patterns on matching historical performance.

Estimating the value-function approximations requires simulating the dispatch process (using a particular set of approximations) iteratively. After each iteration (in which we simulate a month of dispatching), we use information from each assignment problem to update the value of drivers in the future. We capture uncertainty by sampling any random variables (e.g., new loads or travel times) as we simulate decisions over the month. Because of the need to sample from random quantities, we must use smoothing techniques to balance out the noise. Again, this apparently simple step of smoothing proved to be another difficult research challenge.

We undertook extensive research in the use of ADP for fleet management problems and showed that we can obtain solutions that are near optimal when compared with the optimal solution of deterministic, single, and multicommodity flow problems (Godfrey and Powell 2002, Topaloglu and Powell 2006). However, we could not obtain optimal solutions (or even tight bounds) for the problem class described in this paper. Instead, we first looked for evidence that the algorithm is generally producing improvements in the overall objective function (see Figure 6). This improvement is significant because the first iteration, in which we set $\bar{v}(a) = 0$, is equivalent to a myopic policy. The more meaningful validations are that (1) we produce results that closely match historical performance, and (2) the value functions $\bar{v}(a)$ accurately approximate the marginal value of increasing the number of drivers of a particular type.

Matching Patterns

As we mentioned in the previous section, simply optimizing the objective function was not sufficient to obtain realistic behaviors. A major issue that Schneider faced was the need to assign drivers to loads of an appropriate length. Teams expected the longest loads, whereas single drivers using company-owned equipment were given the shorter loads. However, a team could be assigned a shorter load; moreover, although single drivers using company-owned equipment were assigned shorter loads, they still needed to drive enough miles per week to maintain their income.

Solving this problem by simply using penalties when the length of a load is higher or lower than the average for a particular type of driver is impossible. Every driver type needs to pull loads of different lengths—only the averages count. Assigning a team to a shorter load is not a problem, as long as the average length of haul matches historical data. If we deviated significantly from historical averages, it is likely that we would start incurring higher driver turnover. Therefore, Schneider was unwilling to experiment with dispatch rules that deviated from its historical patterns.

We solved this problem by introducing the concept of a *pattern metric*, as Marar et al. (2006) and Marar and Powell (2009) propose. In brief, it involves adding a term that penalizes deviations from the historical percentage of times that drivers of a particular type (e.g., team, single) take loads of a particular length (e.g., between 500 and 550 miles). As decisions are made over time, the model keeps track of an estimate of how often we move loads of a particular length over the entire horizon. We only introduce penalties when the average over the entire horizon starts deviating from historical performance. This logic also requires iterative learning; therefore, it fits naturally within the iterative learning of the value functions. Section 4.1 in Simão et al. (2009) describes the application of the pattern logic to this problem.

Figure 7 illustrates how the model learns to match historical performance using value functions with and without a pattern. Our goal was to move a particular metric (in this case, average length of haul for a particular class of driver) within a range that would be acceptable to Schneider. We note that by simply using value functions, we were able to move the model within the acceptable range. Introducing the patterns moved the metric within the allowable range more quickly and moved it closer to the center of the range. The pattern logic played a significant role in generating other statistics, which we highlight in the next section.

Calibrating Against History

The central goal of our research was the development of a model that we could use to analyze the effect of changes in policies (e.g., allowing drivers to drive 11 hours per day instead of 10) or inputs (e.g., hiring more team drivers housed in a particular location). If the results of these analyses are to be credible, the model must closely match historical performance. For example, if we did not model the rules governing hours of service, we would not be able to analyze the effect of changes in the rules that limit the number of hours a driver can be on duty to fewer than 70 hours in eight days. Alternatively, if simplifications in the model allowed us to overstate the productivity of a driver, then we would not obtain a realistic estimate of the marginal value of additional drivers of a particular type.

Accurately capturing the attributes of drivers and loads and the rules governing how long a driver can be on the road is not enough. To have a realistic simulation, we must closely mimic the *decisions*. Because we are testing changes in policies and inputs, we have to believe how the decisions will change under these new inputs. For example, if we want to trust the estimates of the marginal value of team drivers in a location, we must believe that the model is making realistic decisions about assigning team drivers to longer loads. If the model starts assigning team drivers to shorter loads, then it is making decisions that the company would not make in practice, thus overstating the value of team drivers.

Despite extensive academic literature on the optimization of truckload carriers, we could find only a single instance of a model that has been shown to calibrate against actual historical performance; Simão et al. (2009) provide a detailed technical description of this model. The research community that works on the development of models for freight transportation has primarily focused on optimization models in which the goal is to outperform the decisions made by a company.

A central tenet of our project was that a carefully constructed *optimization* model would closely *simulate* decisions made by the dispatchers. It is important to emphasize that we are optimizing a utility function that uses various bonuses and penalties to achieve realistic behaviors, such as getting drivers home, achieving appropriate utilization for different driver types, and realizing on-time customer service. However, we believe that if the decisions of the dispatchers were significantly suboptimal, our model would not have calibrated against history. Our numerical work showed that as the objective function improved (i.e., we were doing a better job looking into the future because the model automatically optimizes at a point in time), the model became more realistic, with no evidence that we were outperforming the dispatchers. We have to keep in mind that Schneider pioneered the use of optimization models for real-time

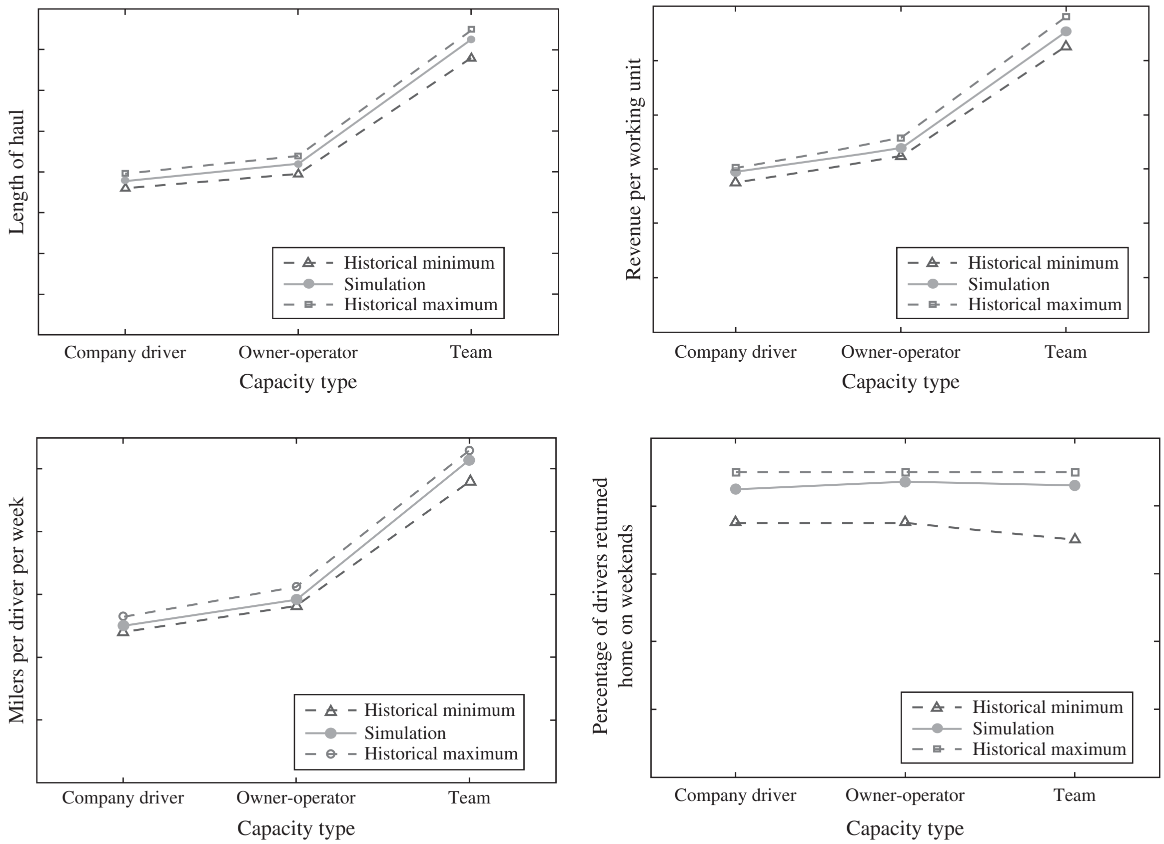

Figure 8: We show system results compared against historical extremes for length of haul, revenue per working unit, driver utilization, and percentage of driver time-at-home spent on weekends. See Simão et al. (2009) for details.

dispatch. In addition, the company uses an array of advanced decision support systems; therefore, we view this result as gratifying rather than surprising.

Figure 8 shows comparisons we made between our model's results and those obtained based on historical data. For each statistic, Schneider provided a range based on the variability it observed on a month-to-month basis. As the figure shows, we were able to calibrate the model to match history for each statistic.

Capturing the Marginal Value of Drivers

One application of the model was guiding Schneider in its hiring of new drivers. For example, we needed the marginal value of hiring teams who live in northern Illinois. This marginal value would reflect both the revenues of the loads that these drivers can move and the cost of getting the drivers home. It is important to emphasize that the marginal value of these drivers differs greatly from the average value of drivers with these attributes, a quantity that can be easily calculated by tracking the paths of these drivers over the planning horizon. The marginal value requires that we understand how the solution would *change* if we added additional drivers of a particular type.

We compared the marginal value of a driver characterized by domicile and driver type as estimated by the value-function approximations against the estimates produced by adding 10 drivers of a particular type and rerunning the system for several iterations. Given the noise in the estimates obtained by adding 10 drivers and rerunning the system, we repeated this

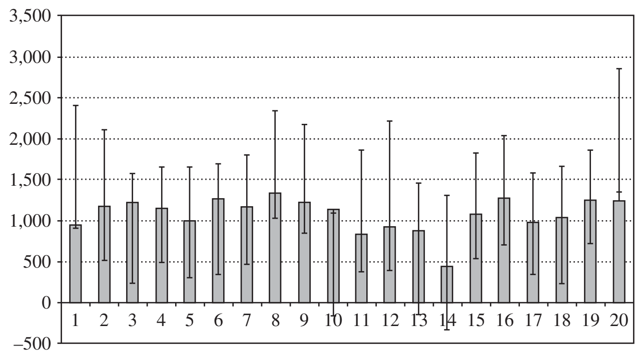

Figure 9: We show the simulated value of additional drivers compared to estimates based on value-function approximations for 20 different types of drivers. See Simão et al. (2009) for details.

exercise 10 times for each type of driver to obtain confidence intervals (i.e., we performed a simulation).

Figure 9 shows the results of our simulation. We note that for the 20 different estimates (representing different combinations of driver domiciles and driver types), in 18 instances, the value produced by the value-function approximation fell within the 95 percent confidence interval from the simulation. We view this as a validation that the value-function approximations are consistent with the estimates of the marginal values produced through simulation. However, from a single calibration run of the system, we obtained estimates, $\bar{v}_0(a)$, for all possible combinations of driver domiciles with driver types at the beginning of the simulation. That is, $\bar{v}_0(a)$ is an estimate of the value of a driver with attribute a at time 0, which provides an estimate of how the entire simulation should change if we add one more driver with attribute a at the beginning of the simulation. We have to compute $\bar{v}_t(a)$ for all attributes (e.g., location, driver type, domicile) and all periods as part of the ADP algorithm; however, for driver valuations, we use only the estimates at the beginning of the simulation.

Having an Impact

Schneider's tactical planning system (TPS, as our model has become known within the company) has been and continues to be used for a variety of analyses that lead to significant financial benefits through operational policy and procedure improvements, better-informed negotiating positions, and cost reductions or avoidance. The system's principal benefit over the traditional aggregated-flow network models used previously is its capability to capture driver and load attributes in great detail and to produce a very realistic simulation of real-world operations. The particular strengths of this modeling platform are that it (1) produces good (near-optimal) solutions to complex problem scenarios and (2) provides comprehensive operating characteristics and statistics that can be used to determine potential impacts and uncover unintended consequences associated with proposed changes within a complex transportation network. Schneider, a company that has won the INFORMS Prize for its widespread use of operations research, was unable to solve the problem using standard optimization or simulation modeling tools.

In the last several years, Schneider used TPS to analyze several situations. Below, we summarize these analyses and show their corresponding business benefits.

- Driver time at home: Long-haul drivers are typically away from home for two to three weeks at a time. To address driver retention, Schneider management approved a business plan to significantly increase the amount of time drivers spend at home; however, TPS runs showed that the plan had a potential annual negative impact of $30 million on network operating costs, considerably outweighing the anticipated benefits. Schneider used TPS to develop an alternative strategy that provided 93 percent of the proposed self-scheduling flexibility but incurred an estimated annual cost impact of only $6 million.
- Driver hours-of-service rules: During the last six years, the US Department of Transportation has introduced several changes for driver work-schedule constraints. Using TPS runs, Schneider was able to substantiate and quantify the impacts of these changes, allowing it to effectively negotiate adjustments in customer billing rates and freight tendering and handling procedures, and leading to margin improvements of 2 to 3 percent.
- Setting appointments: A key challenge in the order-booking process is determining both the timing and flexibility of the pickup and delivery appointments. Using TPS, Schneider was able to quantify the

impacts of different types of commitments, allowing it to identify the best choices. This produced margin improvements in the range of 4 to 10 percent and reduced the number of late deliveries by more than 50 percent.

• Cross-border relay network: The Schneider freight network includes a large number of loads that move between the United States and Canada. Using TPS runs, Schneider was able to design a strategy that accomplished cross-border operations using only Canadian drivers. This reduced the number of drivers that crossed these borders by 91 percent, avoiding $3.8 million in training, identification, and certification costs, and providing annual cost savings of $2.3 million.

• Driver domiciles: Schneider manages over 6,000 drivers who operate its long-haul network; these drivers must be away from home for weeks at a time. Getting them home requires sending drivers to specific regions from which there is a good likelihood that they can get home on time. As a by-product of the ADP methodology, TPS provides an estimate of the marginal value of drivers for each home domicile. Schneider uses these estimates to guide its hiring strategy, resulting in an estimated annual profit improvement of $5 million.

• Time-window reduction: One of Schneider's largest customers asked for tighter time windows on delivered freight, covering 4,500 loads per month. When Schneider used TPS to show that it would cost approximately $1.9 million per year to meet this demand, the customer withdrew the request.

References

George, A. 2005. Optimal learning strategies for multi-attribute resource allocation problems. Doctoral dissertation, Princeton University, Princeton, NJ.

Godfrey, G. A., W. B. Powell. 2002. An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times. *Transportation Sci.* **36**(1) 21–39.

Marar, A. 2002. Information representation in large-scale resource allocation problems: Theory, algorithms and applications. Doctoral dissertation, Princeton University, Princeton, NJ.

Marar, A., W. B. Powell. 2009. Capturing incomplete information in resource allocation problems through numerical patterns. *Eur. J. Oper. Res.* **197**(1) 50–58.

Marar, A., W. B. Powell, S. Kulkarni. 2006. Combining cost-based and rule-based knowledge in complex resource allocation problems. *IIE Trans.* **38**(2) 159–172.

Simão, H. P., J. Day, A. P. George, T. Gifford, J. Nienow, W. B. Powell. 2009. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Sci.* **43**(2) 178–197.

Spivey, M. J. 2001. The dynamic assignment problem. Doctoral dissertation, Princeton University, Princeton, NJ.

Topaloglu, H., W. B. Powell. 2006. Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS J. Comput.* **18**(1) 31–42.