Quantifying the value of advance load information in truckload trucking

Darsono Tjokroamidjojo a,*, Erhan Kutanoglu b,1, G. Don Taylor c,2

a North Dakota State University, Department of Industrial and Manufacturing Engineering, Fargo, ND 58102, United States
b Operations Research and Industrial Engineering Program, Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, United States
c Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, United States

Received 26 July 2004; received in revised form 8 December 2004; accepted 28 January 2005

Abstract

We present an optimization-based computational study to quantify the relative benefits and costs of sharing advance load information and preparing an advance pick-up and delivery plan (or preplanning) in load assignment problems in truckload trucking. Considering that the benefits of preplanning comes in many intangible forms, we take the strategy of showing the cost differences between preplanning and different forms of dispatching, instead of computing the benefits directly. Computational study under several settings not only shows the benefit of using advanced load information, but also uncovers the minimal cost difference between intelligent preplanning and the best performing dispatching policies.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Transportation; Load assignment; Driver assignment; Preplanning; Advance load information; Optimization

* Corresponding author. Tel.: +1 701 231 7223; fax: +1 701 231 7195.
E-mail addresses: Darsono.Tjokroamidjojo@ndsu.edu (D. Tjokroamidjojo), erhank@mail.utexas.edu (E. Kutanoglu), don.taylor@vt.edu (G.D. Taylor).
1 Fax: +1 512 232 1489.
2 Fax: +1 540 231 3322.

1366-5545/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2005.01.001
1. Introduction and motivation

Making the best allocation of limited resources while keeping a high level of customer service is a common challenge in many companies including transportation service providers, or carriers. To address this challenge, carriers are increasingly starting to investigate ways of collaborating with their shippers, and their shippers’ partners (vendors and customers). One obvious and potentially less costly way of collaboration between a carrier and its shippers is via timely communication of load information (from the shippers to the carrier) and of a pick-up/delivery plan (from the carrier to the shippers). Considering that transportation activities are essential links that tie many entities, one can think that this type of collaboration would be an important step to improve both the shippers’ and the carrier’s performance, and ultimately to optimize the whole supply chain. At the very least, the goal of collaboration is to smooth the flow of both information and material between entities in the supply chain and to eliminate inefficiencies, including those of carriers. In this study, we conduct an optimization-based computational study to quantify the relative benefits and costs associated with shippers providing advance load information and carriers preparing (or preplanning) an advance pick-up and delivery plan.

Quality advance load information can be very beneficial for truckload trucking carriers because their customer base usually sees servicing a load as the purchase of a commodity. Truckload shippers are likely to wait until the same day of a load pick-up need before booking transportation. The truckload industry is so fragmented that a shipper can find available carriers at reasonable prices even at the day of the required pick-up. Hence truckload shippers do not have any incentive or reason for advance booking. Most shippers have their internal reasons for not revealing their load information ahead of time. A major reason is the decision making flexibility that can be gained associated with delaying commitment time. Without advance load information, the carrier must delay decisions about load assignments, and follow a “last-minute dispatching” approach. Only then can the carrier myopically optimize its current decisions considering the latest available information about the available loads and drivers.

There are additional benefits of having advance load information and preplanned loads. Preplanned drivers are less likely to have excessive dwell periods between loads, and are more satisfied, thus lowering the driver turnover rate which is significant in the industry. Shippers are better able to plan for future events and are less likely to have late pick-ups. The carriers can achieve better trip and fuel stop planning while minimizing empty repositioning (deadhead) costs.

The challenge in this new setting is to explicitly quantify the benefits of advance load information and preplanning. Only then can one compare the benefits of advance load information and preplanning with the associated costs, and decide whether the new strategy of using advance load information has a net positive benefit. Even though quantifying the benefits of advance load information and/or preplanning is a challenge, one can look at the total cost difference between preplanning and last-minute dispatching to decide if the difference is significant enough to provide incentives to the shippers to encourage them to reveal their load information ahead of time. In this paper, we take the strategy of showing the cost differences between the preplanning strategy and the dispatching strategy under several scenarios and settings. Ultimately, the planner and the carrier have to decide if the difference is significant or not.

Motivated by a real problem at J.B. Hunt Transport, Inc., the largest publicly held truckload trucking company in the US (J.B. Hunt web site (2004)), we formulate a model for the “load
assignment problem” (LAP). In its static form, as it has been used before, the model assigns a static set of available loads to a static set of drivers as a one-time activity. The dynamic implementation of this model is the key to quantifying the value of preplanning versus last-minute dispatching. Moreover, the dynamic implementation provides insights to how early that the advance load information should come to gain the maximum value from preplanning.

2. Literature review

Many transportation and logistics problems have been investigated extensively in the literature. There are optimization models specifically developed for transportation problems, such as vehicle routing and pick-up and delivery problems. Most studies have focused on the static versions of these problems, where all load information is available at the time of planning. Just to name a few in this vast area, we list static vehicle routing problems (Golden and Assad, 1988; Fisher, 1995). The dynamic version where the load information becomes available gradually as the shipper requests arrive over time has been unnoticed until recently. The literature here includes dynamic vehicle routing problems (Powell, 1995; Psaraftis, 1995; Bertsimas and Simchi-Levi, 1996; Gendreau and Potvin, 1998). Another important problem in this domain has been that of assignment of loads to trucks (or equally to drivers). This load assignment problem finds the minimum-cost assignment of loads to trucks, given a set of loads, where each load is a full truck-load size, and each is to be picked up at a given location at a specific time and delivered to a known destination.

The dynamic load assignment problem with advance load information considerations, to the best of our knowledge, has not been investigated before. According to Powell’s (1995) definition, our dynamic load assignment problem is also a dynamic application in a realistic dynamic setting as both loads and the information about the loads become available over time. Moreover, we implement the same model under static information assumptions to obtain benchmark solutions.

The dynamic load assignment problem, in its simplest form, can be described as a simple network problem of matching driver/truck nodes to load nodes over time (Powell, 1995). Generally, the simple driver assignment problem does not have the tour building capability, in which drivers are assigned to a series of loads. This simple version has been used in practice as a dispatching tool because of its extreme simplicity and its easy solution using network optimization techniques. However, this model cannot account future impacts of the current decisions. For example, the model cannot make driver repositioning recommendations, which would likely be necessary if there are too many drivers concentrated in a certain region at a certain time, nor can it make load rejection recommendations if there are too many profitable loads from which to choose. The model presented in this study has the built-in tour-building capability and look-ahead features that are essential to make use of advance dynamic load information. In this respect, the most relevant modeling effort is by Keskinocak and Tayur (1998), who use a similar model for aircraft scheduling.

There are several papers dealing with the simple dynamic driver assignment problem. Powell (1996) builds a model that assigns drivers to loads on a real time basis by taking into account uncertainties about demand for truckload motor carriers. He presents a methodology for evaluating the dynamic assignment models in a continuous setting using rolling horizon simulations.
The developed stochastic model handles both known and forecast demand considering multiple periods of travel times. The stochastic model is then extended to take into account uncertainty factors such as the possibility that the recommendations of the model were not applied (Powell et al., 2000).

There are also studies that take a simulation-based or reactive approach. For example, Regan et al. (1998) develops an evaluation method for dynamic truckload operations with real time information using a simulation framework. More recently, Wang and Regan (2002) develop stochastic assignment models to represent stochastic travel times. Similarly, Yang et al. (1999) describes a methodology to dynamically reassign trucks to loads, as real-time information becomes available. Most recently, Yang et al. (2004) extend this to a study that is arguably the most relevant work to ours. They investigate reoptimization techniques to be used when the new load information becomes available within a pickup and delivery problem. They do not, however, address the preplanning and dispatching issues, especially these with load-ahead feature. Moreover, their model is more detailed as it tries to capture time-related complexities within the model, whereas ours handle them at a preprocessing stage.

As this review depicts, there has not been a focused research study on the advance load information and its effect on preplanning and dispatching in truckload trucking. While no extensive body of literature is found on the use of advance load information in transportation, a more in-depth search shows that there has been a recent interest on the issue in a different context, production planning and inventory management. Although it is prohibitive to list all related work, we provide references to two most recent studies that report benefits of such explicit modeling and analysis of advance demand information: Gallego and Ozer (2001) develop a form of optimal inventory policy where the benefits of advance demand information are realized. Similarly, Lu et al. (2001) investigate the value of advance demand information under an assemble-to-order system for component replenishment.

None of the models have specifically addressed the focal issues of this research, the analysis of when assignment decisions should be made and the value of advance load information in truckload transportation. Our contribution is to fill this void by conducting an optimization-based computational study.

3. Dynamic policies using advance load information

The load assignment problem finds the minimum-cost solution for assigning a given set of available loads to a set of trucks. The cost function is the total of the cost of empty miles due to repositioning trucks between loads, the dwell costs due to idle periods, and lateness penalty costs. Every load must be served, either by a carrier truck or by a subcontracted truck (when it is impossible or uneconomical to use the carrier’s own trucks). It is possible to assign more than one load to a truck over time as long as the truck can meet their pick-up and delivery times within the allowed maximum lateness.

To formulate the load assignment problem in a dynamic environment, we first consider that available loads are distributed over time. In previous implementation of such models, it is assumed that the model is static, i.e., at the time of planning, the planner has perfect information on a static set of loads that are to be served over a specific planning horizon (see, for example,
Fisher, 1995). In practice, however, the load information comes in pieces and the information at the time of planning may not be complete. For example, a subset of loads that will require service during a time period, say Tuesday–Friday, may not be known ahead of time at the time of planning, say on Monday. Moreover, the planner can change the assignment of a specific load until a decision is fixed (or frozen) or even until a truck is actually dispatched to pick up the load. Hence, both the set of available loads and the decisions themselves are dynamic. In this study, we explicitly consider the timing of advance load information (how much time in advance with respect to its pickup time a load is known, i.e., the amount of advance load information) and the timing of fixing the load assignment decision (how much time in advance a load’s assignment to a specific truck is fixed; i.e. the amount of preplanning).

We now define the alternative policies that a carrier can use in different advance load information settings and preplanning requirements. We define “dispatching” as waiting until the last minute to fix a load’s assignment decision. This usually means making the load’s assignment on the same day as the load’s pickup time. However, dispatching also implies that the load information has not been made available in advance, which means the load’s assignment decision is made once and immediately fixed. That is, the planner using a dispatching policy has to make myopic assignment decisions without looking ahead and without explicitly considering the effects of these decisions on future assignments.

“Dispatching with look-ahead” is a more refined and sophisticated policy. In this case, the carrier encourages its shippers to give advance load information, uses all or part of the available information about current and future loads, and still makes the decisions in a dispatching fashion. In this case, the planner considers the effects of current decisions on future loads, but freezes only the decisions that are immediate. This strategy permits planners to use as much information as possible about the future loads while delaying fixing decisions about them until absolutely necessary.

“Preplanning” is a policy that allows the carrier to have advance load information but requires the assignments to be done in advance also. Depending on when the planner has to fix the assignment decisions, preplanning can take one of several versions. For example, if loads have to be fixed at the same time that they are available, then the planner will not have “look-ahead” capability. In this case, if the planner knows a load’s information $n$ days in advance, then its assignment must be made $n$ days in advance also. In more flexible versions of preplanning, the planner can have the load information $n$ days in advance, but the assignment decision does not have to be fixed until $m$ days ($m < n$) before its pick-up time.

4. Problem statement and mathematical model

We use an optimization-based approach to analyze the value of advance load information. The overall approach is to build a deterministic mathematical model, and use it dynamically over time to emulate the dynamics of load arrivals and changing load assignment decisions. To this end, we first define the static version of the mathematical model, and then explain its dynamic implementation.

To formulate the problem defined verbally in previous sections, we modify the aircraft scheduling model presented in Keskinocak and Tayur (1998) to our truckload trucking environment. The modified model itself is simpler due to the nature of the truckload business (e.g., there are
no “number of landings” considerations). The usage of this model in this new setting, which is dynamic, is one of the primary contributions of this research. We first introduce the inputs of the model along with the notation used throughout the paper.

4.1. Inputs

$I$ set of trucks, indexed by $i = 1, \ldots, I$

$J$ set of loads, indexed by $j$ and $k = 1, \ldots, J$

$K$ set of locations/cities, indexed by $m$, and $n = 1, \ldots, K$

$\gamma_i$ initial location/city of truck $i$ at the time of initial planning

$z_j$ departure city of load $j$ (pick-up location)

$\beta_j$ destination city of load $j$ (delivery location)

$d(m, n)$ travel time between city $m$ and $n$

$a_j$ departure time of load $j$

$\tilde{b}_j$ destination time of load $j$. We define $\tilde{b}_j = a_j + d_j$, where $d_j$ is the “service time” for load $j$, which is in general the time it takes to pick up load $j$ from its departure city, drive, and deliver it to its destination city. In many cases, this is the travel time from load $j$’s departure city to its destination, i.e., $d_j = d(z_j, \beta_j)$. In general, however, it can include other elements for greater flexibility. Moreover, note that travel/drive time information can be converted to distance using average speed information. In this way, one can use distance-based costs instead of time-based costs. In our formulation, all costs are defined based on time.

$v(j)$: preplanning status of load $j$. In the preplanning policy, some of the loads will be preplanned and their assignments will be fixed. If load $j$ has already been assigned to a truck in the model’s previous runs and that decision is “frozen” (cannot be changed in this planning horizon), then $v(j)$ shows the truck number to which the load is assigned. The model’s current run does not change the assignment of these fixed loads. If load $j$ has not been assigned to any truck (free to be assigned to any of the available trucks in the current planning horizon), then $v(j)$ takes on a value of 0. Sometimes, a driver arrives at a pick up location before the departure time of a load. The amount of time spent idle by the driver/truck is called dwell time. We define two versions of dwell time, one for an initial load picked up by a truck and another for other load pickups.

$w^0(i, j)$: The amount of dwell time experienced if truck $i$ serves load $j$ first. This is the extra time between the initial location of truck $i$ and the departure location of load $j$. One can preprocess the input data and obtain $w^0(i, j) = (a_i - d(\gamma_i, z_j))^+$, where $(x)^+ = \max(0, x)$.

$w(j, k)$: The amount of dwell time experienced if a truck serves load $j$ and then load $k$. Considering the available time between the delivery time of load $j$ and the pick-up time of load $k$, one can compute $w(j, k) = ((a_k - b_j) - d(\beta_j, z_k))^+$. Similar to dwell times, we define two types of lateness in pick-up times:

$l^0(i, j)$: The amount of lateness experienced if truck $i$ serves load $j$ first. This is positive only if there is not enough time to travel to departure city of load $j$ by its pick-up time. One can preprocess the input data and obtain $l^0(i, j) = (a_i - d(\gamma_i, z_j))^-$, where $(x)^- = \min(0, x)$.

$l(j, k)$: The amount of lateness experienced if a truck serves load $j$ and then load $k$. Considering the available time between the delivery time of load $j$ and the pick-up time of load $k$, one can compute $l(j, k) = ((a_k - b_j) - d(\beta_j, z_k))^-$.
e: The cost of empty miles captured in the time dimension ($/h), i.e., driving empty one hour costs $e dollars.

f: The hourly dwelling penalty cost.

g: The hourly lateness penalty cost.

In our model, we assume that all loads must be assigned to a truck; hence all loads must be served. Under this assumption, a load can be subcontracted to another carrier due to unavailability of a company-owned truck or due to economic reasons. This assumption can be relaxed or changed depending on the real-life situation. We also assume that subcontracting carries a higher cost than serving the loads with a regular company-owned truck. Hence, we define the cost of subcontracting load $j$ to be $s_j = s d_j$, where $s$ is a scalar that can be adjusted according to affordability or desirability of subcontracts.

4.2. Preprocessing for time-based restrictions

To handle the dynamics of the problem and related cost and constraint issues, one can model time (either in continuous or in discrete form) explicitly in the problem formulation. In our formulation, however, we preprocess the input data and handle the dynamics of the decisions and time-related constraints a priori. This allows us to formulate the remaining problem without explicit reference to time. In preprocessing the input data, we create two pieces of new data that restrict (1) feasible truck–load combinations, and (2) feasible multiple load assignments to any truck, which controls the assignment of a load immediately after another load. For both data, we first define a new parameter:

e: The maximum amount of delay in pickup (lateness) that is allowed by the planner.

For the first preprocessed data, we define binary parameters $T L_{ij}$ for all trucks $i$ and all loads $j$. $T L_{ij}$ takes a value of 1 if truck $i$ can serve load $j$ and 0 otherwise. Restrictions on the availability of trucks for certain loads are captured here. The restrictions can be in terms of the initial location of the truck or the load, in terms of technical constraints such as trailer type or capacity, or anything else that would limit the feasible/possible combinations of (truck, load). For example, if truck $i$ cannot serve load $j$ within the maximum allowed lateness $u$, then truck–load assignment $(i,j)$ is not possible, i.e., if $l(i,j) > u$ then $T L_{ij} = 0$. Note that the maximum lateness allowed ($u$) parameterizes $T L_{ij}$. However, as we have noted above, other restrictions can also be handled in $T L$.

The second preprocessed data item is the parameter $L L_{jk}$ which handles the combinations of loads that are possible to serve one immediately after another. Again, many restrictions can be handled explicitly in $L L$, but the main use of this new data is to model the time dimension outside the formulation. Hence, the main restriction is due to pick-up and delivery times of the corresponding loads, i.e., there may not be enough time for any truck to drive from the destination location of one load to the pick-up location of the second (without being late more than $u$ hours). Hence, $L L_{jk}$ is 1 if load $k$ can be fulfilled immediately after load $j$ by the same truck with a lateness that is not more than the allowed maximum lateness $u$, i.e., $l(j,k) = [(a_k - b_j - d \beta_j - \alpha_k)] < u$, and 0 otherwise. Here, again the introduction of $u$ parameterizes the data captured in $L L$.

If more lateness is allowed by the planner (increase in $u$), more truck–load and load–load combinations will be “feasible”. For example, when $u$ is 5 h, then we allow loads being delivered late up to 5 h. If $u$ is set to 0, lateness is not allowed. One can similarly define limits on the maximum
allowable dwell time and further parameterize TL and LL. In the experimental study, we consider
the lateness parameter explicitly.

4.3. Decision variables

We are now ready to define the decision variables of the model. The first set of variables takes
care of the assignments of the loads to company-owned trucks:

\[
X_{ijk} = \begin{cases} 
1, & \text{if truck } i \text{ is assigned to load } j \text{ then load } k, \\
0, & \text{otherwise}
\end{cases}
\]

and

\[
\text{TL}(i,j) = \text{TL}(i,k) = \text{LL}(j,k) = 1
\]

\[
X_{0k} = \begin{cases} 
1, & \text{if truck } i \text{ is assigned to first load } k, \\
0, & \text{otherwise}
\end{cases}
\]

\[
X_{00} = \begin{cases} 
1, & \text{if truck } i \text{ is not assigned to any load} \\
0, & \text{otherwise}
\end{cases}
\]

Note that these variables are only defined for the feasible truck–load and load–load combinations.
This way, preprocessing the data to create TL and LL not only helps model the time dimension
implicitly but also reduces the number of variables, and the number of constraints, as will be seen
shortly. The second type of decision variables handle subcontracting as defined below:

\[
Y_j = \begin{cases} 
1, & \text{if load } j \text{ is subcontracted} \\
0, & \text{otherwise}
\end{cases}
\]

In the following, we present the mathematical model as an integer programming problem.

Minimize

\[
\begin{align*}
& e \sum_{i \in I} \sum_{k \in K} d(\gamma_i, z_k)X_{0k} + e \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} d(\beta_j, z_k)X_{ijk} \\
& + f \sum_{i \in I} \sum_{k \in K} X_{00} + f \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} w^0(\gamma_i, z_k)X_{0k} \\
& + f \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} w(\beta_j, z_k)X_{ijk} \\
& + g \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} l^0(\gamma_i, z_k)X_{0k} + g \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} l(\beta_j, z_k)X_{ijk} + \sum_j s_j Y_j
\end{align*}
\]

subject to

\[
\sum_{i=1}^{I} \sum_{j=0}^{J} X_{ijk} + Y_k = 1 \quad \forall k \text{ if } v(k) = 0
\]

\[
\sum_{j=0}^{J} X_{ijk} = 1 \quad \forall i, k \text{ if } v(k) = i
\]

\[
\sum_{k=0}^{K} X_{0k} = 1 \quad \forall i
\]

\[
\sum_{k=0}^{K} X_{ijk} - \sum_{q=0}^{J} X_{iqj} = 0 \quad \forall i, j, \quad \text{TL}_{ij} = 1
\]

\[
X_{ijk} \in \{0, 1\} \text{ and } Y_k \in \{0, 1\} \quad \forall i, j, k, \quad \text{TL}_{ij} = 1, \quad \text{LL}_{jk} = 1
\]
We minimize the total costs of empty miles due to repositioning trucks, dwell times, lateness of delivery, and subcontracting. All of these elements are captured in the objective function (1) of the model. The empty miles (and corresponding driving times) are due to two sources, each captured in a separate term in the objective function. The first source (first summation in the objective) is moving a truck empty from its initial location to serve a load in another city/location as its first load. The second source of empty miles (second summation in the objective) is repositioning the trucks (moving them empty) between two loads (i.e., between the destination location of one load to the departure location of the next one), if they are served consecutively by the same truck.

There are three types of dwell costs. The first type (third summation in the objective function) is the costs incurred when a truck is not assigned to any loads. The second type (fourth summation) arises when a dwell time is experienced by a truck driving from its initial location to the pickup location of its first load. The fifth summation accounts for dwell time experienced between loads consecutively served. The sixth and seventh summation terms are included to capture lateness costs that are due to the first loads served by specific trucks and that are due to lateness experienced between loads, respectively. Finally, the last term is the total subcontract costs.

The first set of constraints (2) state that if a load has not been assigned to a specific truck previously (i.e., \( v(j) = 0 \)), then it should be served either by a company-owned truck or a subcontracted truck. Constraints (3) ensure that if a load has been pre-assigned to a truck, the model will use that pre-assignment. This allows a certain load to be pre-assigned to a certain truck. In many cases, we can assume that this set is empty; then, there would be no constraints of (3).

Constraints (4) and (5) cover the truck assignments. Constraints (4) state that every truck is either assigned to a first load (that is first for the truck) or it is not assigned to any loads at all. This means a solution of the problem does not have to utilize all available trucks. Constraints (5) state that if truck \( i \) serves load \( k \) right after load \( j \) then load \( j \) must be either the first load for the truck or must be served after another load.

The formulation of the load assignment problem would be significantly different (and more complicated) if the time-related constraints were handled explicitly in the model. The offline handling of time constraints is an advantage of our approach. Another advantage of the model is its built-in flexibility to handle time-based or load-based maintenance or other regulatory restrictions (e.g., modeling Department of Transportation duty time restrictions and sleep requirements is easy in the current model). The inclusion of these issues would certainly make the model more realistic, but due to our focus on dynamics of the system and timing issues, we leave that detail for a future study.

Keskinocak and Tayur (1998) present several maintenance-type constraints in their model used for an aircraft scheduling problem. They show that when there are no prescheduled loads and no maintenance restrictions, the problem can be solved rather easily as it can be reformulated as a minimum cost network flow problem on a directed acyclic graph. However, in its general form, when there are either scheduled loads or maintenance restrictions, the load assignment problem is NP-hard. Although we use the model to gain managerial insights on the timing of decisions and on the value of early information on loads, we should note that the problems are computationally easy problems when formulated for reasonable numbers of loads, time periods, and prescheduled trips. To create a more realistic environment, we handle these timing-specific issues (availability of advance load information and preplanning) outside the model. This leads to a
rather unique implementation of the static model in a dynamic environment. The next section explains this implementation in more detail.

5. Dynamic implementation

In this section we explain how we make use of the static mathematical model introduced in the previous section in a dynamic environment. In this new environment, we assume that loads show up over time and that information about loads (their pick-up and delivery locations and times) becomes available some time before their corresponding pick-up times. We also consider the timing of decisions explicitly: when the planner has to make the assignment decisions and commit trucks to certain loads. More specifically, how long before the load’s departure time does the planner have to fix a load assignment? To create a controlled experimental setting for both the availability of advance load information and the amount of preplanning, we use two parameters: Load Knowledge Window (LKW) and Decision Time Window (DTW). The load knowledge window (denoted by $\tau$) refers to the time window during which the planner knows about the load’s availability. The decision time window (denoted by $T$) is the number of days in advance of a load’s departure time that the planner has to fix the load’s assignment decision. For example, if $T$ is 1 day, then the scheduler has to fix the assignment decisions for loads on Tuesday by Monday at the latest. $T$, of course, cannot be greater than $\tau$. We use $\tau$ and $T$ to emulate the dynamic availability of load information, and to assess its impact on the preplanning activity and the decisions themselves.

The load knowledge and decision time window parameters affect the set of available loads that are considered in a specific execution of the static model. Hence, every time the new information becomes available, the model that is populated with the corresponding data is solved (an instance of the load assignment problem). Part of the solution generated in every execution of the model with the corresponding set of known loads will be fixed according to the decision time window parameter, $T$. Let $L(t_1, t_2)$ be the set of loads with pick-up times between $t_1$ and $t_2$, or $L(t_1, t_2) = \{j \in J : t_1 < a_j < t_2\}$.

---

3 We specify different policies characterized by the two parameters, $\tau$ and $T$.

[$\tau = 1$, $T = 1$]. The planner knows loads 1 day in advance (1 day before their departure times) and makes (and fixes) the assignment decisions 1 day in advance. Note that this setting corresponds to pure dispatching (wait till the last minute) policy outlined earlier.

[$\tau = 3$, $T = 1$, and [$\tau = 5$, $T = 1$]]. For the first policy, the planner has load availability information 3 days in advance but has to make assignment decisions 1 day in advance. The second policy, [$\tau = 5$, $T = 1$], is the same, except that the load assignment decisions are known 5 days in advance but they don’t have to be fixed until 1 day before the loads’ departure times. Note that these settings are two different versions of “dispatching with look ahead”.

[$\tau = 3$, $T = 3$], [$\tau = 5$, $T = 5$] [$\tau = 5$, $T = 3$]. These are three forms of preplanning policy. For the first policy, we have information about the loads 3 days in advance and we also make assignment decision 3 days in advance. Similarly, we have a policy in which we have information about the loads 5 days in advance with assignment decisions being made 5 days in advance. We should note, though, that the policy that leaves some flexibility in preplanning is the third one [$\tau = 5$, $T = 3$], in which the planner does not have to fix the decisions for 2 extra days even though he/she has the load information in finding the solutions.
We now give the details of the dynamic implementation:

Step 1: Set the load knowledge window \((\tau)\), and the decision time window \((T)\). Consider the following set of alternatives: \([1,1], [3,1], [3,3], [5,1], [5,3], [5,5]\).

Step 2: For every time period \(t\) (assume \(t = 1, \ldots, H\), where \(H\) is the length of the overall planning horizon to be used in the experiments), create an instance of the load assignment problem by populating it with the corresponding data and fix the portion of solution obtained. More specifically:

Step 2.1: Define \(LAP(t)\) to be the load assignment problem with loads that have pick-up times from \(t + 1\) to \(t + \tau\), \(L(t + 1, t + \tau)\).

Step 2.2: Solve \(LAP(t)\) for the optimal assignments \(X(LAP(t))\).

Step 2.3: Fix the assignments for loads, whose pick-up times are between \(t + 1\) and \(t + T\), i.e., \(X'_j = X_j(LAP(t))\) for all \(j\) in \(L(t + 1, t + T)\). Since we have \(v(j) = i\) for some \(i\) for these decisions, insert necessary constraints into the \(LAP(t + 1)\), and update \(t\).

Step 3: Report necessary total costs collected through fixed decisions for all the loads under consideration. For reasons that will become clear in the discussion regarding the experimental study, we report two total cost values: one for all the loads in the whole planning horizon, and the other for loads in the “middle portion” of the overall planning horizon. Choong et al. (2002) support the use of a planning horizon on the order of 15–30 days in a related study in the barge industry. In this study, we utilize a 20-day planning horizon. In the computational study, we examine costs for the entire 20-day planning horizon, and also for the loads in \(L(5,15)\).

A benchmark or performance baseline for the dynamic implementation is the somewhat unrealistic solution of the same LAP model for all available loads simultaneously; say at the beginning of the overall planning horizon. However this benchmark solution is useful in determining what the best possible total costs would be if perfect information about all the loads were available at once. We call this as the static benchmark implementation. The static implementation would assume that the planner knows about all of the loads and all their characteristics (costs, locations and times) on day 1. The static benchmark solution also makes the assignment decisions for all the loads on day 1. Since there is no new information during the rest of the planning horizon, we do not need to revise the assignment decisions; hence this provides the ultimate minimum cost solution.

For models where \(T\) is greater than 1, there are anomalies in the beginning and in the end of the overall planning horizon. For example, for \(\tau = 5\) and \(T = 3\), loads that are departing on day 2 or day 3 must have been decided before day 1. Similarly, at the end of planning horizon with the same model, on day 18 we should have information about loads departing on day 21. The same observation can be made for days 19 and 20. For the cases where \(T = 3\) or 5, we assume that those loads departing on day 2 and day 3 are decided on day 1. We also assume that on days 18, 19 and 20 we will not have any further information about the loads departing on day 21 or later. To minimize the effect of these anomalies, we use a planning horizon of 20 days and have 50 loads arriving over 20 days. As indicated earlier, we compare the models in two ways; one with the performance during the overall 20-day planning horizon, and the second with the performance between days 5 and 15. The second approach is taken to ‘normalize’ the effect of the beginning and ending anomalies.
6. Computational study

To fully investigate the effects of knowing load information and making assignment decisions early or late, we use the following experimental factors:

(1) Preplanning/dispatching policies, each with a combination of \([\tau, T]\), six levels in total. This factor has been discussed in detail earlier.
(2) Size of the underlying transportation network, with two levels, a 10-city network and a 50-city network.
(3) The maximum allowed lateness \((u)\). We use two levels of this factor: 0 and 5 h. Note that when it is set to 0, lateness is not allowed. Using this as an explicit factor, we will see if there is any inherent flexibility in dispatching/preplanning if lateness is allowed.

To find out the relative performances of the policies, we randomly generate 20 problems for each experimental design point. For each problem, we generate the following data:

- The locations \((x, y)\) coordinates of the cities are determined randomly with a discrete uniform distribution from a Cartesian coordinate system of 20 by 20 \((20 \times 20\) grid). The set of cities produces the underlying transportation network.
- We generate 50 loads that are serviced among the cities. Origin and destination cities for loads are chosen randomly among those cities with a discrete uniform distribution.
- The initial locations of trucks are randomly generated from the \(20 \times 20\) grid. The initial locations are considered as drivers’ domiciles.
- The departure time of each load is randomly generated with a discrete uniform distribution from the beginning of the planning horizon to the end of the planning horizon with a predefined interval. In this research, we used an interval of 720 min (12 h). The overall planning horizon is fixed to be 20 days.
- The distance between cities is calculated using the Euclidean distance function.
- Both the hourly dwelling penalty cost \((f)\) and the hourly lateness penalty cost \((g)\) are set at $25 per hour.

Every 10-city and 50-city load assignment problem is solved and emulated with the dynamic implementation using each of the policy settings. The costs for all days in the overall planning horizon are reported along with the costs over the mid-10-day period (i.e., days 5 through 15 as discussed earlier). For benchmarking purposes, each problem is also solved using the static implementation. We should note that although our model allows loads to be subcontracted, the optimal solutions of the all problem instances assign all the loads to the carrier trucks in the experiments.

Tables 1 (for \(u = 0\)) and 2 (for \(u = 5\) h) show the average total costs (across 20 problems in each size category) both for all loads during the overall horizon of the problem (20 days) and for the loads during the mid-10 days. From these tables, we see that the static implementation always gives the lowest costs for 20 days. This is expected, since the static implementation unrealistically exploits all opportunities for optimization by considering all the loads simultaneously. Also, the average cost of the static optimal schedule over mid 10-day period is still lower than the costs of
more realistic policies. There are instances though where preplanning/dispatching policies outperform the static benchmark for middle 10-day performance, as a 20-day static optimum might have a suboptimal mid-10-day schedule.

Tables 1 and 2 also show that with some level of lateness allowed (from 0 to 5 h), we can achieve lower total costs due to lateness flexibility despite the penalty cost. The interesting overall result is that relative ordering of the policies (and their comparison with the static optimum) does not seem to be affected by the network size and the maximum allowed lateness. One detail is that the larger (or denser) networks magnify the differences among the policies, especially for mid 10-day performance, as shown in Figs. 1–4 for both levels of lateness and both network sizes. However, for both network sizes and for both 10-day and 20-day costs, the dispatching with look-ahead ([3, 1], [5, 1]) and the preplanning with look-ahead [5, 3] policies produce very close performances to the static optimal, especially for the 10-day performance horizon. The [5, 1] dispatching policy with look-ahead attains the lowest total costs over the 10-day period in 6 out of 20 problems with 50 cities, and in 8 out of 20 problems with 10 cities when lateness is not allowed. When the preplanning/dispatching policies are compared among themselves (without the static optimum benchmark), the relative benefit of waiting till the last minute and considering as much information as possible before committing assignments is even more visible both in average performance and in number

### Table 1
Total costs for all policies averaged across 20 problems in each setting for \( u = 0 \)

<table>
<thead>
<tr>
<th>Horizon</th>
<th>Network size 50 cities</th>
<th>Network size 10 cities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 days</td>
<td>20 days</td>
</tr>
<tr>
<td>[1, 1]</td>
<td>79,470</td>
<td>219,985</td>
</tr>
<tr>
<td>[3, 1]</td>
<td>61,791</td>
<td>193,445</td>
</tr>
<tr>
<td>[3, 3]</td>
<td>80,166</td>
<td>220,015</td>
</tr>
<tr>
<td>[5, 1]</td>
<td>58,508</td>
<td>189,016</td>
</tr>
<tr>
<td>[5, 3]</td>
<td>61,722</td>
<td>193,286</td>
</tr>
<tr>
<td>[5, 5]</td>
<td>76,460</td>
<td>217,763</td>
</tr>
<tr>
<td>Static</td>
<td>56,833</td>
<td>183,955</td>
</tr>
</tbody>
</table>

* [3, 1], [5, 1] and [5, 3] are significantly different and better than [1, 1], [3, 3] and [5, 5].

### Table 2
Total costs for all policies averaged across 20 problems in each setting for \( u = 5 \) h

<table>
<thead>
<tr>
<th>Horizon</th>
<th>Network size 50 cities</th>
<th>Network size 10 cities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 days</td>
<td>20 days</td>
</tr>
<tr>
<td>[1, 1]</td>
<td>75,007</td>
<td>209,581</td>
</tr>
<tr>
<td>[3, 1]</td>
<td>57,870</td>
<td>185,729</td>
</tr>
<tr>
<td>[3, 3]</td>
<td>75,949</td>
<td>209,704</td>
</tr>
<tr>
<td>[5, 1]</td>
<td>55,163</td>
<td>182,249</td>
</tr>
<tr>
<td>[5, 3]</td>
<td>58,240</td>
<td>185,982</td>
</tr>
<tr>
<td>[5, 5]</td>
<td>72,041</td>
<td>208,316</td>
</tr>
<tr>
<td>Static</td>
<td>54,597</td>
<td>177,722</td>
</tr>
</tbody>
</table>

* [3, 1], [5, 1] and [5, 3] are significantly different and better than [1, 1], [3, 3] and [5, 5].
of instances that the policies finding the lowest costs. Similar observations can be made when limited lateness is allowed with \( u = 5 \).

The results in Tables 1 and 2 further show that using a pure dispatching or pure preplanning (without any flexibility to look-ahead) produces much higher costs. For example, the policies with \([1,1]\) and \([3,3]\) yield the worst average costs in both 10-day and 20-day performances for both network sizes. The performance of the pure preplanning policy with \([\tau = 5, T = 5]\) also does not produce better results than the worst two. This can be explained by recalling that the model with these policies is essentially the least flexible one among the preplanning models and has a much more limited view of multiple days with multiple load assignment opportunities.

These observations are strengthened by ANOVA tests conducted for the six policies. ANOVA tests conducted (not shown for brevity) indicate that the difference among the policies is statistically significant as observed by a very low \( P \)-value (essentially 0) in every category of problems. Furthermore, Fisher’s statistical pairwise comparison tests indicate that there are two distinct groups of policies, with a statistically significant performance difference between the groups. These tests show that the dispatching with look-ahead (\([\tau = 3, T = 1]\), and \([\tau = 5, T = 1]\)) and the flexible preplanning policy (\([\tau = 5, T = 3]\)) are significantly better than the performances of
pure dispatching (no look-ahead $[\tau = 1, \ T = 1]$) or pure preplanning ([$\tau = 3, \ T = 3$], and [$\tau = 5, \ T = 5$]), with no significant differences within the group members. Further statistical analysis shows that at least one of the policies (sometime all three) in the former group is not distinguishable from the static optimum in all cases and settings except the 20-day costs in networks of 10 cities. In this case, the static optimum is statistically better, but still the performance of the best policy in the former group is very close, thus practical significance is an issue.

One part of this result is expected: Looking ahead and considering the effects of decisions in dispatching improves the performance of both dispatching (compare [3, 1], and [5, 1] with [1, 1]) and preplanning (compare, e.g. [5, 1], and [5, 3] with [5, 5]). From the same results, we can also see that when you do look ahead, using the available load information (i.e., when $\tau$ is greater than 1 day, or look-ahead with 3 or 5 days), having a shorter $T$ (delaying decisions which induces some flexibility) tends to improve results. For example, the policy [5, 5] has a worse performance than the models [5, 3] and [5, 1]. Similarly, the policy [3, 3] also has a worse performance than the policy [3, 1]. In other words, if you have load information $\tau$ days in advance where $\tau > 1$, it is wise not to make all load assignment decisions immediately. That is why dispatching with look-ahead
produces overall best results, which are very close to the static benchmark optimum. Having at least partial load information (3–5-day look-ahead, instead of all 20 days) in conjunction with some delay in committing decisions performs over time almost as well as if planners had the perfect information for all 20 days at once.

More interestingly, the preplanning policy \([5, 3]\) performs as well as these outstanding performers and the static implementation. There is minimal additional cost to follow a preplanning policy as long as it has some flexibility in considering the effects of the plan on other loads using an appropriate delay with \(T < \tau\). We do not need to wait till the last minute (as in dispatching) to fix the decisions to exploit all optimization opportunities. Instead we can practically preplan for the loads as long as we have enough time to consider the effects of preplanning on other loads.

As stated previously, Figs. 1–4 graphically depict the average differences between the policies and the static implementation optimum. Figs. 1 and 2 present results for 20-day horizon and mid 10-day horizon, respectively, when no lateness is allowed. Similarly, Figs. 3 and 4 plot average differences between the policy performances and the static optimum for \(u = 5\) h. The distinction between the two groups of policies mentioned earlier is very apparent.

Another result is that the policy with \([s = 5, T = 1]\) is not statistically different from the policy with \([s = 3, T = 1]\). This means to be able to take advantage of load information, we do not need 5-day advance notice. Even a 3-day advance notice would be enough time to benefit from advance load information. Note that both policies implement dispatching with look-ahead.

We believe that the parts of the savings obtained through preplanning with look-ahead \((5, 3)\) or dispatching with look-ahead \((5, 1), (3, 1)\) (as compared to pure dispatching \((1, 1)\)) can be used to provide incentives to the customers to implement the idea of providing advance load information. Also, additional savings are possible due to benefits to improve driver trip/fuel stop planning, reduce driver turnover, and other coordination-based benefits obtained by having preplanning. Although, these are not captured explicitly in the model, the savings through preplanning and advance load information are very real and are one of the reasons why the carriers are interested in having better advance load information.

7. Managerial insights and conclusions

There are many studies in the literature that focus on static transportation logistics problems where information about all loads is available at the time of initial planning. However, there has not been a computational study that explicitly investigated the dynamics of information where loads arrive in a dynamic fashion and load information becomes available over time. We conducted an extensive computational study to quantify the relative benefits and costs of preplanning that utilizes available load information with respect to pure dispatching and associated static benchmarks in the context of full truckload transportation. We simulated the dynamics of the system by repetitively solving a load assignment optimization model which is being used in the truckload setting for the first time. We obtained practical insights into how a truckload carrier can take advantage of the available load information to lower costs.

The computational results show that utilizing an appropriate delay in fixing the assignment decisions while considering all available information along the way (dispatching with look-ahead) leads to outstanding performance, as expected. Only in the future can we know the effects of
current decisions on the future loads. However, there are preplanning policies that work as well as dispatching with look-ahead when their parameters are appropriately set. The results also lead to two significant observations in terms of the overall performance of the preplanning policies: (1) Using advance load information for preplanning purposes without additional flexibility to consider the effects of preplanning on other loads (the planner knows about the loads ahead of time and has to fix their decisions right away) produces average costs as high as the pure dispatching policy (without look-ahead), as if early information is not beneficial. (2) On the other hand, if the preplanning policy has some flexibility in delaying the decisions and in considering the effect of preplanning on the other loads, the planner can do as well as dispatching with long look-ahead and perform very close to the full/perfect information static optimum. This basically means that there is no additional cost of preplanning if the policy allows some delays in committing to load assignments.

The planner can use the minimum cost differential between preplanning and dispatching with look-ahead to provide incentives to the shippers so that they will reveal advance load information to the carrier. Knowing that there is little (and statistically insignificant) cost increase as compared to dispatching with look-ahead, the planner can try to conservatively estimate the benefits of preplanning and justify its use with a net positive benefit. These conclusions can be drawn for both small and large transportation networks, and both for the overall planning horizon and for only a relevant portion of the horizon.

For future research, we can refine the simple policies here and make them more sophisticated. One such modification could be the addition of a “frequency” parameter. One can reduce the computational burden of solving an LAP model everyday by solving it every other day, or with an appropriately selected frequency. In addition, we can explore other factors that could affect the relative performance of the policies. Such factors include the demand distribution over time—uniform versus more realistic distributions (for examples, more loads at the beginning and end of the work weeks), the market type—backhaul (bad market in terms of load density) versus headhaul (good market), and load type—intermodal versus truck only. Another factor that one can investigate is the relative performances of the preplanning policies under stochastic conditions. Load service time uncertainty, pick-up and delivery time changes, and load cancellations are just a few of sources of uncertainty in dynamic load assignment problems. In an effort to more fully examine related factors along with uncertainty, we anticipate using simulation in future studies.

References


