Quantifying the Value of Advance Load Information in Truckload Trucking:

Insights for Preplanning in Load Assignment Problems

Darsono Tjokroamidjojo and Erhan Kutanoglu*
Department of Industrial Engineering
University of Arkansas
4207 Bell Engineering Center
Fayetteville, AR 7203

G. Don Taylor
Department of Industrial Engineering
University of Louisville
Louisville, KY

* Corresponding author, e-mail: erhank@engr.uark.edu

Abstract

A way of collaboration between a transportation service provider and its customers is *timely* communication of load information (from customers) and of a pick-up/delivery plan (from the service provider). Motivated by real challenges faced by a large carrier, we conduct an optimization-based computational study to quantify the relative benefits and costs of sharing advance load information and preparing (or preplanning) an advance pick-up and delivery plan 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 (fixing assignment decisions in advance) and dispatching, the method of choice by many practitioners. Computational study under several settings does not only show the benefit of using advanced load information, but also undercover that the minimal cost difference between intelligent preplanning and the best performing policies such as dispatching with look-ahead. This means that the carrier can and should provide reasonable incentives for shippers to have them communicate their load information early knowing that this will bring benefits to both sides. Finally, the results also show that to get the benefits it is not necessary to have the load information very early, just couple of days advance notice could be enough.

Keywords: Transportation, Load Assignment, Driver Assignment, Preplanning, Advance Load Information, Optimization

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, increasing number of carriers are investigating ways of collaborating with their main service users (or shippers), and their shippers' partners (vendors and customers). One obvious and potentially least costly way of collaboration between a carrier and its shippers is *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 least, the goal of collaboration is to smooth the flow of both *information* and material between entities in the supply chain and eliminate inefficiencies, including those of carriers, the transportation service providers. In this study, we conduct an optimization-based computational study to quantify the relative benefits and costs of sharing advance load information (coming from shippers) and preparing (or preplanning) an advance pick-up and delivery plan to be used by shippers.

Quality advance load information could be very beneficial for a truckload trucking carrier whose customer base sees servicing a load as a purchase of a commodity. Truckload shippers usually wait till the last minute to book their loads (maybe not just before the load's pickup time, but more likely the same day of the load). Without advance load information, the carrier also delays its decision about the load assignments, and follows 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.

However, the benefits of having advance load information and preplanned loads are several. Preplanned drivers are less likely to have excessive dwell period between loads, and are more satisfied thus lowering the drivers' turnover rate. Customers are better able to plan future events and less likely to have late pickups. The carriers can achieve better trip and fuel stop planning while minimizing empty repositioning (deadhead) costs.

The challenge in this new setting is quantifying the benefits of preplanning explicitly. 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 in their specific conditions is significant or not.

Motivated by a real problem at JB Hunt Transport, Inc., the largest publicly held truckload trucking company in the U.S. (JB Hunt web site (2002)), we formulate a stylized model for the so-called "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 one time activity. The dynamic implementation of this model is the key to address the questions raised earlier, as to quantify the value of preplanning versus last minute dispatching. Moreover, the dynamic implementation provides insights to how much early that the advance load information should come to gain the maximum value from preplanning.

2. Literature Review

There are currently many optimization models, each dealing with different aspects of transportation and logistics. One important problem that has been addressed by many has been the assignment of loads to trucks (or equally to drivers) by using several approaches, either exact or heuristic. We call this load assignment problem: For a given set of loads, each with size of full truckload to be picked up at a certain location at a certain time and to be delivered to a delivery location, find the minimum-cost assignment of loads to trucks.

Although static versions of this problem in many contexts have been tackled, the dynamic version with advance load information considerations, to the best of our knowledge, has not been investigated in detail before. For the dynamic problems and their "dynamic representations (mathematical models), we follow Powell's definition (1995). Powell differentiates "*dynamic problem*", "*dynamic model*" and the "*application*" of a model. A *dynamic problem* uses parameters that are functions of time. There are two kinds of problems in this regard: (1) one that uses dynamic data like real-time load information as it becomes available, and (2) one that uses time-dependent data, which is available at once at the beginning of planning horizon. For the latter, one can give the example of a vehicle routing problem with all information (which is a function of time) known in advance. A *dynamic model* represents and shows the interaction of activities over a time horizon. A *dynamic application* is when a model (which can be a static model) is solved repeatedly over time as new real information about loads arrives. As will be clearer, our *dynamic load assignment problem* is a dynamic model, which considers time-based interactions implicitly. We also present its *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 assumption to obtain benchmark solutions.

In transportation and logistics, there are many dynamic models including (1) the dynamic traveling salesman problem, 2) the dynamic traveling repairman problem and 3) the dynamic driver assignment problem.

The dynamic traveling salesman problem is when a driver must visit a series of points at each of which there is a constant task to perform (Powell, 1995). Psaraftis (1988) provides an example of the dynamic traveling salesman problem, where demands need to be satisfied by a salesman who travels from one location to another and provides a needed service at each demand location. Another version of the dynamic traveling salesman problem is called the time-dependent traveling salesman problem. Picard and Queyranne (1978) and Malandraki (1989) describe the problem by making the travel times vary as a function of time.

In the dynamic traveling repairman problem, tasks require a random length of time at each point. This version can be viewed as special case of the so-called dynamic and stochastic vehicle routing problem (Powell, 1995). Bertsimas and Van Ryzin (1991) investigate this problem with uniformly distributed task times and Poisson task arrivals, where the entire network is served by a single uncapacitated vehicle.

The closest to our load assignment problem, the dynamic driver assignment problem, in its simplest form, can be described as a simple network problem by matching driver nodes to load nodes over time (Powell, 1995). Generally, the simple driver assignment problem does not have the tour building capability/requirement, in which drivers are assigned to a series of tasks (or loads). This simple version has been used in industry because of its extreme simplicity and its easy solution using network optimization codes. However, this model cannot account future impacts of the decisions made now. For example, the model cannot make any driver repositioning recommendations, which could be necessary if there are too many drivers concentrated in a certain region, or it cannot 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 feature that are essential to make use of advance dynamic load information. In this respect, the closest 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 forecasted demand considering multiple periods of travel times. The stochastic model is then refined to take into account uncertainty factors such as the possibility that the recommendations of the model were not applied (Powell, 2000).

There are also studies that have taken a simulation-based or a reactive approach. For example, Regan (1998) develops an evaluation method for dynamic truckload operations with real time information using a simulation framework. Similarly, Yang, et al (1999) describes a methodology to dynamically reassign trucks to loads, as real-time information becomes available. More recently, Wang and Regan (2002) develop stochastic assignment models to represent stochastic travel times.

As this review depicts, there has not been a focused research on the advance load information and its effect on preplanning and dispatching in truckload trucking. While no 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 studies, we give 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, Song and Yao (2001) investigate the value of advance demand information under an assemble-to-order system for component replenishment.

None of the models have specifically addressed on the focal issues of this research, the analysis of when assignment decisions should be made and the value of advance load information in 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 is finding the minimum-cost solution for assigning a given set of available loads to a set of trucks. The cost function is the total of empty miles costs due to repositioning of trucks between loads, dwell costs due to idle periods, and lateness penalty costs. Every load has to 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 make their pick-up and delivery times within the allowed maximum lateness.

To formulate the load assignment problem in a dynamic environment, first we 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. 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 his/her mind in terms of the assignment of a specific load until actually a decision is fixed (or *frozen*) or even until a truck is actually dispatched to pickup 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 – 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 – 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*" herein as waiting until 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 more refined and sophisticated. 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 could provide the best of both worlds by permitting dispatchers to use as much information as possible about the future loads while delaying fixing decisions about them until necessary, i.e., just in time.

"*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 has to 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 its assignment decision does not have to be fixed until *m days* ($m<n$) before its pick-up time.

4. Problem Statement and Mathematical Model

To answer the questions raised in previous sections, we use an optimization-based approach. 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 is 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.

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 $n = 1, \ldots, K$

γ_i: initial location/city of truck i at the time of planning.

α_j: departure city of load j.

β_j: destination city of load j.

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

a_j: departure time of load j.

b_j: destination time of load j. In the case of deterministic travel times, $b_j=a_j+d_j$, 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, $d_j = d(\alpha_j,\beta_j)$. 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.

$v(j)$: 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.

$w^0(i,j)$: 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 $w^0(i,j) = (a_j - d(\gamma_i,\alpha_j))^+$, 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 $w(j,k) = ((a_k - b_j) - d(\beta_j,\alpha_k))^+$.

$l^0(i,j)$: The amount of lateness 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 $w^0(i,j) = (a_j - d(\gamma_i,\alpha_j))^-$, 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 $l(j,k) = ((a_k - b_j) - d(\beta_j,\alpha_k))^-$.

e: The cost of empty miles captured in the time dimension, i.e., driving empty one hour costs *e* dollars.

f: The hourly dwelling penalty cost. In the experimental study, this is equal to $25 per hour or $600 per day.

g: The hourly lateness penalty cost. In the experimental study, this is equal to $25 per hour.

In our model, we assume that all loads should be assigned to a truck, i.e., all loads should be served. Under this assumption, a load is subcontracted to another carrier in case of unavailability of a company-owned truck or due to economic reasons. This assumption can be relaxed or changed depending on the real-life scenarios. We also assume that subcontracting costs quite a bit more 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 the scalar that can be adjusted according to affordability or desirability of subcontracts.

Preprocessing for Time-Based Restrictions:

To handle the dynamics of the problem and to handle 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 beforehand. This allows us to formulate the remaining problem without explicit reference to time in the model. In preprocessing the input data, we create two pieces of new data that restricts (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:

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

For the first preprocessed data, we define binary parameters TL_{ij} for all trucks i and all loads j. TL_{ij} takes a value of 1 if truck i "can serve" load j. Restrictions on the availability of trucks for certain loads are captured here. The restrictions can be in terms of the 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 the only way for truck i to serve load j with lateness that is more than the allowed amount u, then truck-load assignment (i,j) is not possible, i.e., if $l^0(i,j) > u$ then $TL_{ij} = 0$. Note that the max-lateness allowed (u) parameterizes TL_{ij}. However, as we have noted above, other restrictions can also be handled in TL.

The second preprocessed piece of data is the parameter $LL_{j,j'}$ which handles the combinations of loads that are possible to serve one immediately after another. Again, many restrictions can be handled explicitly in *LL*, 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. Hence, LL_{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 *LL*.

As one can expect, as 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 hours, then we allow loads being delivered late up to five hours. 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.

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:

$$X_{ijk} = \begin{cases} 1, \text{if truck } i \text{ is assigned to load } j \text{ then load } k, \text{ and } TL(i,j) = TL(i,k) = LL(j,k) = 1 \\ 0, \text{otherwise} \end{cases}$$

$$X_{i0k} = \begin{cases} 1, \text{if truck } i \text{ is assigned to first load } k,\ TL(i,k) = 1 \\ 0, \text{otherwise} \end{cases}$$

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

$$e\sum_{i\in I}\sum_{k\in K} d(\gamma_i,\alpha_k)X_{i0k} + e\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} d(\beta_j,\alpha_k)X_{ijk} + f\sum_{i\in I} X_{i00} + f\sum_{i\in I}\sum_{k\in K} w^0(\gamma_i,\alpha_k)\, X_{i0k} + f\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} w(\beta_j,\alpha_k)X_{ijk} + g\sum_{i\in I}\sum_{k\in K} l^0(\gamma_i,\alpha_k)X_{i0k} + g\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} l(\beta_j,\alpha_k)X_{ijk} + \sum_{j\in J} s_j Y_j \quad (1)$$

subject to

$$\sum_{i=1}^{I}\sum_{j=0}^{J} X_{ijk} + Y_k = 1 \qquad \forall\ k \text{ if } v(k) = 0 \qquad (2)$$

$$\sum_{j=0}^{J} X_{ijk} = 1 \qquad \forall\, i,k \text{ if } v(k) = i \qquad (3)$$

$$\sum_{k=0}^{K} X_{i0k} = 1 \qquad \forall\, i \qquad (4)$$

$$\sum_{k=0}^{K} X_{ijk} - \sum_{q=0}^{J} X_{iqj} = 0 \qquad \forall\, i, j, TL_{ij} = 1 \qquad (5)$$

$$X_{ijk} \in \{0,1\} \text{ and } Y_k \in \{0,1\} \qquad \forall\, i, j, k, TL_{ij} = 1, LL_{jk} = 1$$

We try to 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 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) make sure that a load that has already been assigned to a truck (say truck i, i.e., $v(j)=i$) has to be assigned to the same truck in the current planning horizon. This allows a certain load to be “pre-loaded” or “pre-assigned” to a certain truck although in many cases we can assume that this set is empty. In these cases, there are no constraints of type (3).

Constraints (4) and (5) cover the possibility of trucks being assigned or not being assigned. Constraints (4) state that every truck is either assigned to a load (that is first for the truck) or not, which means it is not assigned to any loads at all. This means a solution of the problem does not require all available trucks. Constraints (5) state that if truck i serves load j' right after load j then load j is either the first load for the truck or is served after a load.

Note that the formulation of the load assignment problem could be significantly different (potentially more complicated) if the time-related constraints are handled explicitly in the model. The offline handling of time constraints is considered 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 requirements is simple 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 a similar 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-complete. Although we use the model to gain managerial insights on the timing of decisions and to 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 of course 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; information about loads (their pick-up and delivery locations and times) becomes available desirably some time before their corresponding pick-up times. We also consider the timing of decisions explicitly: When does the planner have to make decisions, i.e., fix the assignment of the 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 τ) refers to the time window during which the planner knows about the load's availability. In other words, τ represents the number of time periods in advance that the planner knows about the availability of the loads. For example, when τ is set to 3 days, then all the loads on a certain day (more specifically all the loads that have departure time on a certain day), say Thursday, are known by the scheduler 3 days in advance, on Monday. Similarly loads on Friday are known by the planner on Tuesday, and so on. 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 τ.

We use τ and T to emulate the dynamic availability of load information and its impact on the preplanning activity and the decisions themselves. Figure 1 depicts an example setting with ten loads to be served over a six-day period. In the following, we specify different policies characterized by the two time window parameters, τ and T, and illustrate them using the example (These settings are the ones tested in the computational study explained in the next section):

[τ=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. For the example in Figure 1, we have information about loads 1 and 2 on day 1, and we make/fix assignment decisions for both on day 1. On day 2, we know and have to fix loads 3 and 4, and so on. Note that this setting corresponds to *pure dispatching* (*wait till the last minute*) policy outlined earlier.

[τ=3, T=1], and [τ=5, T =1]

For the first policy, the planner has load availability information 3 days in advance but have to make assignment decisions 1 day in advance. For Figure 1, we have information about loads 1-6 on day 1 (and consider them in every planning opportunity), but have to fix assignment decisions for loads 1 and 2 on day 1. Next day (day 2), we know the availability of loads 7 and 8, and have to assign loads 3 and 4. The same logic follows for the subsequent loads and days. The second policy, [τ = 5, T = 1], is the same, except that the load assignment decisions are known and considered 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*."

[τ =3, T=3], [τ=5, T=3], [τ=5, T=5]

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. For the example in Figure 1, we know about loads 1-6 on day 1, and fix their assignment decisions on the same day. Considering that we must have completed the assignments of loads 1-4 before the current planning horizon (recall T=3), we are effectively making decisions on loads 5 and 6. On day 2, we make the decisions for loads 7 and 8, and so on. The other two policies can be similarly characterized. We should note, though, that the policy that leaves some flexibility in preplanning is [τ=5, T=3], in which the planner does not have to fix the decisions for 2 extra days even though he/she has and considers the load information in finding the solutions.

Note that 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. That is, 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. We first introduce the following:

$L(t_1, t_2)$: set of loads with pick-up times between t_1 and t_2, or $L(t_1, t_2) = \{j \text{ in } J: t_1 < a_j < t_2\}$

We now give the details of the dynamic implementation:

Step 1: Set the load knowledge window (τ), 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, …, H, e.g., t =1,…,20 days), create an instance of the load assignment problem by populating it with the corresponding data and fix the portion of solution obtained. That is:

Step2.1: Define *LAP*(t) to be the load assignment problem with loads that have pick-up times from t+1 to t+τ.

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 the reasons that will become clear in the experimental study, we report two total costs values: One for all the loads in the whole planning horizon, and the other for loads in the middle of the overall planning horizon (In the computational study, the first one is for all the loads in 20-day planning horizon, and the second is calculated for the loads in $L(5,15)$).

A benchmark 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. It is unrealistic since the information about the loads is not available at once under the current setting. However this benchmark solution is useful to see what the best possible total costs would be if perfect information about all the loads were available. We call this as the *static benchmark implementation*. For the example in Figure 1, the static implementation would assume that the planner knows about all of the loads (1 through 10), and all their characteristics (costs, locations and times) on day 1. Same day, its solution makes the assignment decisions for all the loads.

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 τ=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 will assume that those loads departing on day 2 and day 3 are decided on day 1. We will also assume that on day 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 extend the planning horizon into 20 days and increase the number of loads to 50. We also compare the models in two ways; we consider performance during the overall 20-day planning horizon, and we consider the performance only 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 [τ, T], 6 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. In choosing network size as one of the factors, we hope to use the 10-city network to represent a less dense region while the 50-city network represents a more dense and populated region.

(3) The maximum allowed lateness (u). We use two levels of this factor: 0 and 5 hours. 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 (see Table 1 for summary of these factors and their levels).

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 and y coordinates) of the cities are determined randomly with a discrete uniform distribution from a Cartesian coordinate system of 20 by 20 (20x20 grid). The set of cities produces the underlying transportation network.
- We generate 50 loads that are serviced among the cities generated previously. Origin and destination cities for loads are determined randomly with a discrete uniform distribution from those cities. That is, each load has an equal chance of (1/*K*) to get an origin city or a destination city between 1 to *K*.
- The initial locations of trucks are randomly generated with a discrete uniform distribution from the 20x20 grid. This location is considered the driver domicile. Hence, each truck has an equal chance to get a domicile city between 1 to *K*.
- 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 minutes. The overall planning horizon is fixed to be 20 days.
- The distance between cities is calculated using the Euclidean distance function.

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.

Tables 2 (for u=0) and 3 (for u=5 hours) 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. Tables 4 and 5 presents the number of problems (out of 20) for which each policy produces the lowest costs, for u=0 and u=5 hours, respectively. From these tables, we see that the static implementation always gives the lowest costs for 20 days. This is expected, since the static implementation exploits all opportunities for optimization by considering all the loads simultaneously. The results in the tables also show that this is not the case any more when the total costs for the loads during mid 10 days are considered. When we compare the mid 10 days' average performances (days 5-15), we see that in a reasonable number of problems, a select group of preplanning or dispatching with look-ahead policies performs better than the "static optimal." This is also expected since a 20-day optimal solution might have a sub-optimal 10-day period in the middle of the overall horizon. However, the average cost of static optimal schedule over mid 10-day period is still lower than the costs of more realistic policies.

We also see that with some level of lateness allowed (from 0 to 5 hours), 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 Figures 2-5 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. To see if these average performance levels are stable, we list the number of problems that each policy finds the lowest costs for in Tables 4 and 5. First number in each cell in these tables includes the static optimal as an alternative; hence it is the best for all 20-day performance under every setting. A specific dispatching policy with look-ahead [5,1] 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. Without the static optimal values as possibility, as shown as second values in each cell in the same tables, the relative benefit of waiting till the last minute and considering as much information as possible is even more visible. Similar observations can be made when limited lateness is allowed (see Table 5 for u=5 hours).

The results in these tables also 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 performance for both network sizes. The performance of the pure preplanning policy with τ=5 and T=5 also does not produce much better results than the worst two. This can be explained by recalling that this model is essentially is 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 ([τ=3, T=1], and [τ=5, T=1]) and the flexible preplanning policy ([τ=5, T=3]) are significantly better than the performances of pure dispatching (no look-ahead, [τ=1, T=1]) or pure preplanning ([τ=3, T=3], and [τ=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 can be 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 τ 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 model [3,3] also has a worse performance than the model [3,1]. In other words, if you have load information n days in advance where n>1, it is wise not to make all load assignment decisions immediately. That is why maximum delay in fixing decisions (dispatching with look-ahead) is producing overall best results, which are very close to the static optimum. Having at least partial load information (3-5-day look-ahead, instead of all 20 days) in conjunction with maximum delay performs over time as if you had the perfect information for all 20 days at once.

More interestingly, the preplanning policy [5,3] performs as well as these star performers and the static implementation. There is practically 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. That is, you do not need to wait till the last minute to fix your decisions to exploit all optimization opportunities, you can practically preplan for your loads as long as you have enough time to consider the effects of preplanning on other loads. The average differences between the policies and the static implementation optimum are plotted in Figures 2 and 3 for 20-day horizon and mid 10-day horizon, respectively, when no lateness is allowed. Similarly, Figures 4 and 5 plot average differences between the policy performances and the static optimum for u=5 hours. The distinction between the two groups of policies mentioned earlier is even more visible. Another result is that the policy with [τ=5, T=1] is not statistically different from the policy with [τ=3, T=1]. This means to be able to take advantage of load information, you do not need 5-day advance notice, even 3-day would be enough. Note that both policies implement dispatching with look-ahead.

7. Managerial Insights and Conclusions

The computational results show that an appropriate delay in fixing the assignment decisions till the last minute by considering all available information along the way is an outstanding performer, as expected. Only then, can we know the effects of current decisions on all the future loads. However, there are preplanning policies that work as well as dispatching with look-ahead when its parameters are appropriately set. The results also lead to two significant observations in terms of the overall performance of the preplanning policies: (1) Having advance load information and using it 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, if not worse, as the pure dispatching policy (without look-ahead, as if early information is not beneficial). (2) On the other hand, if preplanning policy has some flexibility in delaying the decisions and in considering the effect of preplanning on 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 as long as it is patched with some built-in look-ahead feature.

Moreover, the planner can use this minimum cost differential between preplanning and dispatching with look-ahead to provide incentives to the shippers so that they provide 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.

Finally, the planner does not necessarily need to entice customers to plan too early, a reasonable advance notice that would work both the shippers and the carrier provides substantially better result than not having advance information at all and having to do pure dispatching.

Our findings show that regardless of the network size, if we have a choice, we can benefit from knowing about loads earlier and we should exploit the opportunities for improvement by postponing planning until the last minute. By knowing the relative cost differences between making decisions earlier versus later, and by having a large load knowledge window, planners can trade-off the benefits and costs of early decision making and gather more information about the future loads prior to fixing delivery assignments.

For future research, we can refine and make the simple policies outlined here more sophisticated. One such modification could be the addition of a “frequency” parameter. In our current set of policies, each is re-evaluated every day with an execution of the LAP model with the corresponding set of known loads. However, one can save from computational burden by solving 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 workweeks), 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 for uncertainty in dynamic load assignment problems. In an effort to more fully examine related factors along with stochasticity, we also anticipate using simulation in the future.

References

1. Bertsimas, Dimitris, and Garrett van Ryzin. (1991). "Stochastic and dynamic vehicle routing in Euclidean plane," *Operations Research*, 39(4), 601-615.
2. Gallego, Guillermo and Ozalp Ozer. (2001). "Integrating Replenishment Decision With Advance Demand Information," *Management Science*, 47(10), 1344-1360.
3. J.B. Hunt Transport, Inc. (JBHT) (http://www.jbhunt.com)
4. Keskinocak, Pinar, and Sridhar Tayur. (1998). "Scheduling of Time Shared Aircraft," *Transportation Science*, 32(3), 277-294.
5. Lu, Yingdong, Jing-Sheng Song and David D. Yao. (2001). "Order Fill Rate, Leadtime Variability, and Advance Demand Information in an Assemble-to-Order System," *Technical Report*, Columbia University, Department of Industrial Engineering.

6. Malandraki, Chryssi. (1989). “Time dependent vehicle routing problems: Formulations, solution algorithms and computational experiments,” *Ph.D Dissertation*, Department of Civil Engineering, Northwestern University.
7. Picard, Jean-Claude and Maurice Queyranne. (1978). “The time dependent traveling salesman problem and its application to the tardiness problem in one machine scheduling, “ *Operations Research*, 26(2), 86-110.
8. Powell, Warren B. (1995). “Stochastic and Dynamic Networks and Routing,” In M.O Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (editors), *Network Routing*. Elsevier Science Publishers, Amsterdam, The Netherlands, Vol.8, 141-295.
9. Powell, Warren B. (1996). “A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers,” *Transportation Science*, 30 (3), 195-219.
10. Powell, Warren B., Michael T. Towns, and Arun Marar. (2000).“On the Value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Noncompliance,” *Transportation Science*, 34 (1), 67-85.
11. Psaraftis, Harilaos N. (1988). “Dynamic vehicle routing problems,” In B. Golden and A. Assad (editors), *Vehicle Routing: Methods and Studies*, Amsterdam, North Holland, 223-248.
12. Regan, Amelia C., Hani S. Mahmassani and Patrick Jaillet. (1998). “Evaluation of Dynamic Fleet Management Systems: Simulation Framework,” *Transportation Research Record*, 1645, 176-184.
13. Wang, Xiubin and Amelia C. Regan. (2002). “Local Truckload Pickup and Delivery with Hard Time Window Constraints,” *Transportation Research*, Part B, Methodological, 36 (2), 97-112.
14. Yang, Jian, Patrick Jaillet and Hani S. Mahmassani. (1999). “Online Algorithms for Truck Fleet Assignment and Scheduling Under Real Time Information,” *Transportation Research Record*, 1667, 107-113.

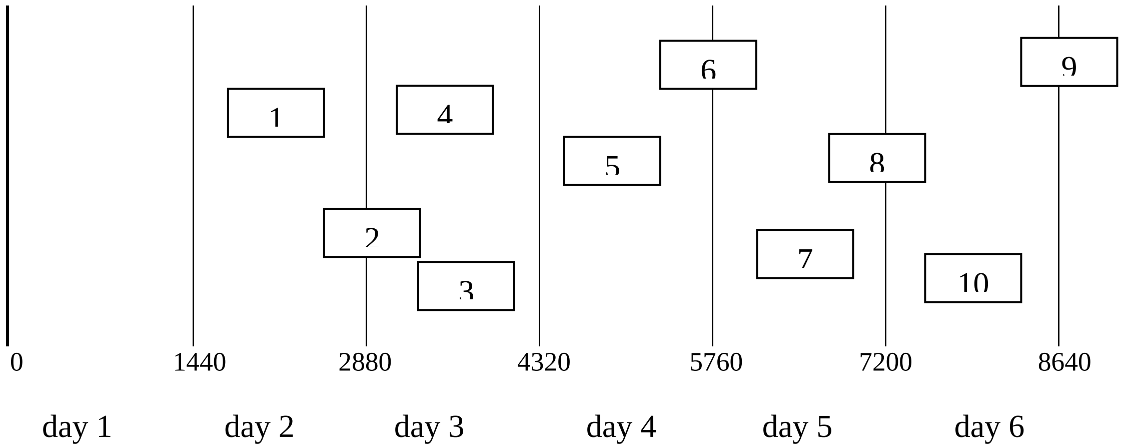

Figure 1: Example problem with 6-day planning horizon and 10 loads

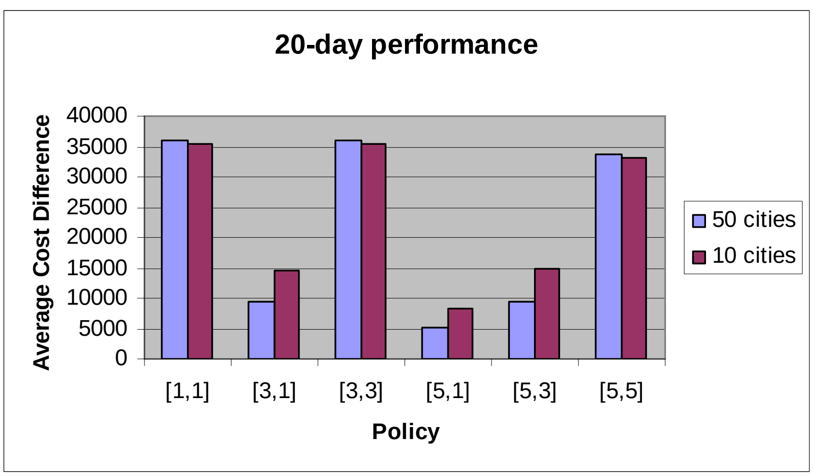

Figure 2: Difference between the costs of each policy and the static optimal for u=0 hrs (Differences in all 20-day costs are calculated for each problem and then averaged over all 20 problems).

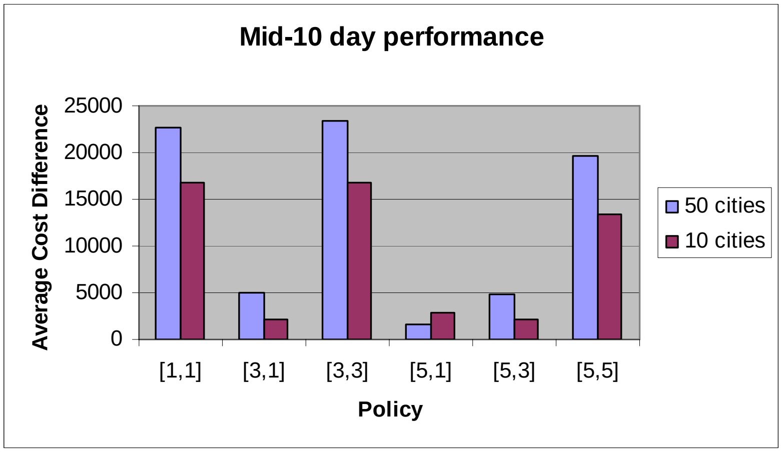

Figure 3: Difference between the costs of each policy and the static optimal for *u*=0 hrs (Differences in mid-10-day costs are calculated for each problem and then averaged over all 20 problems).

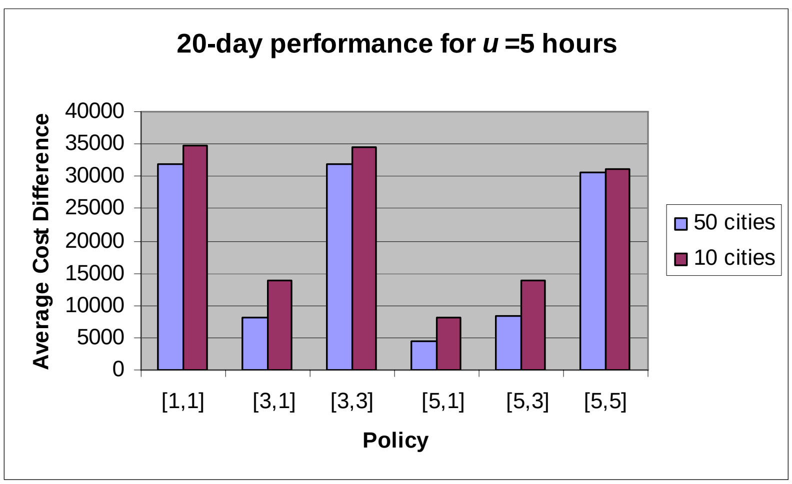

Figure 4: Difference between the costs of each policy and the static optimal for *u*=5 hrs (Differences in all 20-day costs are calculated for each problem and then averaged over all 20 problems).

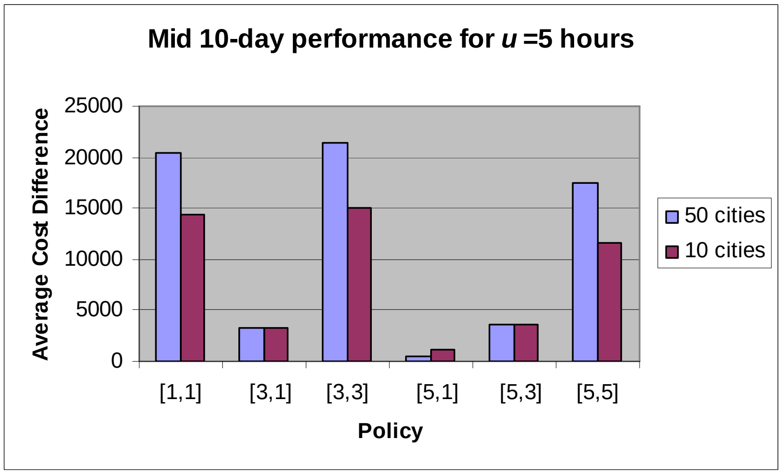

Figure 5: Difference between the costs of each policy and the static optimal for *u*=5 hrs (Differences in mid-10-day costs are calculated for each problem and then averaged over all 20 problems).

Table 1: Experimental Factors

Factors	Levels	No of Levels
[τ , T]	(1,1),(3,1),(3,3) (5,1),(5,3),(5,5)	6
Network Size	50-city network, 10-city network	2
Maximum lateness allowed	0 hours, 5 hours	2

Table 2: Total costs for all policies averaged across 20 problems in each setting for *u*=0 (* [3,1], [5,1] and [5,3] are significantly different and better than [1,1], [3,3] and [5,5])

	Network Size 50 cities		Network Size 10 cities	
Horizon->	*10 days*	*20 days*	*10 days*	*20 days*
[1,1]	79470	219985	74271	218384
[3,1] *	61791	193445	59632	197507
[3,3]	80166	220015	74187	218459
[5,1] *	58508	189016	60329	191055
[5,3] *	61722	193286	59591	197722
[5,5]	76460	217763	70839	216083
Static	56833	183955	57417	182902

Table 3: Total costs for all policies averaged across 20 problems in each setting for *u*=5 hrs (* [3,1], [5,1] and [5,3] are significantly different and better than [1,1], [3,3] and [5,5])

	Network Size 50 cities		Network Size 10 cities	
Horizon->	*10 days*	*20 days*	*10 days*	*20 days*
[1,1]	75007	209581	70548	211021
[3,1] *	57870	185729	59479	190093
[3,3]	75949	209704	71159	210742
[5,1] *	55163	182249	57234	184225
[5,3] *	58240	185982	59811	190002
[5,5]	72041	208316	67807	207341
Static	54597	177722	56142	176228

Table 4: Number of problems for which each policy produces the lowest total costs for *u*=0 (First number in each cell shows the statistic when the static optimal for 20 days is counted, the second number without counting the static optimal)

	Network Size 50 cities				Network Size 10 cities			
Horizon->	*10 days*		*20 days*		*10 days*		*20 days*	
[1,1]	0	0	0	0	0	0	0	0
[3,1]	0	4	0	1	0	8	0	2
[3,3]	0	0	0	0	0	2	0	0
[5,1]	6	16	0	19	8	10	0	19
[5,3]	4	5	0	1	3	10	0	2
[5,5]	0	0	0	0	0	2	0	0
Static	11		20		10		20	

Table 5: Number of problems for which each policy produces the lowest total costs for *u*=5 hrs (First number in each cell shows the statistic when the static optimal for 20 days is counted, the second number without counting the static optimal)

	Network Size 50 cities				Network Size 10 cities			
Horizon->	*10 days*		*20 days*		*10 days*		*20 days*	
[1,1]	2	2	0	0	1	2	0	0
[3,1]	1	3	0	3	4	6	0	2
[3,3]	1	1	0	0	1	1	0	0
[5,1]	5	14	0	17	8	11	0	18
[5,3]	3	6	0	3	4	7	0	2
[5,5]	2	2	0	1	1	1	0	0
Static	10		20		6		20	