Waiting Strategies for Anticipating Service Requests from Known Customer Locations

Barrett W. Thomas
Department of Management Sciences, University of Iowa, 108 John Pappajohn Business Building, Iowa City, IA 52242-1994, USA, barrett-thomas@uiowa.edu

This paper considers a dynamic and stochastic routing problem in which information about customer locations and probabilistic information about future service requests are used to maximize the expected number of customers served by a single uncapacitated vehicle. The problem is modeled as a Markov decision process and analytical results on the structure of the optimal policy are derived. For the case of a single dynamic customer, we completely characterize the optimal policy. Using the analytical results, we propose a real-time heuristic and demonstrate its effectiveness compared to a series of other intuitively appealing heuristics. We also use the heuristic to determine the value knowing both customer locations and probabilistic information about future service requests.

1. Introduction

Wireless communication technologies have become universal in today’s society. These technologies offer great opportunities to transportation companies operating in a dynamic environment. Companies can easily afford to equip every driver with a communication device and then communicate with the driver in real-time. In addition, the availability of relatively inexpensive database software and computer memory has meant that companies can capture, store, and analyze customer data at all levels of their delivery systems. For example, United Parcel Service has spent billions on information technology to capture data from all transactions (Ross and Beath, 2002). Consequently, delivery companies, particularly package express and less-than-truckload (LTL) companies, know where their customers are located and can derive probability distributions on which day of the week these customers request service and even the time of day at which they will request service. This information coupled with wireless technologies enables companies to anticipate and immediately respond to their customers’ service requests (see U.S. DOT, FHWA 2001). In this paper, we study a dynamic and stochastic routing problem which incorporates knowledge about customer locations and about probability distributions on customer requests.

The problem considers a single, uncapacitated vehicle which serves a set of known customer locations. An uncapacitated vehicle reflects the situation in package express and can be used to
approximate the environment in LTL. The vehicle begins service from a known starting point, which might represent the point where the vehicle ends its delivery service and begins its pickup service. At the starting point, the vehicle driver is aware of a subset of customers who have placed advance requests for pickup service. These advance-request customers are routed with the last customer site visited being a depot. We assume that the vehicle must arrive back to the depot at some known time. This time accounts for both work rules limiting a driver’s day [for example, see U.S. Department of Transportation Federal Motor Carrier Safety Administration (2005)] as well as a cut-off time required by the pickup and delivery company so that it can meet deadlines for the overnight linehaul operations. In addition to the advance-request customers is another subset of customers, late-request customers, who are customers who have not requested service by the start of the route, but who may do so while the vehicle is en route. Knowing a probability distribution on the time at which each late-request customer is likely to call, the vehicle driver can choose to wait at his or her current location or move onto the next customer as specified by the route. Late-request customers are not required to request service, but if one does, the driver has the option of inserting the requesting customer into the pre-existing route provided that the time horizon is not violated.

The objective is to determine a policy for selecting the next customer to visit as well as where and how long to wait such that the expected number of late-request customers who are served is maximized. This objective reflects the fact that, in all segments of the trucking industry, wages and benefits are the dominant cost (ATA Economics & Statistical Analysis Department, 1999). Perhaps more importantly, companies in the multiple pickup business, such as package express and LTL companies, tend to be unionized (The Tioga Group, 2003). As a result of these unionized drivers’ wage and benefit packages, the cost of these drivers can be viewed more as fixed than as variable. Consequently, companies want to maximize the utilization of each driver’s time which occurs when each driver serves as many customers as possible.

The focus of the paper will be on explicitly determining where a vehicle should wait in anticipation of service requests from late-request customers. As discussed in the next section, a body of emerging literature shows that both implicit and explicit waiting strategies can greatly improve performance with regard to a number of objectives. In this paper, we demonstrate that the choice of locations at which to wait can be further improved if we know where our late-request customers are located even if we do not know whether or not they will require service.

This paper is outlined as follows. In Section 2, we discuss related literature. Section 3 describes a finite-horizon Markov decision process model for the problem, and Section 4 presents the preliminary results. In Section 5, we examine the structure of the optimal policy. For the general case,
Section 5.1 identifies situations in which the optimal decision is to choose to wait as well situations were the optimal decision is never to choose to wait. For the single late-request customer case, Section 5.2 characterizes the optimal policy.

Section 6 introduces real-time heuristics for the problem. One of these heuristics is based on results in the previous section. Section 7 presents empirical comparisons of the algorithms. Finally, Section 8 summarizes the results and offers directions for future research.

2. Literature Review

The problem discussed in this paper can be categorized as a stochastic and dynamic routing problem. The problems are stochastic in that information about the customer service requests can be described by a random variable with a known probability distribution. The problems are dynamic in the sense that customer service requests occur over the problem horizon. This review also distinguishes between implicit and explicit waiting strategies. Implicit waiting strategies are those in which the vehicle is allowed to wait, but when and where to wait are typically set by some a priori rule rather than as the result of incorporating information about future requests into the model. Explicit waiting strategies incorporate this information about future requests into the model.

Early work on dynamic and stochastic routing routing problems focuses on the dynamic routing and dispatching problem (DRDP). In DRDPs, stochastic information about future requests is typically ignored and the dynamic nature of the problem is not acknowledge in the solution approach. Rather, the research presented in these papers can be considered reactive in that requests for service are considered only when they occur and no effort is made to anticipate that future requests will occur. Overviews of these problems can be found in Powell, Jaillet, and Odoni (1995) and Psaraftis (1988, 1995).

Work that follows the early DRDP work also does not exploit probabilistic information about future service requests, but does implicitly account for future arrivals. Yet, this research demonstrates that even implicitly accounting for future requests leads to improved performance. Kilby, Prosser, and Shaw (1998) demonstrate the advantage of implicitly anticipating future service requests by waiting at the last advance-request customer. Gendreau et al. (1999) and Ichoua, Gendreau, and Potvin (2000) explore diversionary routing in the express courier industry, implicitly acknowledging dynamic customer arrivals by allowing the vehicle to wait at its current location if waiting time would be incurred by moving to the next routing customer. Larsen, Madsen, and Solomon (2002) consider a problem in which some service requests are known in advance of the
start of service, but the locations of yet to arrive service requests are unknown. The authors also consider the objective of minimizing expected travel time rather than expected customer waiting time. They then empirically test a series of heuristics based on heuristics presented in (Bertsimas and Ryzin, 1991). Larsen, Madsen, and Solomon (2004) extend their previous work to consider customers with time windows and modify the objective to minimize lateness with regard to customer time windows. Additionally, they introduce intermediate locations at which the vehicle can wait for new requests to arrive. The purpose of this idling point is to allow the vehicle to wait in a location which may be well-suited for serving newly arrived requests for service. We note that these intermediate locations are chosen by intuition rather than based on an exploitation of information about future demand requests.

With waiting shown to be a useful strategy for handling dynamic customer arrivals, recent work incorporates explicit waiting strategies. Branke et al. (2005) consider a problem in which some customer service requests as well as their locations are known in advance, and assume that customer service requests that arrive over the problem horizon are uniformly distributed in a bounded region. The objective is to maximize the probability of inserting a customer who calls during the problem horizon. The authors show that, for a single late-request customer, the optimal routed customer at which to wait is the customer that maximizes the area covered by the vehicle. For multiple late request customers, the authors test a series of simple deterministic heuristics as well as an evolutionary algorithm. Mitrovic-Minic and Laporte (2004) and Mitrovic-Minic, Krishnamurti, and Laporte (2004) also explicitly discuss heuristic waiting strategies for the dynamic pickup and delivery problem with time windows (DPDPTW). Like the DTRP, the locations of customers requesting service are not known until a service request is made. As in (Larsen et al., 2004, Branke et al., 2005), the heuristic waiting strategies for the DPDPTW focus on strategies that distribute available waiting time to various portions of the route. In (Mitrovic-Minic and Laporte, 2004), the authors find that the best heuristic distributes the available waiting time throughout the already routed customers or clusters of routed customers. Research in (Mitrovic-Minic et al., 2004) extends the earlier DPDPTW work by introducing a double-horizon heuristic. In the short term, the goal of the heuristic is to minimize route length. For the long term, the heuristic preserves flexibility so that future service requests can more easily be accommodated.

Introducing the dynamic traveling salesman problem (DTSP), Psaraftis (1988) was the first to explicitly incorporate probability distributions on future demand over time to improve decision-making in dynamic environments. In the DTSP, customer service requests arrive to known locations according to a Poisson process with rate $\lambda$. For the objective of minimizing the average waiting
time of a request, Psaraftis conjectures that when $\lambda$ is small, the optimal solution resembles a 1-median problem. Modifying the DTSP such that demand arrives uniformly to a bounded region rather than to known points, Bertsimas and Van Ryzin (1991) introduce the single vehicle dynamic traveling repairman problem (DTRP), and Bertsimas and Van Ryzin (1993) extend the problem to the multi-vehicle case. For the light traffic case, the authors show that the optimal policy locates the vehicle at a median point on the graph, in essence anticipating future demand. Because the arrival rate is homogeneous, this optimal policy does not dynamically change over the problem horizon. Papastavrou (1996) introduces routing policies for the DTRP that are stable in both light and heavy traffic. Swihart and Papastavrou (1999) extend the DTRP to the dynamic pick-up and delivery problem. Gendreau, Laporte, and Semet (2001) present a dynamic model and parallel tabu search solution approach for a real-time ambulance relocation problem related to the DTRP.

In a vehicle dispatching context, Powell et al. (1988) introduce a truckload dispatching problem, and Powell (1996) provides formulations, solution methods, and numerical results. In these papers, future demand forecasts are used to determine which loads should be assigned to what vehicles in a truckload environment to account for forecasted capacity needs in the next period. In the event that no demand exists in a particular vehicle’s immediate region, the authors use a distribution on service requests over time to determine when, if at all, the vehicle should be moved to a new region.

Ichoua, Gendreau, and Potvin (to appear) extend (Gendreau et al., 1999) to exploit probabilistic information about future arrivals. The heuristic allows a vehicle to wait in its current zone if the probability of a future request reaches a particular threshold. Other recent work has focused on maintaining a set of potential routes from which a best routing plan can be chosen based on the realized demand and forecasts of future demand. Hvattum, Løkkentangen, and Laporte (submitted for publication) introduce a hedging heuristic that uses sampling and common features of deterministic routes constructed for the sampled customers to build a plan for each time interval in the time horizon. According to the authors, this approach requires some computation and is not necessarily implementable in a real-time setting which requires quick decisions. Bent and Van Hentenryck (2004) also use sampling to construct a set of potential routes containing existing customers as well as possible future customers. Using these project plans to route the existing set of customers better allows for the accommodation of future requests. Waiting is allowed at a customer who has just been serviced. However, waiting is only implicitly incorporated. Similarly exploiting probabilistic information about future requests, van Hemert and La Poutré (2004) introduce the concept of fruitful regions. Fruitful regions are clusters of known customer locations that may
require service in the near future. In the fashion of (Bent and Van Hentenryck, 2004), potential schedules are created by sampling fruitful regions. Van Hemert and La Poutré then provide a evolutionary algorithm for determining when to move to one of the fruitful regions in anticipation of future service requests. The objective is to maximize the number of customers served. No discussion of waiting is included in the paper.

Most closely related to the work in this paper is Thomas and White III (2004). Thomas and White III introduce a problem in which the objective is to minimize cost with rewards (negative costs) being received for serving customers and costs incurred for traveling along arcs. As in this paper, Thomas and White III assume that customer locations are known in advance. In contrast to this work, however, decisions are made at each road intersection. Further, while stochastic information about future service requests is explicitly exploited, there is no explicit discussion with regard to waiting decisions. Rather, Thomas and White III show under what conditions the optimal policy is invariant to changes in problem data and empirically show the cost advantage of an anticipatory versus a reactive policy. Recently, Mes et al. (2006) use similar ideas for calculating opportunity costs in an agent-based routing scheme, and Bregman et al. (2006) explore the use of anticipatory information for real-time management of couriers in London.

Other work that explicitly anticipates changes in the problem environment can be found in the dynamic and stochastic shortest path literature, in which arc status rather than customer status is observed. Boyan and Mitzenmacher (2001) provide a polynomial-time algorithm for the problem of traversing a bus network in which bus-arrival times are distributed according to a probability distribution with the increasing failure rate property. Kim, Lewis, and White III (2005a) extend (Psaraftis and Tsitsiklis, 1993) to include real-time information. They model the problem in a manner similar to that presented in this paper and present results regarding optimal departure times from a depot and optimal routing policies. Kim, Lewis, and White III (2005b) present a state-space reduction technique that significantly improves computation time for the problem introduced in (Kim et al., 2005a). Thomas and White III (to appear) consider an analogous problem in which road congestion clears according to the increasing failure rate property.

This paper makes the following three contributions to the literature. First, this paper analytically identifies situations in which one always waits or never waits and also a situation in which it is known where to wait. Further, these analytical results are independent of the probability distributions on the time of future service requests. Second, we demonstrate that a real-time heuristic derived from the analytical results is superior to a series of other intuitively appealing heuristics when the percentage of late-request customers is moderate. Finally, using the heuristics, we empirically show the value of knowing the locations of customers who will request service.
3. Model Formulation

In this section, we present a formal dynamic programming formulation for the waiting strategies problem. Because of the exponential growth of the state space, the model’s use for finding optimal solutions is limited to moderately-sized datasets. However, a formal model offers a precise description of the problem as well as a framework from which to analytically evaluate the structure of the optimal policy.

Let \( G = (N, E) \) represent the underlying network where \( N \) is the set of customers and \( E = N \times N \) is the set of arcs connecting customers. For every \( n, n' \in N \), we assume that there exists an arc \((n, n') \in E\), including the arc \((n, n)\). That is, the network \( G \) is complete. Further, let \( t_{ij} \) be the amount of time required to traverse arc \((i, j) \in E\). We can think of the time \( t_{ij} \) as deterministic or as the mean travel time on arc \((i, j)\). We assume 1 time unit is required to traverse a self arc.

Let \( N_J \subseteq N \) be the set of advance-request customer, customers for whom, at the beginning of the time horizon, it is known that service is required. Let \( N_I \subseteq N \) be the set of late-request customers, customers such that, at the beginning of the time horizon, it is not known whether or not \( n \in N_I \) will require service. We assume that \( N_I \cup N_J = N \) and \( N_I \cap N_J = \emptyset \). The vehicle begins its route at node \( s \in N_J \), the start node, and must complete its journey at a node \( \gamma \in N_J \), the goal or end node. We note that \( s \neq \gamma \) necessarily.

A decision epoch occurs when the vehicle arrives at a node. Let \( t_k \) be a positive integer representing the time of the \( k^{th} \) decision epoch, and let \( n_k \in N \) be the position of the vehicle at time \( t_k \). Let \( U = \{0, 1, \ldots, T\} \) be the set of possible times when decisions are made, where \( T \) is the time after which no more decisions can be made. We can think of \( T \) as the time at which the vehicle must have returned to the depot \( \gamma \). Let the random variable \( K \) represent the total number of decisions and be such that, for a realization of \( K \), \( K < T \), where \( K \) is a realization of \( K \). Our assumptions imply that \( t_0 = 0, n_0 = s \), and \( K \leq T \) for every possible realization of \( K \) of \( K \).

The random variable \( Z^n(t) \) represent the status of customer \( n \in N \) at time \( t \). We let

\[
Z^n(t) = \begin{cases} 
0 & \text{if customer } n \text{ has not yet requested service at time } t \\
1 & \text{if customer } n \text{ has requested service, but no decision has been made as to whether or not to service the customer by time } t \\
2 & \text{if customer } n \text{ has requested service, has been approved for service, but has not been visited by time } t \\
3 & \text{if customer } n \text{ has been visited or if customer } n \text{ requested service and was rejected by time } t 
\end{cases}
\]

for \( n = 1, \ldots, |N| \), where \(|N|\) is the cardinality of \( N \). The mechanism used to determine approval for service will be discussed subsequently. Let \( Z(t) = \{Z^1(t), \ldots, Z^{|N|}(t)\} \), and denote a realization
of $Z(t)$ by $z$. Thus, $z \in H = \{0, 1, 2, 3\}^{[N]}$. Further, a realization of $Z^n(t)$, $z^n$, is not equal to 0 for any $n \in N$ for any time $t$, and at time $t = 0$, $z^n = 0$ for all $n \in N$.

For $z$, a realization of $Z(t)$ for some $t$, define $M(z) = \{n : n \in N, z^n = 2\}$. Then, $M(z)$ is the set of customers who have requested service and been approved by time $t$. We assume that the customers $m_i \in M(z), i = 1, \ldots, |M(z(t))|$, for any $t$ and where $|M(z(t))|$ is the cardinality of $M(z)$, are ordered according to some rule or algorithm. In other words, the sequence of customers in $M(z)$ defines a route of approved customers at time $t$. For a realization $z$ of $Z(t = 0)$, we assume that all customers $m \in M(z)$ can be visited and the end node reached by time $T$. The results in Section 5 make no assumption regarding the methodology used to determine the order of customers given by $M(z)$.

As will be discussed subsequently, this assumption means that late-request customers are then inserted into the route. While in reality drivers could dynamically modify the order in which customers are visited, such a priori customer routes are often used for operational convenience and are common in the dynamic vehicle routing literature (Ghiani et al., 2003) recently including Branke et al. (2005) and Mitrovic-Minic et al. (2004). As discussion of the action space makes clear, the effect of the a priori route assumption is that, at each state, the only decisions are to wait or to go onto the next customer.

For a realization $z$ of $Z(t)$ for some $t$, let $Q(z) = \{n : z^n = 1\}$. Thus, $Q(z)$ is the set of late-request customers who have requested service, but who have not yet been routed by time $t$. We denote the power set of $Q(z)$ as $2^{Q(z)}$. Then, let $X(q, z), q \in 2^{Q(z)}$, be a function such that:

\[
X(q, z) = \begin{cases} 
0 & \text{if } q \text{ cannot be feasibly inserted into the route } M(z) \\
1 & \text{if } q \text{ can be feasibly inserted into the route } M(z). 
\end{cases}
\]

We say that $q \in 2^{Q(z)}$ can be feasibly inserted somewhere into the current route $M(z)$ if every customer $i \in q$ can be inserted into $M(z)$ without the total length of the route exceeding $T$. Finally, define $\tilde{2}^{Q(z)} = \{q : q \in 2^{Q(z)}, X(q, z) = 1\}$. That is, $\tilde{2}^{Q(z)}$ is the set of subsets of newly arrived customer requests that can feasibly be inserted into the current route $M(z)$. We assume $\emptyset \in \tilde{2}^{Q(z)}$.

Define the state space of our problem as $\Omega = \{(n, t, z) : n \in N, t \in U, z \in H\}$. For each $(n, t, z) \in \Omega$, there exists a set of actions $A(n, t, z)$, which are available to the decision maker. At each decision epoch, the decision maker chooses an action from this set. For the problem described here, the decision maker simultaneously controls both insertion and movement with action selection. The insertion actions control which customers are selected for service and thus routed. At any decision epoch and state $(n, t, z), n \neq \gamma$, any set of customers $q \in 2^{Q(z)}$ can be chosen for insertion. Our assumption that $M(z)$ can be completed before time $T$ for a realization of $Z(t = 0)$ and the
construction of $2^Q(z)$ imply the vehicle must always reach the end node by time $T$. A movement action controls the motion of the vehicle. For each state $(n, t, z), n \neq \gamma$, the movement actions are to wait at the current location $n$ or to move onto the next node $m_{i+1} \in M(z)$, where $n = m_i$. We assume that, if a subset of customers $q$ is chosen for service, then $M(z)$ is instantaneously augmented and thus $m_{i+1} \in q$. Formally, $A(n, t, z) = \{n = m_i, m_{i+1}\} \times 2^Q(z)$. We assume that the selection of movement action $\gamma$ requires the selection of insertion action $\emptyset$. Further, we assume that, for any $t$ and $z$, the only action available in state $(\gamma, t, z)$ is the movement action $\gamma$ and the insertion action $\emptyset$. Subsequent discussion will outline the transitions resulting from action selection.

A decision rule at time $t$ is a function $\delta(\cdot, t, \cdot)$ that selects an available action. Thus, $\delta(n, t, z) \in A(n, t, z)$. A policy is a sequence of decision rules $\pi = \{\delta(n, t, z) : (n, t, z) \in \Omega\}$. We remark that $\delta(\cdot, t, \cdot)$ is implemented only if $t$ is a decision epoch.

For any $n \in N$, $t < T$, and customer status $z^n > 0$, the state dynamics are determined by the selected action. For $n$ such that $z^n = 2$, the selection of the action that includes movement to $m_{i+1}$, where $m_{i+1} = n \in M(z)$, changes the status of customer $n$ from 2 to 3. If the action selected is to wait at the current customer $n$, then the status of customer $n$ remains unchanged. If we choose to insert some number of service requesting customers into the current route, then the status of each $i \in q$, where $q \in 2^Q(t)$, transitions from status 1 to status 2. We assume that this transition is instantaneous.

We assume that $\{Z^i(t), t = 0, 1, \ldots, T\}$ and $\{Z^j(t), t = 0, 1, \ldots, T\}$ are independent Markov chains for $i \neq j$. Accordingly, for each $n \in N_I$, we assume that the state dynamics for $z^n(t) = 0$ are described by the one-step transition matrix:

$$R_n^{(t, t+1)} = \begin{bmatrix} 1 - \alpha_n^t & \alpha_n^t \\ 0 & 1 \end{bmatrix},$$

where $\alpha_n^t$ is the probability that, between time $t$ and $t + 1$, customer $n$ transitions from $z^n(t) = 0$ to $z^n(t + 1) = 1$. We note that, no decisions are made while a vehicle is in transit. A customer requesting service during transit cannot be inserted until the next decision epoch occurs.

Let $P(z' | t, z, t')$ be the probability of a transition occurring from $z$ at time $t$ to $z'$ at time $t' > t$. By our assumptions,

$$P(z' | t, z, t') = P(Z(t') = z' | Z(t) = z) = \prod_{n=1}^{[N_I]} P(z^n(t') = (z^n)' | z^n(t) = z^n),$$
where each term in the product satisfies an extension of Kolmogorov’s equations for the non-stationary case [see Kim, Lewis, and White (2005a)] such that

\[
P_n(t, t') = \left[1 - \alpha_n^t \alpha_n^{t'} \right] \times \left[1 - \alpha_n^{t+1} \alpha_n^{t'+1} \right] \times \cdots \times \left[1 - \alpha_n^{t' - 1} \alpha_n^{t' - 1} \right].
\]

Our reward structure is described by the function \( c(a), a \in A(n, t, z) \), whose value is the number of previously unrouted customers who have requested service and are chosen to have their service request fulfilled. Thus, for an action \( a \in A(n, t, z) \), \( c(a) = |q| \), where \(|q|\) is the cardinality of the set \( q \in 2^{Q(z)} \) identified by the insertion portion of the action. No cost is incurred as the result of travel time or waiting.

Let

\[
v^\pi(s, 0, z) = E^\pi \left[ \sum_{k=0}^K c(\delta(n_k, t_k, z_k)) \right].
\]

be the problem criterion for a policy \( \pi \). The problem objective is to find a policy \( \pi^* \), called an optimal policy, such that \( v^{\pi^*}(s, 0, z) \geq v^\pi(s, 0, z) \) for all policies \( \pi \) and for all \( z \in \{0, 1, 2, 3\} \).}

4. Preliminary Results and Optimality Equations

All of the results in this section can be found in (Puterman, 1994, Section 4.3). The optimality equation for \( t < T \) is

\[
v(n, t, z) = \max \{c(a) + \sum_{z'} P(z' \mid t, z, t') v(n', t', z') : a \in A(n, t, k)\},
\]

where \( t' = t + t_{n,n'} \) and \( n' \) is the node chosen by the movement portion of the action. The boundary condition is such that \( v(\gamma, t, z) = 0 \) for all \( t \) and \( z \). The solution of the optimality equation is unique, and \( v(n, t, z) = \max_\pi v^\pi(n, t, z) \), for all \( n, t, \) and \( z \). A necessary and sufficient condition for \( \pi^* \) to be optimal is that it is composed of decision rules that cause the maximum in the optimality equation to hold. We refer to \( v(n, t, z) \) as the reward or cost-to-go function.

5. Structure of the Optimal Policy

In this section, we develop results related to the structure of the optimal policy. We emphasize that all of the results are independent of the distributions on service requests from late-request customers. The results for the general problem identify a situation in which choosing to travel to the goal node will not be optimal and a situation in which the decision will never be to wait. We also characterize the optimal policy for the case where there is only one late-request customer. This characterization will guide the development of a real-time heuristic for this problem.
5.1. General Structural Results

Our first general result shows that the action of waiting at the current node is always optimal if the next routed customer is the goal node and there is still enough time to serve at least one unrouted customer if that customer requests service. Essentially, given our assumptions, the only action available at the goal state or depot, $\gamma$, is to stay and wait at $\gamma$ over the rest of the time horizon. Hence, no late-request customer who requests service can be visited, and the cost-to-go is 0. We simplify presentation of the result by defining the set $W(z) = \{n: n \in N_I, z^n = 0\}$. That is, $W(z)$ represents the set of late-request customers who have not requested service given the realization of customer status’ $z$.

**Theorem 1.** Assume for a given state $(n, t, z)$ that $M(z)$ is such that $m_{i+1} = \gamma$, $W(z) \neq \{\emptyset\}$, and $\bar{Q}(z) = \{\emptyset\}$. Then, the optimal movement action is to choose to wait at the current customer $n$.

**Proof.** The result follows directly from the optimality equations. Q.E.D.

Knowing where we would not wait is as important as knowing where and when we would wait. In both cases, we can eliminate actions from the set that needs to be considered. The next result shows that we do not need to wait at any customer location such that no currently unrouted customer would ever follow this customer on the tour. Before we formally present the theorem, it will be useful to define the set $Y(z) = Q(z) \cup W(z)$. That is, $Y(z)$ is the set of all customers who have not yet requested service or who have requested service, but have not yet been inserted into the route.

**Theorem 2.** For a state $(n, t, z)$, assume that $n$ is such that, for any $y \in Y(z)$ if $y$ would request service and then be chosen to be routed, $y$ would never follow $n$ on a route $M(z)$ for any realization of customer status $z$. Then, let the policy $\pi'$ be such that $\delta(n, t, z)$ includes the action to wait at customer $n$ and let the policy $\pi$ be such that $\delta'(n, t, z)$ includes the action to move to $m_{i+1} \in M(z)$. Assume that $\pi$ and $\pi'$ are optimal otherwise. Then, $v^\pi(n, t, z) \geq v^{\pi'}(n, t, z)$.

**Proof.** The result follows directly from the optimality equations. Q.E.D.

5.2. Special Case: One Customer who has Not Yet Called

In this section, we show that our total expected cost is maximized by waiting at the location that allows for the latest allowable departure time. To build intuition, consider a late-request customer with a stationary value of $\alpha$. Let the random variable $\mathcal{X}$ represent the time at which the late-request customer requests service. By construction of $R^{t,t+1}$, $\mathcal{X}$ is a geometric random variable.
Because the cumulative distribution function is nondecreasing, the policy that allows that longest possible waiting time has the greatest likelihood of being able to serve a late-request customer who requires service. We again emphasize that this result, like the first two, is independent of the arrival distribution of the late-request customer.

Recall \( W(z) = \{ w : w \in N_I, z^w = 0 \} \). Let \( \| W(z) \|_\gamma \) be the minimum amount of time required to travel from \( n \) to \( \gamma \) via any node \( w \in W(z) \). Finally, let \( \bar{t}_n = T - \| W(z) \|_\gamma \). To aid our discussion, we define \( t_j = \sum_{i=0}^j t_{ij} \), where the index of customers is given by \( M(z(0)) \) which is the initial route to serve the early-request customers. In other words, \( t_j \) is the earliest time that we can reach customer \( j \) given the initial routing of customers.

**Theorem 3.** Let \( N_I = \{ n \} \). That is, \( N_I \) is a singleton. Let \( \bar{m} = \arg \max_{m \in N_I} \{ \bar{t}_m \} \) and let \( \pi \) be a policy that travels to \( \bar{m} \) and waits until \( n \) requests service or until \( \bar{t}_m \). We assume that, if \( n \) requests service, \( n \) is serviced under the policy \( \pi \). Then, \( v^\pi (s, 0, z) \geq v^{\pi'} (s, 0, z) \), where \( \pi' \) is any policy \( \pi' \neq \pi \).

**Proof.** Suppose that the result does not hold. Then, there exists a policy \( \pi' \) such that \( v^{\pi'} (s, 0, z) > v^\pi (s, 0, z) \). For the policy \( \pi' \) to exist, there must be a situation such that the late-request customer \( n \) requests service, \( n \) cannot be feasibly serviced under policy \( \pi \), and \( n \) can be serviced under policy \( \pi' \). By the selection of \( \bar{m} \), the situation must be such that \( n \) requests service after time \( \bar{t}_m \). Otherwise, policy \( \pi \) could also service the late requesting customer \( n \). However, by our selection of \( \bar{m} \), there does not exist any early-request customer such that a call from \( n \) after time \( \bar{t}_m \) could feasibly be serviced. The existence of \( \pi' \) is then contradicted. Q.E.D.

The results in Theorems 2 and 3 require that the advance-request customers be visited in a defined a priori order.

6. **Heuristic Algorithms for Real-Time Decision Making**

Because the growth of the state space is exponential in the number of customers, exact methods can only solve small instances of the presented problem. Consequently, this section describes five heuristic algorithms for real-time routing when customer locations are known and with waiting decisions explicitly considered.

6.1. **Center-of-Gravity Longest Wait (LW) Heuristic**

The Center-of-Gravity (LW) heuristic is motivated by the analytical results in the previous section. In particular, Theorem 3 demonstrates that, in the one customer case, we want to wait at the location that maximizes that time at which we need to leave in order to finish the route by time \( T \). While we cannot prove a similar result for multiple customers, we make use of the result by
aggregating the customers. For this purpose, we use the center-of-gravity. Our selection of the center-of-gravity is motivated by Campbell (2006) in which the center-of-gravity was successfully used as a means of aggregating customers for the probabilistic traveling salesman problem.

In the general sense, the center-of-gravity problem is the problem of finding the point \((x^*, y^*) \in \mathbb{R}^2\) that minimizes the weighted square of the Euclidean distance between a set of customers located at points \((x_i, y_i), i = 1, \ldots, n\). Formally, we seek \((x^*, y^*)\) that minimizes the objective

\[
f(x, y) = \sum_{i=1}^{n} w_i[(x - x_i)^2 + (y - y_i)^2],
\]

where \(w_i\) is a weight associated with the point \((x_i, y_i)\). While straight-line distance rather than the weighted square objective is more typical in practice, the center-of-gravity is often used as an approximation because the solution is straightforward. The minimizing values are given by:

\[
x^* = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}, \quad y^* = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}.
\]

We compute a center-of-gravity for only those customers who can be feasibly inserted into the current route given the current vehicle location and current time. Additional discussion of the center-of-gravity problem can be found in Nahmias (2001, p. 609).

In our case, for each customer \(i \in N_I\) and \(z^i(t)\) we use one-step probability of calling \(\alpha_i\) to weight each customer’s location. That is, in equation 1, \(w_i = \alpha_i\) for each customer \(i \in N_I\). The center-of-gravity longest wait method then returns a single location for all of the customers who have not yet called. We can then find the routed customer whose location maximizes \(\bar{t}\) and aim to wait at that location. Because the problem is dynamic, we recompute the center-of-gravity at each decision epoch. Further, we incorporate the results of Theorems 1 and 2. Thus, if the successor of the current node \(n\) is the goal node \(\gamma\), we always choose to wait at \(n\) until time \(\bar{t}_n\). Likewise, if the current node \(n\) is such that no customer who requests service will be inserted after \(n\), then we always choose to proceed to the successor of \(n\).

While our analytical results guide our choice of the movement portion of our action selection, they do not give us much guidance for choosing the insertion portion. Consequently, we rely instead on strategies that are intuitively appealing. Our action selection proceeds as follows. At each decision epoch and location \(n\), we check to see if the current location \(n\) is the best waiting location given the updated center of gravity. If \(n\) is the best location, we then find the maximum subset of customers in \(Q(z(t))\) such that, if we routed this maximum subset of customers, we could still complete the route before time \(T\). If we would finish exactly at time \(T\), we route the customers and complete the route. If not, we have time to wait at \(n\) with the hopes of receiving new service requests and increasing the number of customers that will be served. If \(n\) is not the best waiting location, we
choose to move onto \( n \)'s successor. However, before moving onto \( n \)'s successor, we check to see if there are any customers in the set \( Q(z(t)) \) who, if they were routed, would be routed between \( n \) and \( n \)'s successor. We then route the maximizing subset of these customers such that the time deadline \( T \) is met and move onto the the next customer on the now updated route. The logic behind this insertion scheme is that, while we should not wait at the current location, we can minimize some insertion cost at this location and should take advantage of that opportunity.

6.2. Center-of-Gravity Longest Wait without Stochastic Information (LW No) Heuristic

While it seems intuitively appealing to include anticipatory information in the heuristic decisions and has been shown that anticipatory information is important when considering waiting locations for only one late-request customer, the extrapolation from one-to-many late-request customers may gain more from knowing customer locations than from the stochastic information about future requests. Hence, we also test a Center-of-Gravity Longest Wait Heuristic without Stochastic Information (LW No) heuristic. This heuristic is exactly like the above described LW heuristic except that all customers are given equal weight in the computation of the center-of-gravity. That is, for the purposes of computing the center-of-gravity, we assume \( \alpha_i = 1 \) for all \( i \). Comparing the LW No heuristic to the LW heuristic also indicates the value of including stochastic information versus location information about future requests into our heuristic.

6.3. Center-of-Gravity Closest (CL) Heuristic

The Center-of-Gravity Closest (CL) Heuristic is based on the idea that minimizing insertion cost and thus minimizing travel time gives us the ability to insert the most customers. The COGC heuristic is implemented in a manner similar to the LW heuristic except that, instead of choosing the customer that maximizes \( \bar{t} \), we choose to wait at the customer closest to the center of gravity as computed in equation 1.

6.4. Wait at Start (WAS) Heuristic

We also test a wait-at-start (WAS) heuristic because of its intuitive appeal. The intuition is that, in the presence of customers who have not called, it may be best to wait at the starting point so as to minimize the cost of inserting customers and thus allowing the greatest number of insertions. Theorem 3 contradicts this intuition for the one-customer case. However, the simplicity of the heuristic and the resulting ease of implementation suggests that the heuristic may be important to consider.

To implement the WAS heuristic, at each time unit, we check to see that the vehicle can complete the tour of currently routed customers by time \( T \). If we can, we then select that maximum feasible
set of unrouted customers who have requested service. If by waiting one more time unit at the start, this set of customers can be inserted and the tour feasibly completed by time $T$, we wait one more time unit. Otherwise, the vehicle completes the tour.

### 6.5. Distribute Available Waiting Time (DW) Heuristic

The distribute-available-waiting-time (DW) heuristic is based on heuristics that Larsen et al. (2004), Branke et al. (2005), and Mitrovic-Minic and Laporte (2004) show perform well for problems in which customer locations are not known before service requests are realized. As its name implies, the DW heuristic distributes available waiting time equally across routed customers. The heuristic is dynamic in the sense that the available waiting time is recomputed at each decision epoch in light of new customer service requests and insertions.

Customers are inserted in two cases. First, if the vehicle has waited at a location for that location’s allotment of the waiting time, then any late-request customers for whom insertion cost is minimized by inserting the customers between the current location and the next location to be visited are inserted. Second, if at time $t$ the set of late-request customers requesting service is such that a maximum set can only be serviced by inserting the customers at time $t$ and finishing the route, the insertion action of inserting the maximal subset is taken.

### 7. Computational Experiments

This section provides empirical comparisons of the proposed heuristics. In addition, we detail the implementation of the heuristic and discuss the datasets used for testing.

#### 7.1. Implementation and Data Set Generation

The heuristics discussed in the previous section were implemented in C++ using BOOST Graph Library objects (Siek et al., 2001) and an implicit state-space graph representation.

Because the problem in this paper has not been previously explored, data set generation was required. Eight existing datasets were used as the basis for testing. Three of these datasets were generated from 50-customers instances of Solomon’s C101, C102, and RC102 datasets (Solomon, 1987). The other five sets were the 40-customers sets proposed in Dumas et al. (1995). For all eight sets, the time windows were simply ignored. For each set of customers, we then consider instances having only three late-request customers, another having 25% of the customers as late request, and the third instance having 50% of the customers as late request. Larsen et al. (2002) note that 25% and 50% percentages for late-request customers correspond to the degrees of dynamism typically found in LTL and package-express problems.
The probabilities that a late-request customer would request service, $\alpha$, were chosen randomly such that the probability that a customer would call over the time horizon of the problem ranged from 0.10 to 0.75. For each of the datasets, we also tested two different amounts of waiting time. In the first instance, we first used a savings heuristic to insert the late-request customers into the initial tour. The total route time of this updated tour was then multiplied by 1.33. In the second set of instances, we multiplied by 1.66 instead. We thus refer to the instances as having either 33\% or 66\% waiting time.

To construct the initial route of advance-request customers, the first customer in the dataset was set to the start position and the last to the goal nodes position. We then routed the remaining advance-request customers using shortest path through the set of customers. The path was determined by a greedy randomized adaptive search procedure (GRASP) with a savings insertion criterion with a candidate list of size three (Feo and Resende, 1995) and a post-construction greedy local improvement scheme using a 1-shift neighborhood. The choice of the shortest path through a set of customers as the routing methodology was based on ease of implementation and the fact that the purpose of this paper was on dynamic waiting strategies rather than a priori route design. We cannot guarantee that this method of constructing the route of advance-request customers is the best.

Because of the “curse of dimensionality,” we are unable to compare the heuristics to the optimal solution for anything but small datasets. Hence, we compare the heuristics through simulation. For each dataset, we ran each heuristic 1,000 times. The average values returned by the heuristics are labeled as LW, LW No, CL, WAS, and DW, for the LW, LW No, CL, WAS, and DW heuristics, respectively.

Each of the tables presents a series of comparisons. Each comparison compares one of the heuristics to the LW heuristic. This comparison is presented as the percentage less than the mean LW solution value returned by the heuristic, or formally, \[
\left(100 \times \frac{\text{avg. LW solution} - \text{avg. heuristic solution}}{\text{avg. LW solution}}\right). \]
These percentage less than optimal values are labeled $\Delta$ LW No, $\Delta$ CL, $\Delta$ WAS, and $\Delta$ DW for the LW No, CL, WAS, and DW heuristics, respectively.

We also compare the heuristics using paired-\(t\) confidence interval for the difference of means using the LW heuristic as a standard. For further discussion of the details of this comparison, see Law and Kelton (2000). All comparisons use a sample size of 1,000. The comparison between the LW and the LW No heuristics is labeled LW - LW No CI. The comparison between the LW and CL heuristics is labeled LW-CL CI, the comparison between the LW and WAS heuristics is labeled LW-WAS CI.
LW-WAS CI, and the comparison between the LW and DW heuristics is labeled LW-DW CI. A confidence interval containing 0 indicates that the performance of the two heuristics is not different at the 95% confidence level. An interval for which the upper and lower confidence limits are positive indicates that the LW heuristic outperformed the alternate heuristic at a 95% confidence level. An interval for which the upper and lower confidence limits are negative indicates that the alternate heuristic outperforms the LW heuristic at the 95% confidence level.

7.2. Results

This section presents the results of the computational experiments. Tables 1 and 2 present comparisons of the heuristics for the datasets with three late-request customers and 33% and 66% waiting time, respectively. As the average solution values and the confidence intervals indicate, the LW heuristic typically outperforms the CL and WAS heuristics in both the 33% and 66% waiting-time cases. The performance over the WAS heuristic is significant, resulting from that fact that the WAS heuristic allows for the least amount of waiting because no advance-request customers are serviced while the vehicle waits at the start. The generally poor performance of the CL heuristic is the result of a similar phenomenon. By waiting as near as possible to the center-of-gravity, the CL heuristic often waits early in a route and ultimately limits the total amount of waiting time by not serving late-request customers while it waits for service requests.

The performance of the LW heuristic in comparison to DW heuristic is less robust. The average solution value returned by the LW heuristic is better than the value returned by the DW heuristic on seven of the eight datasets. As indicated by the confidence intervals, however, the LW heuristic outperforms the DW at the 95% confidence level on just two of the sets. On all other instances, the two heuristics are indistinguishable at the 95% confidence level.

When the waiting time is increased to 66%, the performance of the LW heuristic in comparison to the DW heuristic improves. In this case, the LW heuristic outperforms the DW heuristic with 95% confidence on half of the sets while the performance is indistinguishable at the 95% confidence level on the other half. The improved performance results from the fact that the LW heuristic better positions the vehicle to use the additional waiting time while by its nature the DW heuristic simply waits more at each advance-request customer.

The most interesting comparison is between the LW and LW No heuristics. In the case of 33% waiting time, the mean solution value of the LW heuristic is better on seven of the eight datasets. However, the performance between the two heuristics is the same at the 95% confidence level on all but one of the sets. With 66% waiting time, the LW No heuristic exhibits superior performance at the 95% confidence level on two of the sets while the LW heuristic demonstrates superior
performance on only one set. This performance comparison suggests that it is more important to
know the locations of late-request customers than the probability distribution on their likelihood
of requesting service.

As we increase the percentage of late request customers to 25% of the total number of customers,
we encounter similar results. Tables 3 and 4 show the results for runs on datasets in which 25%
of the customers are late-request and with 33% and 66% waiting time, respectively. As the data
indicates, the LW heuristic again outperforms both the CL and WAS heuristics. The relative
performance of the WAS heuristic is again particularly poor.

The data shows a decline in the performance of the LW heuristic in comparison to the DW
heuristic on the sets with 33% waiting time. The LW heuristic outperforms the DW heuristic at the
95% confidence level on just three datasets, and the DW heuristic outperforms the LW heuristic on
one dataset. Yet, when the waiting time increases to 66%, the LW heuristic outperforms at the 95%
confidence level the DW heuristic on all but one set. This result follows from the LW heuristic’s
ability to choose where to wait, and with more waiting time, this ability improves performance.

The performance of the LW and LW No heuristics are identical at the 95% confidence level on
the 33% waiting time sets. In fact, the LW No heuristic outperforms the LW heuristic with regard
to the average solution value on two of the datasets. As Table 4 demonstrates, the two heuristics
have also have nearly identical results with 66% waiting time. Essentially, the stochatic information
included in the LW heuristic leads to no performance advantage.

Table 5 presents the results of runs on the datasets in which 50% of the customers are late-
request and 33% waiting time. The relative performance of the heuristics is similar in the 66%
waiting time case, and we consequently omit that table. Table 5 also omits the WAS heuristic from
the comparison as it performs as poorly as it has on the other comparisons. In its place, we add
a comparison between the DW and LW No heuristics. The percentage difference between the two
heuristics is calculated as $\left(100 \times \frac{\text{avg. LW No solution} - \text{avg. DW solution}}{\text{avg. LW No solution}}\right)$. In this comparison, a
confidence interval which is positive means that the DW heuristic outperforms the LW No heuristic
at the 95% confidence level.

The data in Table 5 show that the performance of the LW heuristic has declined relative to the
other heuristics as the percentage of late request customers has increased. With 50% late-request
customers, for example, the LW heuristic outperforms at the 95% confidence level the CL heuristic
on only three of the eight datasets. The CL heuristic outperforms the LW heuristic on two of the
eight. This represents a marked departure from the relative performance of the two heuristics when
a smaller percentage of customers are late request.
<table>
<thead>
<tr>
<th>Set</th>
<th>LW</th>
<th>LW No</th>
<th>Δ LW No</th>
<th>LW - LW No CI</th>
<th>CL</th>
<th>Δ CL</th>
<th>LW - CL CI</th>
<th>WAS</th>
<th>Δ WAS</th>
<th>LW - WAS CI</th>
<th>DW</th>
<th>Δ DW</th>
<th>LW - DW CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.67</td>
<td>1.60</td>
<td>4%</td>
<td>(0.01, 0.14)</td>
<td>1.26</td>
<td>24%</td>
<td>(0.34, 0.48)</td>
<td>0.71</td>
<td>58%</td>
<td>(0.89, 1.03)</td>
<td>1.64</td>
<td>2%</td>
<td>(-0.05, 0.10)</td>
</tr>
<tr>
<td>2</td>
<td>1.93</td>
<td>1.91</td>
<td>1%</td>
<td>(-0.05, 0.09)</td>
<td>1.69</td>
<td>12%</td>
<td>(0.16, 0.31)</td>
<td>0.80</td>
<td>58%</td>
<td>(1.05, 1.20)</td>
<td>1.94</td>
<td>-1%</td>
<td>(-0.09, 0.06)</td>
</tr>
<tr>
<td>3</td>
<td>1.82</td>
<td>1.80</td>
<td>1%</td>
<td>(-0.05, 0.09)</td>
<td>1.68</td>
<td>8%</td>
<td>(0.07, 0.22)</td>
<td>0.82</td>
<td>55%</td>
<td>(0.92, 1.08)</td>
<td>1.76</td>
<td>3%</td>
<td>(-0.01, 0.13)</td>
</tr>
<tr>
<td>4</td>
<td>1.62</td>
<td>1.58</td>
<td>3%</td>
<td>(-0.03, 0.11)</td>
<td>1.22</td>
<td>24%</td>
<td>(0.32, 0.47)</td>
<td>0.39</td>
<td>76%</td>
<td>(1.16, 1.30)</td>
<td>1.52</td>
<td>6%</td>
<td>(0.03, 0.17)</td>
</tr>
<tr>
<td>5</td>
<td>1.76</td>
<td>1.73</td>
<td>2%</td>
<td>(-0.05, 0.10)</td>
<td>1.54</td>
<td>13%</td>
<td>(0.15, 0.30)</td>
<td>0.59</td>
<td>67%</td>
<td>(1.10, 1.25)</td>
<td>1.73</td>
<td>2%</td>
<td>(-0.04, 0.10)</td>
</tr>
<tr>
<td>6</td>
<td>1.92</td>
<td>1.95</td>
<td>2%</td>
<td>(-0.11, 0.04)</td>
<td>1.94</td>
<td>-1%</td>
<td>(-0.10, 0.04)</td>
<td>0.24</td>
<td>88%</td>
<td>(1.61, 1.75)</td>
<td>1.86</td>
<td>3%</td>
<td>(-0.01, 0.13)</td>
</tr>
<tr>
<td>7</td>
<td>1.77</td>
<td>1.69</td>
<td>5%</td>
<td>(0.01, 0.16)</td>
<td>1.45</td>
<td>18%</td>
<td>(0.25, 0.40)</td>
<td>0.82</td>
<td>54%</td>
<td>(0.88, 1.02)</td>
<td>1.65</td>
<td>7%</td>
<td>(0.05, 0.19)</td>
</tr>
<tr>
<td>8</td>
<td>1.79</td>
<td>1.79</td>
<td>0%</td>
<td>(-0.07, 0.07)</td>
<td>1.60</td>
<td>11%</td>
<td>(0.12, 0.27)</td>
<td>0.65</td>
<td>64%</td>
<td>(1.07, 1.22)</td>
<td>1.75</td>
<td>3%</td>
<td>(-0.03, 0.12)</td>
</tr>
</tbody>
</table>

Table 1: Results for 3 Late-Request Customers and 33% Waiting Time
Table 2: Results for 3 Late-Request Customers and 66% Waiting Time

| Customer | Anticipated Time | Actual Time | % Error | 95% CI | Anticipated Time | Actual Time | % Error | 95% CI | Anticipated Time | Actual Time | % Error | 95% CI |
|----------|------------------|-------------|---------|
| 1        | 1.91             | 1.90        | 1%      | (-0.06, 0.09) | 1.50           | 22%         | (0.34, 0.49) | 1.21 | 37%         | (0.63, 0.79) | 1.80 | 6%      | (-0.04, 0.18) |
| 2        | 2.23             | 2.27        | -2%     | (-0.10, 0.03) | 2.05           | 8%          | (0.11, 0.25) | 0.57 | 74%         | (1.57, 1.74) | 2.13 | 5%      | (-0.03, 0.17) |
| 3        | 2.22             | 2.19        | 1%      | (-0.04, 0.09) | 1.98           | 11%         | (0.17, 0.31) | 1.42 | 36%         | (0.71, 0.88) | 2.02 | 9%      | (-0.13, 0.26) |
| 4        | 1.95             | 1.94        | 1%      | (-0.06, 0.08) | 1.55           | 20%         | (0.33, 0.47) | 0.96 | 51%         | (0.90, 1.08) | 1.89 | 3%      | (-0.01, 0.13) |
| 5        | 1.94             | 1.97        | -1%     | (-0.10, 0.04) | 1.86           | 4%          | (0.01, 0.15) | 1.10 | 43%         | (0.76, 0.93) | 1.97 | -2%     | (-0.04, 0.04) |
| 6        | 2.10             | 2.18        | -4%     | (-0.15, -0.01) | 2.14          | -2%         | (-0.11, 0.03) | 0.57 | 73%         | (1.45, 1.61) | 2.07 | 1%      | (-0.04, 0.10) |
| 7        | 1.94             | 2.01        | -3%     | (-0.14, 0.00) | 1.52           | 22%         | (0.35, 0.50) | 1.08 | 44%         | (0.78, 0.94) | 1.88 | 3%      | (0.00, 0.14) |
| 8        | 2.10             | 2.01        | 4%      | (0.02, 0.16)  | 1.85           | 12%         | (0.18, 0.32) | 1.18 | 44%         | (0.85, 1.01) | 2.10 | 0%      | (-0.07, 0.07) |
Table 3: Results for 25% Late-Request Customers and 33% Waiting Time
Table 4: Results for 25% Late-Request Customers and 66% Waiting Time

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LW No</td>
<td>9.91</td>
<td>9.92</td>
<td>-0.10</td>
<td>0.09</td>
<td>9.63</td>
<td>3%</td>
<td>(0.19, 0.38)</td>
<td>7.91</td>
<td>20%</td>
</tr>
<tr>
<td>LW No</td>
<td>10.77</td>
<td>10.72</td>
<td>-0.04</td>
<td>0.12</td>
<td>10.48</td>
<td>3%</td>
<td>(0.20, 0.37)</td>
<td>9.12</td>
<td>15%</td>
</tr>
<tr>
<td>LW No</td>
<td>10.51</td>
<td>10.59</td>
<td>-0.16</td>
<td>0.01</td>
<td>10.29</td>
<td>2%</td>
<td>(0.14, 0.31)</td>
<td>9.14</td>
<td>13%</td>
</tr>
<tr>
<td>LW No</td>
<td>8.38</td>
<td>8.13</td>
<td>0.16</td>
<td>0.33</td>
<td>7.47</td>
<td>12%</td>
<td>(0.81, 1.01)</td>
<td>6.51</td>
<td>22%</td>
</tr>
<tr>
<td>LW No</td>
<td>8.21</td>
<td>8.29</td>
<td>-0.17</td>
<td>0.01</td>
<td>7.57</td>
<td>8%</td>
<td>(0.53, 0.73)</td>
<td>6.73</td>
<td>18%</td>
</tr>
<tr>
<td>LW No</td>
<td>8.58</td>
<td>8.60</td>
<td>-0.11</td>
<td>0.06</td>
<td>8.37</td>
<td>3%</td>
<td>(0.12, 0.30)</td>
<td>6.75</td>
<td>21%</td>
</tr>
<tr>
<td>LW No</td>
<td>8.09</td>
<td>8.11</td>
<td>-0.11</td>
<td>0.07</td>
<td>7.70</td>
<td>5%</td>
<td>(0.29, 0.49)</td>
<td>6.57</td>
<td>19%</td>
</tr>
<tr>
<td>LW No</td>
<td>8.41</td>
<td>8.40</td>
<td>-0.07</td>
<td>0.10</td>
<td>8.14</td>
<td>3%</td>
<td>(0.18, 0.36)</td>
<td>6.72</td>
<td>20%</td>
</tr>
</tbody>
</table>
The relative performance decline of the performance of the LW heuristic is not surprising and is best explained by examining the LW No and DW heuristics. With 50% late-request customers, both heuristics outperform the LW heuristic. The LW No heuristic outperforms at the 95% confidence level the LW heuristic on four of the eight datasets while the LW heuristic outperforms the LW No heuristic on only two of the sets. The relative improvement of the LW No heuristic results from the fact that, with a greater percentage of late-request customers, even a customer with a relatively high probability of requesting service is unlikely to have a higher probability of requesting service than the aggregate of the probabilities of the rest of the customers. Hence, shading the center-of-gravity towards this higher probability customer actually has a deleterious effect on the heuristic’s performance.

At the same time, the DW heuristic outperforms both of the center-of-gravity heuristics. Essentially, as the percentage of late-request customers is increased, the environment more closely resembles that of customers arriving uniformly to some bounded region. Hence, as noted in Section 6.5, the DW heuristic demonstrates good performance. On the other hand, the center-of-gravity offers little discriminating information as it includes greater percentages of customers. Hence, the performance of the center-of-gravity heuristics decline relative to the DW heuristic, and the performance of the LW heuristic equalizes compared to the CL heuristic.

8. Conclusions

This paper explicitly considers the question of when and where a vehicle should wait for future service requests given stochastic information about service requests and known customer locations. Using a Markov decision process model, we show for the general case a situation when the vehicle should always wait at the current location and a situation in which the vehicle should never wait at the current location. For the one customer case, we identify an optimal policy. Using the structural results, we develop a real-time heuristic and compare this heuristic to other intuitively appealing heuristics.

The empirical comparisons reveal that a heuristic based on the structural results performs superiorly when the percentage of late-request customers is 25% or less. The results also show that customer location information is more valuable than information about the likelihood of the customer to request service. When the percentage of late-request customers is higher, the discriminating power of the heuristic’s information declines and concurrently so does its relative performance. In the case of a high percentage of late-request customers, our results show that distributing waiting time across the already routed customers is an effective heuristic strategy. Such a strategy has
Table 5: Results for 50% Late-Request Customers and 33% Waiting Time

<table>
<thead>
<tr>
<th>Season</th>
<th>% Change</th>
<th>% Change</th>
<th>% Change</th>
<th>% Change</th>
<th>% Change</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.006 - 0.34</td>
<td>0.096</td>
<td>0.096</td>
<td>0.096</td>
<td>0.096</td>
<td>0.096</td>
<td>0.096</td>
</tr>
<tr>
<td>0.34 - 0.69</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
<td>0.089</td>
</tr>
<tr>
<td>0.69 - 1.27</td>
<td>0.068</td>
<td>0.068</td>
<td>0.068</td>
<td>0.068</td>
<td>0.068</td>
<td>0.068</td>
</tr>
<tr>
<td>1.27 - 1.85</td>
<td>0.056</td>
<td>0.056</td>
<td>0.056</td>
<td>0.056</td>
<td>0.056</td>
<td>0.056</td>
</tr>
<tr>
<td>1.85 - 2.35</td>
<td>0.030</td>
<td>0.030</td>
<td>0.030</td>
<td>0.030</td>
<td>0.030</td>
<td>0.030</td>
</tr>
<tr>
<td>2.35 - 2.67</td>
<td>0.025</td>
<td>0.025</td>
<td>0.025</td>
<td>0.025</td>
<td>0.025</td>
<td>0.025</td>
</tr>
</tbody>
</table>
previously been shown to perform well in situations where customer location information was not known in advance.

Future work should begin by considering the effects of a capacitated vehicle on the results. The natural extension would be to consider a fleet of capacitated vehicles that serve a set of customers. One difficulty to overcome in the extension to a fleet of vehicles is that the structural results will be difficult to prove and unlikely to hold as the result of interactions between the vehicles. However, the single vehicle results may help heuristic development. In addition to capacity and fleets of vehicles, future work could also consider time windows. Customers are increasingly demanding time-sensitive services and very few papers consider time sensitivity in the context of stochastic and dynamic environments. Finally, in this paper, we assumed that we were given a particular route for the advance-request customers. In the implementation here, the route was generated ignoring the existence of the late-request customers. However, there is likely value in determining an a priori route for the advance request customers that anticipates the service requests of late-request customers.

Acknowledgments
This work was partially supported by a grant from the University of Iowa Old Gold Foundation. The author would also like to thank three anonymous referees and his colleagues Philip Jones and Jeff Ohlmann for their useful suggestions.

References


