TRANSPORTATION SCIENCE
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0041-1655 | EISSN 1526-5447 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

A Comparison of Anticipatory Algorithms for the Dynamic and Stochastic Traveling Salesman Problem

Gianpaolo Ghiani, Emanuele Manni
Dipartimento di Ingegneria dell'Innovazione, Università del Salento,
Via per Monteroni, 73100 Lecce - Italy,
{gianpaolo.ghiani, emanuele.manni}@unisalento.it

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

Advances in information technology and telecommunications, together with ever growing amounts of data, offer opportunities for transportation companies to improve the quality of the service that they provide to their customers. This paper explores this issue in the context of a dynamic and stochastic routing problem in which a single, uncapacitated vehicle serves a set of known customers locations. Two solution approaches are explored. One approach, sample-scenario planning, offers the potential for higher quality solutions, but at the expensive of greater computational effort. Anticipatory insertion offers reduced computation and increased managerial ease, but with the potential for reduced solution quality due to restrictions on solution structure. Our results show that anticipatory insertion can often match the quality of sample-scenario planning, particularly when the degree of dynamism is low.

Key words: vehicle routing, dynamic, stochastic

1. Introduction

Advances in information technology and computing power, together with ever growing amounts of data, offer opportunities for transportation companies to improve the quality of the service that they provide to their customers. In particular, increased computing power allows companies to use routing algorithms that can generate solutions in real-time while incorporating recent information and also anticipating future events. New information technology allows updated solutions to be immediately communicated to drivers in the field.

Historically, neither was possible. Even if companies had the data necessary to anticipate future events, they lacked the computing power to update solutions in real-time, at least without placing some sort of constraint on the structure of the solution. Further, real-time communication has only become universal in the past decade.

Despite the possibilities now available, however, there are important considerations. Notably, while the computing power exists for sophisticated anticipatory, real-time solutions, the computing ability comes at a cost of both upgraded hardware and management. Little research exists that explores what the value of the more sophisticated approaches is relative to an approach that requires less computation.

In this paper, we explore this question. We compare the performance of two anticipatory routing heuristics: anticipatory insertion (AI) and sample-scenario planning (SP). Anticipatory heuristics explicitly incorporate information about future events. Our interest in comparing the algorithms stems from the fact that the latter method offers the potential for improved solution quality, due to the fact that it does not restrict solution structure, but requires significantly more computational effort. The question that we seek to answer is whether or not the less restrictive solution structure and resulting increased computation lead to higher quality solutions.

In an anticipatory-insertion heuristic, a tour or tours of customers is constructed by inserting customers service requests into existing routes, if it is feasible to do so, as the requests occur. To improve the chances of being able to accommodate as many requests as possible, requests are anticipated by having the vehicle wait at specific customer locations on the tour. In our case, the locations at which to wait are chosen by taking advantage of information regarding both the location and likelihood of service requests from late-request customers. The method used for comparison in this paper is derived from Thomas (2007).

Sample-scenario planning constructs a set of tours by sampling future service requests and then routing the sampled requests and known customers using a routing scheme appropriate for the problem. The sampled customers are then deleted from each constructed tour, leaving tours of only known customers. The intuition, backed by the results from Bent and Van Hentenryck (2004), is that, having been constructed with sampled future customers, these routes improve the ability to accommodate future requests. From the set of tours, a "distinguished" plan is chosen by means of a consensus function. The tour is then executed according to the order prescribed by the distinguished plan. As new requests for service occur, the heuristic attempts to insert as many new requests as possible into each of the tours in the previously generated set of tours. Those tours that can accommodate the new requests are maintained, and those that cannot are deleted and replaced by new tours generated using the sampling procedure. A new distinguished tour is then chosen from the set of newly constructed tours and remaining tours. By updating the distinguished tour in this way, sample-scenario planning takes advantage of the information that has been learned over time.

As a testbed for our comparison, we use a version of the well known dynamic and stochastic Traveling Salesman Problem (DTSP). The DTSP is characterized by a single, uncapacitated vehicle serving a set of known customers locations. The choice to use an uncapacitated vehicle reflects the situation in the courier and package-express industries, where the size of parcels is small enough so that vehicle capacity is not a crucial aspect of the problem. The vehicle begins its route from a known starting point and must complete its journey at a given goal or end node (not necessarily the starting point) by a known time horizon. At the beginning of the time horizon, the driver of the vehicle is aware of a set of customers, called advance-request customers, who have already requested service. These customers may represent packages which are on the vehicle for delivery by the end of the day. In addition to the advance-request customers, there is another subset of customers, called late-request customers, such that, at the beginning of the service horizon, it is unknown whether or not these customers will require service. We assume a known probability distribution on the likelihood that late-request customers will request service. Our objective is to maximize the expected number of late-request customers who are served. This objective can be seen as equivalent to maximizing customers' Quality of Service (QoS). The impact of customers' QoS on companies is very important, as low QoS can result in customer reimbursement or lost sales (van de Klundert and Wormer 2010).

The key contribution of this paper is the demonstration that anticipatory insertion offers comparable performance while requiring less computational resources. As a minor contribution, we demonstrate that a waiting strategy borrowed from anticipatory insertion can be used in conjunction with a sample-scenario-planning heuristic.

This paper is outlined as follows. In Section 2, we review the relevant literature. Section 3 formally presents the DTSP, and Section 4 describes in detail our solution approaches. Section 5 outlines the experimental design and discusses the results of computational experiments comparing anticipatory insertion and sample-scenario planning. Finally, Section 6 concludes the paper and discusses future work.

2. Literature Review

This paper is interested in the solution of routing problems that can be categorized as stochastic and dynamic. A routing problem is stochastic when information about the customer service requests can be described by a random variable with a known probability distribution. A routing problem is dynamic when information about customer demand or service requests are revealed over the problem horizon. There are two classes of literature most relevant to the discussion in this paper. We will first provide an overview of the literature encompassing the spectrum of solution methods for dynamic and stochastic vehicle routing problems in which a subset of customer service requests become known while the vehicle/vehicles is/are enroute. We then discuss a particularly relevant subset of the dynamic and stochastic routing literature. This subset discusses comparisons between what are called a priori or fixed route methods and reoptimization approaches.

We categorize the solution methods in two ways. First, solution methods are distinguished by the degree to which they constrain the solution. We consider two methods: insertion and unrestricted. Insertion methods insert new service requests into existing routes. Thus, the method preserves some structure of a previous solution. Unrestricted methods allow the solution to change in anyway throughout the problem horizon. We note that unrestricted solutions are often called rolling-horizon solutions as the solution is updated in an iterative or rolling fashion as new information is learned. Second, the methods are categorized by the degree to which they use available information. We consider two ways: myopic and anticipatory. Myopic methods make decisions without incorporating knowledge of the future. On the other hand, anticipatory methods incorporate information about the future.

As exact approaches to dynamic and stochastic routing problems are limited to small problem sizes, most solution methods for dynamic and stochastic routing research focus on heuristic solutions. The earliest insertion methods were myopic in nature and applied to problems often called the dynamic routing and dispatching problem (DRDP). Overviews of the DRDP and related literature can be found in Powell et al. (1995) and Psaraftis (1988, 1995). More modern myopic insertion methods implicitly acknowledge the possibility of future requests by allowing waiting within the vehicle tours. Larsen et al. (2002) consider a variety of insertion procedures for dynamic routing.

Anticipatory-insertion methods exploit information about future requests. Branke et al. (2005) use a sampling procedure incorporated into an evolutionary algorithm to determine the best a priori routes for the advance-request customers. The routes are then used in conjunction with waiting strategies and insertion techniques to handle dynamic requests. Thomas (2007) considers a problem similar to the one in this paper and demonstrates when and where a vehicle should wait in anticipation of future requests. Thomas extrapolates the structure for the optimal policy for one late-requesting customer to develop a real-time heuristic that performs well when the percentage of late-request customers is 25% or less. The author shows that a strategy that distributes waiting time across advance request customer locations works well as the percentage of late-request customers increases. The results also demonstrate that waiting is an effective strategy for improving performance through anticipation. Larsen et al. (2004) demonstrates an analogous result for a different dynamic routing problem. In this paper, we also use results from Manni (2007) to construct better initial routes into which to incorporate waiting.

Myopic and unrestricted solution methods often take advantage of modern computing to incorporate real-time information into solutions as it arrives. Early work can be found in Kilby et al. (1998). Gendreau et al. (1999) and Ichoua et al. (2000) use a parallel implementation of a tabu search to continuously update vehicle routes as new requests occur. Mitrović-Minić and Laporte (2004) consider a dynamic pick-up and delivery problem, and develop four waiting strategies. We note that we call these waiting strategies myopic because they do not explicitly incorporate information about future customer demand. The authors show that an adequate distribution of this waiting time may affect the planner's ability to make good decisions at a later stage. Mitrović-Minić et al. (2004) extend Mitrović-Minić and Laporte (2004) to include "double-horizon" heuristic that minimizes route distance for customers served in the near-term while trying to maintain flexibility in the longer term.

With computer processing power having increased enough to make such implementations possible for real-world sized problems, authors have recently begun to explore anticipatory, unrestricted solutions. Some extend existing heuristics to incorporate information about the future. For example, Ichoua et al. (2006) 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. Alternatively, Secomandi and Margot (2009) exploit structure specific to the Vehicle Routing Problem with Stochastic Demand (VRPSD). In contrast to the problem considered here, in the VRPSD, customer demand volumes rather than the customers' need for service is random.

Another approach is to rely on a more general framework such as approximate dynamic programming (ADP). In ADP, an action for the current state of a Markov decision process is chosen using some estimate of the cost-to-go. A routing example is given in Secomandi (2000) and Secomandi (2001). These works iteratively apply or rollout a heuristic estimate future costs and use the estimate to choose the next customer to visit in a VRPSD. The procedure is anticipatory because the heuristic incorporates information about possible customer demand. Goodson et al. (2010) generalizes the rollout method used in Secomandi (2000) and Secomandi (2001) to solve a multi-vehicle version of the VRPSD. Meisel et al. (2009) use a form of ADP in which the cost-to-go is approximated via a parametric function. While Meisel et al. (2009) solve a problem similar to that in this paper, our preliminary results indicate the solution quality of the two algorithms compared here is better.

In many anticipatory, unrestricted algorithms, the anticipation is handled via a sampling procedure. Ghiani et al. (2009) propose anticipatory algorithms for a dynamic routing and dispatching problem, in which alternative solutions are evaluated through a short-term demand sampling and a fully sequential procedure for indifference zone selection. Computational results show that this approach provides consistently better solutions than an algorithm that simply reacts to demand as it occurs. van Hemert and La Poutré (2004) introduce the concept of fruitful regions, which are clusters of known customer locations that may require service in the near future. Potential schedules are created by sampling fruitful regions. The authors then provide an evolutionary algorithm for determining when to move to one of the fruitful regions in anticipation of future service requests.

Of most interest to the discussion in this paper are sampling methods generally called sample-scenario planning or progressive hedging. In a sample-scenario approach, at each decision point, a number of samples of future demand are generated. For each sample, a deterministic routing problem is solved that incorporates the sample information as known information. From the resulting set of deterministic solutions, some method is used to choose a best action. The method was introduced for dynamic vehicle routing by Bent and Van Hentenryck (2004). The authors consider a dynamic vehicle routing problem with time windows, and use sampling to construct a set of potential routes containing existing customers as well as possible future customers obtained via sampling. From this set of routes, a "distinguished" route is chosen using a consensus function. The consensus function is a measure of how similar two solutions are. In Bent and Van Hentenryck (2004), the distinguished solution is the one most alike the others solutions in the set. The immediate next actions are then chosen based on the distinguished solutions. Considering a problem similar to Bent and Van Hentenryck (2004), a comparable procedure is used by Hvattum et al. (2006). The key difference is that, instead of a consensus mechanism, Hvattum et al. (2006) use a multi-phase procedure to identify the customers who should be visited in the near future.

In this paper, we use a sample-scenario framework as our anticipatory, unrestricted solution method. Because we are solving a different problem, some features of our implementation are necessarily different than those in Bent and Van Hentenryck (2004) and Hvattum et al. (2006). Notably, because we have a single vehicle, we use a consensus function that, although similar in spirit to that in Bent and Van Hentenryck (2004), is different. At the same time, both Bent and Van Hentenryck (2004) and Hvattum et al. (2006) consider problems with time windows. In those cases, when and where to wait is determined by the time windows. In our problem, there are no time windows. Resultantly, some other mechanism waiting mechanism is required (as two examples, both Mitrović-Minić and Laporte (2004) and Thomas (2007) improves performance). Full discussion of our algorithm is given in Section 4.

In this paper, we are most interested in the effect that imposed solution structure has on results. For a different class of routing problems, a related comparison is made in the literature comparing a priori or fixed route solution methods and reoptimization methods. The a priori method is analogous to anticipatory insertion and reoptimization approach to the reactive DRDP methods discussed above. The comparison of the methods is done on problems in which the set of customers is known in advance, but who will need to be visited or the volume of demand is not revealed until the day that the routes need to be operational (in constrast, in our problem, the information is revealed while the vehicle is en route.) The question is then whether or not it is better to route all the customers in advance (the a priori or fixed route solution) or to reoptimize once the actual demand is revealed. Research such as Waters (1989), Benton and Rosetti (1992), Haughton (2000), and Savelsbergh and Goetschalckx (1995) demonstrates that the fixed route solutions can frequently achieve comparable cost to the less restricted solutions of a reoptimization approach. This conclusion is important because, in addition to the possibility that route reoptimization is computationally infeasbile, as Benton and Rosetti (1992) note, reoptimization has "hidden costs" not captured by the routing models. As an example of such costs, Haughton (2002) points to the daily changes in route assignments to drivers and thus the need for drivers to frequently learn new routing territories.

The research presented here is analogous to the comparison discussed above for the case in which the dynamic information is revealed not only daily but also while the vehicle is en route. In our case, the fixed route methods discussed above do not apply because it cannot account for the timing of service requests. Nonetheless, our anticipatory-insertion method does fix at least relatively the sequence of some customers in advance. While the managerial implications (such as learning or service consistency) of the studied algorithms requires further study, like those discussed above, our results demonstrate that there is a minimal decline in solution quality when restricting the solution structure in advance (anticipatory insertion) versus having no restriction (rolling horizon).

3. Problem Formulation

In this section, we present a formulation for the described dynamic routing problem. A formal dynamic programming model for a similar problem is presented in Thomas (2007).

Let N be the set of customers and t_{ij} be the travel time (assumed to be integer) between customers i and j in N. The set N is partitioned into a set of advance-request customers N_A and a set of late-request customers N_L. The set N_A includes a dummy customer s from which the vehicle leaves at time 0, and a dummy customer γ that the vehicle must reach by time T.

The status of a customer i at time t can be represented by a random variable $Z_i(t)$ defined as follows:

$$Z_i(t) = \begin{cases} 0 & \text{if customer } i \text{ has not yet requested service by time } t; \\ 1 & \text{if customer } i \text{ has requested service, but it has not been decided by time } t \\ & \text{whether he/she will be serviced or not;} \\ 2 & \text{if customer } i \text{ has requested service, has been approved for service,} \\ & \text{but has not been visited by time } t; \\ 3 & \text{if customer } i \text{ has been visited or rejected by time } t. \end{cases}$$

We emphasize that, for each advance-request customer i, $Z_i(t)$ can be equal to only 2 or 3. We can then describe the state of system as the tuple (n,t,z), where n is the location of the vehicle at time t and z is a vector representing the status of the customers at time t.

At a decision epoch, both an *acceptance* and a *movement* action are selected. The acceptance action amounts to selecting those customers currently in status 1 who will be served over the problem horizon. We require that any accepted customers can be feasibly served, given the set of previously accepted customers. The movement action prescribes the vehicle to wait at the current location or to move onto the next customer in the current route. We assume that the vehicle can travel to a customer who has been accepted by the concurrent acceptance action. However, we assume that customers in status 3 cannot be visited. A decision epoch occurs when the vehicle arrives at a customer or after the vehicle has waited for one unit of time following the selection of the waiting action.

The dynamics of customer status are as follows. Customers selected by the acceptance action undergo a deterministic status transition from 1 to 2. A customer in status 1 who was not chosen for acceptance transitions to status 3. If the state of the system is such that the vehicle is currently at location n in status 2, then selecting the action to move to another customer induces a deterministic transition of customer n from status 2 to status 3. Thus, the state of the system will reflect that all visited customers are in status 3. On the other hand, the status of a customer n remains unchanged if the movement action requires the vehicle to wait at its current location.

For customers in status 0, their status is described by a Markov chain. We assume the Markov chains of the individual customers are independent. For a customer i, the $(t'-t)$-step transition matrix is:

$$R_i^{(t,t')} = \begin{bmatrix} 1-\alpha_i^t & \alpha_i^t \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1-\alpha_i^{t+1} & \alpha_i^{t+1} \\ 0 & 1 \end{bmatrix} \times \cdots \times \begin{bmatrix} 1-\alpha_i^{t'-1} & \alpha_i^{t'-1} \\ 0 & 1 \end{bmatrix},$$

where α_i^t is the probability that, between time t and $t+1$, customer i transitions from status 0 to status 1. The associated probability $P(z' \mid t,z,t')$ of a transition from customer status vector z at time t to status vector z' at time $t' > t$ is:

$$\begin{aligned} P(z' \mid t,z,t') &= P(Z(t') = z' \mid Z(t) = z) \\ &= \prod_{i \in N_L} P(z_i(t') = z_i' \mid z_i(t) = z_i). \end{aligned}$$

Our reward structure is described by a function $c(a)$ whose value is the number of new service requests (status 1 customers) who have been accepted for service as a consequence of action a. Thus, $c(a)$ is equal to the number of newly arrived customers that action a accepts. No cost is incurred as the result of travel time or waiting. The problem objective is to maximize the expected reward obtained during the horizon.

4. Solution Approaches

In this section, we describe the two solutions approaches that we employ to solve the DTSP. The approaches are: anticipatory insertion and sample-scenario planning.

4.1. Anticipatory Insertion

Anticipatory insertion restricts the action space by following a fixed, greedy policy for the insertion portion of the action and by imposing a rigid solution structure in the case of the movement action. Both the insertion and movement restrictions require that, at each time t, the customers who have requested service and been approved by that time are ordered according to some criterion. Let τ_t be the route made up of the approved customers at time t.

At each decision epoch k, the insertion action is simply the policy of inserting the maximum possible number of new customers into the route τ_{t_k}, such that τ_{t_k} can be completed by the end of the time horizon. Formally, for a realization z of $Z(t)$ for some $t > 0$, we have a set of newly arrived customer requests (status 1 customers). At each decision epoch, we iterate through the set of subsets of these status 1 customers and choose the largest subset that can feasibly inserted via minimum cost insertion. We break ties by choosing the subset that results in the shortest tour. We note that, in our implementation, if n is a status 1 customer, and n is not included in the best subset just found, we reject it and do not again consider it for insertion. This assumption differs from that made in Thomas (2007).

At each decision epoch k, the movement action is restricted to one of two actions: wait at the current location or move onto the next customer in the tour. For both anticipatory-insertion and sample-scenario planning, we use the LW heuristic described in Thomas (2007) in order to choose if and where to wait. The LW method is motivated by the results in Thomas (2007) in which, for a single late-request customer n and a given route τ of advance-request customers, it is optimal to wait at the location that allows for the latest allowable departure time. Specifically, for a single late-request customer n, let $t^n_{j,\gamma}$ be the minimum amount of time required to travel from j to γ via n, where n is ordered according to τ. That is, $t^n_{j,\gamma} = t_{j,n} + t_{n,j+1} + \sum_{k\in\tau,k=j+1}^{\gamma-1} t_{k,k+1}$. Moreover, let $\bar{t}_j = T - t^n_{j,\gamma}$. We define $\underline{t}_j = \sum_{i=0}^{j-1} t_{i,i+1}$, for each $j \in N_A$, where the index of customers is given by the initial route to serve the advance-request customers. In other words, $\underline{t}_j$ is the earliest time that we can reach customer j given an ordering of advance-request customers.

While no similar result exists for multiple customers, we can make use of the result by aggregating the customers. For this purpose, we use the center-of-gravity (see Nahmias (2001) for further details on the center-of-gravity calculation). Formally, at time t, for each customer $i \in N_L$ with status 0, we use α_i, the one-step probability of i to request service, to weight each such customer's location in the center-of-gravity calculation. With the center-of-gravity, we create a single location for all of the customers who have not yet called. Because the problem is dynamic, we recompute the center-of-gravity at each decision epoch. For complete details of the method, see Thomas (2007). In our case, we choose to wait at the current location only if the current location maximizes $\bar{t}$ over all currently routed customers. We otherwise choose to move onto the next customer in the tour τ_t.

An important question to answer is how to design this initial route for the advance-request customers. Manni (2007) explores this question and concludes that a tour of all possible customer visits provides the best initial tour. We construct our initial tours in the same manner. More precisely, we set the first advance-request customer as the start node, whereas the last advance-request customer is set as the goal node. We then route the remaining customers through a GRASP procedure (Feo and Resende 1995) with a savings insertion criterion and a post-construction variable neighborhood search improvement scheme using 1-shift and 2-opt as neighborhoods. Let $\tau^\star$ be the best found tour after five runs of GRASP. To obtain the initial route τ_0, we simply delete from $\tau^\star$ all customers i such that $i \in N_L$.

4.2. Sample-Scenario Planning

Our rolling-horizon approach is modeled on the sample-scenario planning heuristic proposed in Bent and Van Hentenryck (2004). In our implementation of sample-scenario planning, at each time t, we maintain a set of routing plans Σ_t. A routing plan is a route of the customers that contains all the customers who have requested service and been approved by time t, ends at γ, and can be completed by time T.

Analogous to anticipatory insertion, the insertion policy is greedy. At each decision epoch k, we choose the largest subset of newly requesting customers that, using minimum cost insertion, can be feasibly inserted into a route in the set of routing plans Σ_t. We break ties by choosing the subset that can be inserted into the greatest number of plans in Σ_t.

Also analogous to anticipatory insertion, the movement actions are limited to waiting at the current location and moving onto the next customer. For both the waiting and movement to the next customer case, the action is chosen with respect to a particular tour $\sigma_t^\star \in \Sigma_t$ called the distinguished plan. In the case of the waiting action, we apply the LW heuristic as in the anticipatory-insertion case, using $\sigma_t^\star$ in the computation of $\bar{t}$. We again note that our choice of waiting strategy necessarily differs from that in Bent and Van Hentenryck (2004) because the problem presented here is not constrained by time windows.

At each decision epoch k, the distinguished plan is chosen from Σ_{t_k} by a consensus function. Our consensus function is inspired by the consensus function of Bent and Van Hentenryck (2004), and like that of Bent and Van Hentenryck, represents a least-commitment strategy (see Stefik (1981) for additional examples of least-commitment strategies). Before presenting our consensus function, we first introduce the vector $Y(\sigma,\sigma')$, where $\sigma,\sigma' \in \Sigma_t$, for which the i-th component $Y_i(\sigma,\sigma')=1$ if and only if $\sigma_i=\sigma'_i$ and $Y_i(\sigma,\sigma')=0$ otherwise, letting σ_i be the i-th customer in the tour σ. For each tour $\sigma \in \Sigma_t$, we define our consensus function as:

$$f_t(\sigma)=\sum_{\sigma'\in\Sigma,\sigma\neq\sigma'}\sum_{i\in N_A} Y_i(\sigma,\sigma').$$

We select as our distinguished plan $\sigma_t^\star \in \Sigma_t$ such that $f(\sigma_t^\star)\geq f(\sigma')$ for every $\sigma' \in \Sigma_t$.

To generate $\Sigma_{t'}$ for a decision epoch occurring at time t', we first examine the set of routing plans Σ_t, the set of routing plans before the most recent decision was made at time t. After the previous decision was made, some of the plans in Σ_t likely have become incompatible with the decision. A plan $\sigma_t \in \Sigma_t$ is said to be incompatible if the insertion action at time t is not feasible for it, or if the previous movement action as defined by the distinguished plan was not the next customer on the tour defined by σ_t. Then, to generate $\Sigma_{t'}$, all compatible plans in Σ_t are maintained in $\Sigma_{t'}$ and all incompatible plans are deleted and replaced with new plans. These new plans are created by considering a horizon of length $T-t'$ and by building routing plans for scenarios that must include all of the advance-request customers who have not been served by time t', all late-request customers who have been accepted for service but not served by time t', and that include as many of a set of sampled late-request customers as possible. Realizations of late-request customers are obtained by sampling their probability distributions, among those customers that, at time t', have not yet requested service.

To determine the route resulting from each of the scenarios, we model the problem as an orienteering problem with time windows. Let the set S_l be the set of late-request customers sampled for the generation of the l-th plan at time t'. For the advance-request customers remaining to be served at time t', we set the reward r to be equal to the cardinality of the set S_l. On the other hand, for each of the sampled late-request customers, we set the reward to 1. The result is that advance-request customers are always preferred to the sampled late-request customers in the orienteering problem. The time windows result from the fact that, in addition to sampling the presence of the late-request customers, we must sample the time at which they request service. This time represents the opening of the time window for the late-request customers. The end of the horizon T represents the closing of the time window. The advance-request customers do not have time windows. Because of the potentially wide time windows that result in our orienteering problems, published solution methods are incapable of solving the problem. We thus developed a variable neighborhood search (VNS) with a scheduled penalty multiplier that allows the time window constraints to be relaxed into the objective. The method has its roots in the compressed annealing method discussed in Ohlmann and Thomas (2007). The VNS is discussed in Manni et al. (2009). The initial set of routing plans Σ_0 is generated analogously to $\Sigma_{t'}$ except scenarios and routes are generated for each tour up to the desired cardinality of Σ_0.

5. Computational Comparison of Anticipatory Insertion and Sample-Scenario Strategies

In this section, we compare the performance of the anticipatory insertion and sample-scenario planning heuristics. Both procedures are coded in C++ and tested on Linux machines equipped with Pentium D CPUs clocked at 3.2 GHz.

5.1. Description of the instances

To compare the performance of the two heuristics, we generate a total of 192 instances. The customers and their geographical distributions are taken from eight datasets found in the literature. Three are from Solomon's 50-customers instances C101, C201, and RC101 datasets (sets 1, 2 and 3 in the following) (Solomon 1987). The other five sets are the 40-customers sets proposed in Dumas et al. (1995) (sets 4, 5, 6, 7, 8 in the following). For all eight sets, the time windows are ignored. With the eight sets as a basis, we then generate the sets by varying the percentage of late-request customers, the geographical distribution of late-request customers, the amount of time available for waiting, and the characteristics of the probabilities that the late-request customers will require service.

For the percentage of late-request customers, we consider both instances having 25% of the customers as late-request and instances 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. To choose which customers in the eight sets should be late-request customers, we consider two cases. In one case, we randomly choose late-request customers from all customers in the dataset. In the other case, we generate clusters of late-request customers. Specifically, we randomly choose a customer and then randomly choose a number of its nearest neighbors to form a cluster. For the instances having 25% of the customers as late-request we consider two clusters of 5 or 6 customers each for the 40 and 50 customer instances, respectively. In the case of 50% late-request customers, we determine four clusters of 5 or 6 customers each for the 40 and 50 customer instances, respectively. We denote clustered instances with an _C. We further characterize the instances by considering two values for the time horizon. In the manner of Thomas (2007), we refer to these horizons as the 33% and 66% cases. The two cases represent increasing amounts of time available for waiting relative to the time required to service the advance-request customers. We set the maximum total route time for each instance according to the values used by Thomas (2007).

The final piece of information characterizing an instance is the probability that a late-request customer will request service. We consider three possibilities. In the first, we randomly generate a heterogeneous set of probabilities. Specifically, the probability that a late-request customer requests service, α, is 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. We denote these heterogeneous instances by k_He ($k = 1, \dots, 8$). We also consider the cases in which late-request customers are all very likely to request service or all are not likely to request service. We denote instances in which service-request probabilities are high by k_H ($k = 1, \dots, 8$). These instances are obtained by setting all the late-request probabilities to the highest value in the corresponding heterogeneous probability instance k. Analogously, the instances of low service-request probabilities are denoted by k_L and are obtained by setting all the probabilities to the lowest value in the original set. The instances denoted k_CHe, k_CH and k_CL then represent the instances of clustered late-request customers with heterogeneous probabilities of requesting service, a high likelihood of requesting service, and with a low likelihood of requesting service, respectively.

5.2. Description of the performance measure

We compare the performance of the heuristics through simulation. For each dataset, we generate 100 samples of customer requests and run both heuristics with these samples. This method has the effect of running each solution method with common random numbers. We do not make use of a larger sample size because of the computation time required by the SP algorithm. The per decision computation time of the SP algorithm is reported in the next section. It is important to note that the number of late-requests differs among the 100 samples for each dataset because not all of the late-request customers are required to request service during the time horizon.

We present a measure of the performance of each of the heuristics, based on a comparison of how many customers were served versus how many late-request customers actually requested service. The number of late-request customers actually requesting service is then an upper bound on performance.

As a measure of performance, we present the average number of requests not served. Formally, let X_{ij}^k be the number of customers served by heuristic i on sample j of dataset k. We with let $i=0$ be SP and $i=1$ be AI. Let b_j^k be the total number of customers requesting service in sample j for dataset k. We then compute the average number of unserved requests as follows. Let $Y_{ij}^k = b_j^k - X_{ij}^k$. Then, we report for heuristic i, $\hat{\mu}_i^k = \frac{\sum_j Y_{ij}}{100}$. The difference between the two means is then $\hat{\mu}_0^k - \hat{\mu}_1^k$. We label the difference between sample means as Δ SP - AI.

Because of the sampling error in our estimate of the values of $\hat{\mu}_i^k$ for each i and k, we also present a confidence interval on the difference between the respective sample means. As a result of using common samples for each heuristic on each dataset, we compute paired-t confidence intervals. The details of the computation can be found in Law and Kelton (2000). For both measures, the confidence intervals are labeled SP - AI 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 AI heuristic outperforms the SP heuristic at a 95% confidence level. An interval for which the upper and lower confidence limits are negative indicates that the SP heuristic outperforms the AI heuristic at the 95% confidence level.

5.3. Computational Results

Table 1 reports the results for runs on datasets in which 25% of the customers are late requesting and for which we maintain 25 routing plans for the SP heuristic. Table 1(a) relates the results for all instances in the 33% waiting time case and shows that AI outperforms SP in 36 cases out of 48 at a 95% confidence level, whereas, in terms of average values, AI is better than SP on all instances but five. The two heuristics demonstrate comparable behavior, both in terms of confidence level and of average values, in the 66% waiting time case (Table 1(b)). In general, SP performs poorly when the late-request customers occurrence probabilities are low. These results occur because, the lower the probability of occurrence, the less likely any of the sampled customers used to construct the plans is likely to occur. Likewise, the SP heuristic performs relatively better on the high probability instances. The result follow from that fact that, the higher the likelihood that a customer will request service, the more likely the generated routing plans are to provide an accurate estimation of future events. Finally, we note that SP performs relatively better on the clustered than the non-clustered sets. This result occurs because, in the clustered set, even if a sampled customer does not actually request service, there is potentially a customer in the cluster who will. Thus, the plan resulting from the sampled clustered customers offers more information about the future than do the plans for the non-clustered customers.

To provide an explanation for SP's relatively poor performance, we note that, when the percentage of advance-request customers is high, the majority of the route time is consumed by the advance-request customers. Thus, there is greater importance on the tour that serves those customers. Our observations show that the SP heuristic often routes these advance-request customers poorly in the early part of the time horizon. To show the differences between the initial routes obtained by AI and SP, we present in Figure 1 the initial tours for both heuristics for sample

Table 1 Results for 25% Late-Request Customers and 25 SP plans

(a) Comparison (SP - AI) with 33% waiting time

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	0.40	0.65	0.25	(0.14, 0.36)
2_He	0.32	0.64	0.32	(0.20, 0.44)
3_He	1.16	1.36	0.20	(-0.03, 0.43)
4_He	0.32	0.47	0.15	(0.01, 0.29)
5_He	0.35	0.49	0.14	(0.04, 0.24)
6_He	0.06	0.56	0.50	(0.35, 0.65)
7_He	0.52	0.57	0.05	(-0.08, 0.18)
8_He	0.26	0.60	0.34	(0.22, 0.46)
1_H	0.32	0.68	0.36	(0.22, 0.50)
2_H	0.30	0.73	0.43	(0.26, 0.60)
3_H	0.30	0.73	0.43	(0.26, 0.60)
4_H	0.43	0.47	0.04	(-0.12, 0.20)
5_H	0.45	0.75	0.30	(0.17, 0.43)
6_H	0.04	0.63	0.59	(0.42, 0.76)
7_H	0.68	0.73	0.05	(-0.11, 0.21)
8_H	0.34	0.73	0.39	(0.24, 0.54)
1_L	0.24	0.62	0.38	(0.17, 0.59)
2_L	0.16	0.54	0.38	(0.25, 0.51)
3_L	0.61	0.82	0.21	(0.04, 0.38)
4_L	0.21	0.48	0.27	(0.06, 0.48)
5_L	0.22	0.35	0.13	(-0.06, 0.32)
6_L	0.05	0.52	0.47	(0.29, 0.65)
7_L	0.13	0.30	0.17	(0.00, 0.34)
8_L	0.19	0.43	0.24	(0.01, 0.47)
1_CHe	0.35	0.50	0.15	(-0.04, 0.34)
2_CHe	0.30	0.55	0.25	(0.10, 0.40)
3_CHe	1.24	1.06	-0.18	(-0.36, 0.00)
4_CHe	0.50	0.28	-0.22	(-0.40, -0.04)
5_CHe	0.25	0.53	0.28	(0.14, 0.42)
6_CHe	0.06	0.64	0.58	(0.40, 0.76)
7_CHe	0.35	0.63	0.28	(0.15, 0.41)
8_CHe	0.40	0.55	0.15	(0.04, 0.26)
1_CH	0.36	0.58	0.22	(0.08, 0.36)
2_CH	0.46	0.63	0.17	(0.03, 0.31)
3_CH	1.38	1.14	-0.24	(-0.49, 0.01)
4_CH	0.50	0.25	-0.25	(-0.41, -0.09)
5_CH	0.28	0.53	0.25	(0.13, 0.37)
6_CH	0.10	0.61	0.51	(0.36, 0.66)
7_CH	0.53	0.70	0.17	(0.03, 0.31)
8_CH	0.34	0.67	0.33	(0.20, 0.46)
1_CL	0.17	0.50	0.33	(0.17, 0.49)
2_CL	0.28	0.76	0.48	(0.31, 0.65)
3_CL	0.85	1.01	0.16	(-0.03, 0.35)
4_CL	0.30	0.18	-0.12	(-0.27, 0.03)
5_CL	0.20	0.44	0.24	(0.05, 0.43)
6_CL	0.05	0.58	0.53	(0.37, 0.69)
7_CL	0.30	0.57	0.27	(0.08, 0.46)
8_CL	0.24	0.50	0.26	(0.06, 0.46)

(b) Comparison (SP - AI) with 66% waiting time

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	0.21	0.46	0.25	(0.12, 0.38)
2_He	0.22	0.50	0.28	(0.16, 0.40)
3_He	0.69	0.95	0.26	(0.08, 0.44)
4_He	0.29	0.42	0.13	(0.01, 0.25)
5_He	0.29	0.48	0.19	(0.08, 0.30)
6_He	0.14	0.37	0.23	(0.10, 0.36)
7_He	0.21	0.48	0.27	(0.16, 0.38)
8_He	0.19	0.48	0.29	(0.18, 0.40)
1_H	0.23	0.55	0.32	(0.18, 0.46)
2_H	0.18	0.56	0.38	(0.23, 0.53)
3_H	0.18	0.56	0.38	(0.23, 0.53)
4_H	0.23	0.32	0.09	(-0.04, 0.22)
5_H	0.28	0.42	0.14	(0.02, 0.26)
6_H	0.18	0.46	0.28	(0.15, 0.41)
7_H	0.27	0.63	0.36	(0.20, 0.52)
8_H	0.18	0.49	0.31	(0.19, 0.43)
1_L	0.11	0.44	0.33	(0.18, 0.48)
2_L	0.15	0.36	0.21	(0.09, 0.33)
3_L	0.40	0.64	0.24	(0.11, 0.37)
4_L	0.22	0.35	0.13	(-0.01, 0.27)
5_L	0.19	0.33	0.14	(0.01, 0.27)
6_L	0.15	0.41	0.26	(0.10, 0.42)
7_L	0.15	0.52	0.37	(0.17, 0.57)
8_L	0.26	0.34	0.08	(-0.07, 0.23)
1_CHe	0.34	0.36	0.02	(-0.12, 0.16)
2_CHe	0.19	0.41	0.22	(0.11, 0.33)
3_CHe	0.75	0.52	-0.23	(-0.43, -0.03)
4_CHe	0.32	0.28	-0.04	(-0.18, 0.10)
5_CHe	0.19	0.46	0.27	(0.14, 0.40)
6_CHe	0.29	0.47	0.18	(0.08, 0.28)
7_CHe	0.19	0.58	0.39	(0.23, 0.55)
8_CHe	0.22	0.48	0.26	(0.14, 0.38)
1_CH	0.40	0.47	0.07	(-0.08, 0.22)
2_CH	0.16	0.46	0.30	(0.15, 0.45)
3_CH	1.03	0.46	-0.57	(-0.86, -0.28)
4_CH	0.45	0.21	-0.24	(-0.39, -0.09)
5_CH	0.20	0.45	0.25	(0.12, 0.38)
6_CH	0.18	0.58	0.40	(0.26, 0.54)
7_CH	0.22	0.57	0.35	(0.21, 0.49)
8_CH	0.23	0.53	0.30	(0.17, 0.43)
1_CL	0.30	0.47	0.17	(0.01, 0.33)
2_CL	0.12	0.38	0.26	(0.15, 0.37)
3_CL	0.50	0.41	-0.09	(-0.27, 0.09)
4_CL	0.30	0.21	-0.09	(-0.25, 0.07)
5_CL	0.25	0.38	0.13	(-0.06, 0.32)
6_CL	0.11	0.44	0.33	(0.16, 0.50)
7_CL	0.23	0.60	0.37	(0.15, 0.59)
8_CL	0.19	0.40	0.21	(0.05, 0.37)

instance 4_He with 25% late-request customers and 33% waiting time. Figure 1(a) shows the initial tour (the tour of the advance-request customers) for the AI heuristic and Figure 1(b) for the SP heuristic. The SP tour clearly crosses itself several times. Such crossing results in more time being required to serve the advance-request customers, leaving less time to serve the late-request customers who request service.

In part, the crossing of the tour is a result of the SP heuristic in that it uses sampled customers in the tour construction. The sampled customers in essence have a start time that corresponds to the time when they will request service. Accounting for these start times is what causes the tour to cross itself.

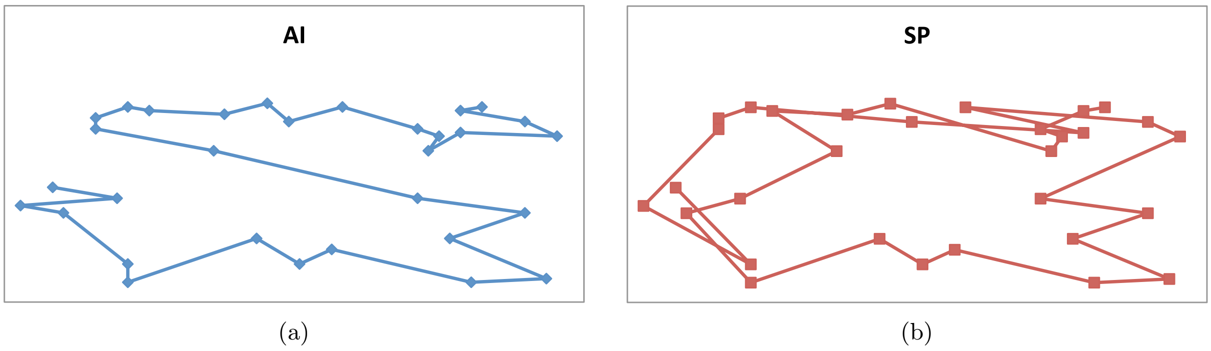

Figure 1 **Comparison of initial tours for AI and SP for instance 4_He in the case with 25% Late-Request Customers and 33% Waiting Time**

A potential solution to the problem is to initialize the SP heuristic with a different tour. In our case, we initialized the SP with the same tour of advance-customers used to initialize the AI heuristic. However, the performance of the SP heuristic was not significantly different from the results with the consensus-function initialization. The reason is because the initial tour is usually discarded at the first decision epoch as new information necessitates route updating. Thus, the potential subject of future work is to determine when the transition from the initial tour to the dynamically updated tour should occur.

Table 2 presents the comparison for the two heuristics in the case where the percentage of late-request customers is 50%. With respect to the 33% waiting time, SP outperforms AI with a confidence level of 95% in 12 cases, the two strategies are statistically equivalent in 27 cases, whereas AI outperforms SP in the remaining 9 cases. As for the 66% waiting time, in terms of 95% confidence intervals, the two heuristics are statistically equivalent in 23 cases, AI is better than SP in 22 cases, whereas SP outperforms AI in the remaining three cases. These results show an improvement of the relative performance of SP as the degree of dynamism increases. The improved relative performance of SP is due to the greater value that updating of the routing plans offers as the dynamism increases. The relative performance with respect to the instance characteristics is analogous to what we observe in the 25% late-request customers case, with the worst SP results obtained on the _L and _CL sets.

While the characteristics of the instances on which SP performs relatively better or worse are perhaps not surprising, it is surprising that AI outperforms SP on a great number of instances. Thus, the question still remains as to whether or not there is value in employing the SP heuristic in terms of the ability to serve more customer demand. Without a doubt, the performance of SP is partly the result of the number of samples used to construct the distinguished tours. To assess the value of using a greater number of SP plans, in Tables 3 and 4, we further increase the number of SP plans up to 50 and 75, for the _He and the _CH instances in the 33% case. In particular, Table 3 consider the case of 25% late-request customers and 33% waiting time, whereas Table 4 report the same analysis in the case of 50% late-request customers and 33% waiting time. We choose the two presented instance classes for further exploration as the relative performance of SP was better on these instances relative to others and thus SP was likely to show positive results as the number of plans increased. The results show that, when the number of SP plans increases up to 75 and there is a low degree of dynamism (25% late-request customers), the performance of AI is still comparable to or, in some cases, better than SP. On the other hand, with an high degree of dynamism (50% late-request customers) and 75 SP plans, there is only one case out of 16 in which AI is better than SP at a confidence level of 95%, whereas the two approaches are statistically indistinguishable in 7 cases, and SP outperforms AI in the remaining 8 cases.

Table 2 Results for 50% Late-Request Customers and 25 SP plans

(a) Comparison (SP - AI) with 33% waiting time

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	1.10	1.10	0.00	(-0.16, 0.16)
2_He	1.00	1.15	0.15	(-0.02, 0.32)
3_He	2.13	2.19	0.06	(-0.21, 0.33)
4_He	0.57	0.30	-0.27	(-0.45, -0.09)
5_He	0.77	0.90	0.13	(0.01, 0.25)
6_He	0.61	0.91	0.30	(0.15, 0.45)
7_He	0.96	1.19	0.23	(0.06, 0.40)
8_He	1.25	1.00	-0.25	(-0.46, -0.04)
1_H	1.48	1.31	-0.17	(-0.38, 0.04)
2_H	1.80	1.91	0.11	(-0.13, 0.35)
3_H	1.80	1.91	0.11	(-0.13, 0.35)
4_H	1.42	0.42	-1.00	(-1.29, -0.71)
5_H	1.28	1.50	0.22	(0.03, 0.41)
6_H	0.73	1.01	0.28	(0.12, 0.44)
7_H	1.78	1.54	-0.24	(-0.44, -0.04)
8_H	2.23	1.59	-0.64	(-0.90, -0.38)
1_L	0.13	0.15	0.02	(-0.11, 0.15)
2_L	0.14	0.31	0.17	(-0.01, 0.35)
3_L	0.39	0.44	0.05	(-0.16, 0.26)
4_L	0.14	0.08	-0.06	(-0.17, 0.05)
5_L	0.07	0.15	0.08	(-0.04, 0.20)
6_L	0.08	0.14	0.06	(-0.04, 0.16)
7_L	0.14	0.31	0.17	(0.01, 0.33)
8_L	0.09	0.22	0.13	(-0.01, 0.27)
1_CHe	1.49	1.26	-0.23	(-0.46, 0.00)
2_CHe	0.97	1.05	0.08	(-0.13, 0.29)
3_CHe	2.34	2.00	-0.34	(-0.64, -0.04)
4_CHe	0.92	0.81	-0.11	(-0.29, 0.07)
5_CHe	0.72	0.81	0.09	(-0.07, 0.25)
6_CHe	0.78	1.03	0.25	(0.09, 0.41)
7_CHe	0.89	0.97	0.08	(-0.07, 0.23)
8_CHe	0.55	0.78	0.23	(0.11, 0.35)
1_CH	2.47	1.77	-0.70	(-0.95, -0.45)
2_CH	2.20	1.51	-0.69	(-0.97, -0.41)
3_CH	3.55	2.75	-0.80	(-1.17, -0.43)
4_CH	1.66	1.24	-0.42	(-0.69, -0.15)
5_CH	1.33	1.40	0.07	(-0.14, 0.28)
6_CH	1.13	1.34	0.21	(0.03, 0.39)
7_CH	1.60	1.55	-0.05	(-0.26, 0.16)
8_CH	1.68	1.34	-0.34	(-0.63, -0.05)
1_CL	0.20	0.26	0.06	(-0.11, 0.23)
2_CL	0.18	0.32	0.14	(-0.03, 0.31)
3_CL	0.33	0.39	0.06	(-0.12, 0.24)
4_CL	0.15	0.22	0.07	(-0.06, 0.20)
5_CL	0.13	0.13	0.00	(-0.11, 0.11)
6_CL	0.09	0.12	0.03	(-0.07, 0.13)
7_CL	0.09	0.16	0.07	(-0.03, 0.17)
8_CL	0.13	0.12	-0.01	(-0.11, 0.09)

(b) Comparison (SP - AI) with 66% waiting time

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	1.01	1.04	0.03	(-0.15, 0.21)
2_He	0.64	1.03	0.39	(0.23, 0.55)
3_He	1.67	1.69	0.02	(-0.21, 0.25)
4_He	0.60	0.42	-0.18	(-0.38, 0.02)
5_He	0.57	0.70	0.13	(-0.02, 0.28)
6_He	0.47	0.67	0.20	(0.03, 0.37)
7_He	0.60	0.91	0.31	(0.16, 0.46)
8_He	0.72	0.90	0.18	(0.05, 0.31)
1_H	0.99	1.19	0.20	(0.00, 0.40)
2_H	0.88	1.25	0.37	(0.17, 0.57)
3_H	0.88	1.25	0.37	(0.17, 0.57)
4_H	0.81	0.43	-0.38	(-0.61, -0.15)
5_H	0.98	1.14	0.16	(0.00, 0.32)
6_H	0.40	0.71	0.31	(0.16, 0.46)
7_H	1.15	1.14	-0.01	(-0.19, 0.17)
8_H	1.06	1.02	-0.04	(-0.23, 0.15)
1_L	0.12	0.20	0.08	(-0.08, 0.24)
2_L	0.13	0.22	0.09	(-0.05, 0.23)
3_L	0.33	0.60	0.27	(0.06, 0.48)
4_L	0.11	0.15	0.04	(-0.08, 0.16)
5_L	0.10	0.10	0.00	(-0.11, 0.11)
6_L	0.07	0.20	0.13	(-0.01, 0.27)
7_L	0.15	0.16	0.01	(-0.12, 0.14)
8_L	0.15	0.19	0.04	(-0.11, 0.19)
1_CHe	0.89	1.01	0.12	(-0.06, 0.30)
2_CHe	0.33	0.74	0.41	(0.26, 0.56)
3_CHe	1.02	1.30	0.28	(0.09, 0.47)
4_CHe	0.63	0.37	-0.26	(-0.44, -0.08)
5_CHe	0.81	0.84	0.03	(-0.13, 0.19)
6_CHe	0.43	0.75	0.32	(0.16, 0.48)
7_CHe	0.68	0.82	0.14	(-0.02, 0.30)
8_CHe	0.61	0.94	0.33	(0.19, 0.47)
1_CH	1.37	1.52	0.15	(-0.07, 0.37)
2_CH	0.46	0.91	0.45	(0.27, 0.63)
3_CH	1.42	1.80	0.38	(0.13, 0.63)
4_CH	0.85	0.52	-0.33	(-0.58, -0.08)
5_CH	0.78	1.00	0.22	(0.06, 0.38)
6_CH	0.45	0.92	0.47	(0.30, 0.64)
7_CH	0.79	1.17	0.38	(0.18, 0.58)
8_CH	0.78	1.28	0.50	(0.30, 0.70)
1_CL	0.18	0.20	0.02	(-0.13, 0.17)
2_CL	0.10	0.17	0.07	(-0.05, 0.19)
3_CL	0.25	0.36	0.11	(-0.08, 0.30)
4_CL	0.09	0.15	0.06	(-0.07, 0.19)
5_CL	0.08	0.14	0.06	(-0.09, 0.21)
6_CL	0.08	0.15	0.07	(-0.06, 0.20)
7_CL	0.05	0.18	0.13	(0.00, 0.26)
8_CL	0.06	0.20	0.14	(0.00, 0.28)

While including additional plans does improve the performance of the SP heuristic, we note that the magnitude of the differences is small. The greatest average difference between AI and SP is 0.88 customers (dataset 3_CH with 50% late-request customers and 33% waiting time) when 50 plans are used. The average difference over all of the 50% late-request customers, 33% waiting time, and 50 plans instances that were tested is 0.18 customers in favor of the SP heuristic. With 75 plans, the average difference is only 0.20 in favor of the SP heuristic.

While the performance of the SP heuristic can be improved by increasing the number of plans, the improvement is obtained at the expenses of the effort needed to choose an action. For this reason, it is also interesting to compare the two heuristics in terms of average time needed to

Table 3 **Results for 25% Late-Request Customers and 33% waiting time**

(a) Comparison (SP - AI) with 50 SP plans

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	0.40	0.65	0.25	(0.11, 0.39)
2_He	0.32	0.70	0.38	(0.24, 0.52)
3_He	1.16	1.31	0.15	(-0.08, 0.38)
4_He	0.32	0.45	0.13	(0.01, 0.25)
5_He	0.35	0.43	0.08	(-0.01, 0.17)
6_He	0.06	0.59	0.53	(0.37, 0.69)
7_He	0.52	0.47	-0.05	(-0.19, 0.09)
8_He	0.26	0.51	0.25	(0.12, 0.38)
1_CH	0.36	0.52	0.16	(0.02, 0.30)
2_CH	0.46	0.57	0.11	(-0.04, 0.26)
3_CH	1.38	1.03	-0.35	(-0.60, -0.10)
4_CH	0.50	0.21	-0.29	(-0.45, -0.13)
5_CH	0.28	0.47	0.19	(0.09, 0.29)
6_CH	0.10	0.63	0.53	(0.36, 0.70)
7_CH	0.53	0.72	0.19	(0.05, 0.33)
8_CH	0.34	0.50	0.16	(0.04, 0.28)

(b) Comparison (SP - AI) with 75 SP plans

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	0.40	0.67	0.27	(0.12, 0.42)
2_He	0.32	0.72	0.40	(0.26, 0.54)
3_He	1.16	1.20	0.04	(-0.31, 0.39)
4_He	0.32	0.49	0.17	(0.05, 0.29)
5_He	0.35	0.53	0.18	(0.08, 0.28)
6_He	0.06	0.56	0.50	(0.35, 0.65)
7_He	0.52	0.47	-0.05	(-0.18, 0.08)
8_He	0.26	0.46	0.20	(0.09, 0.31)
1_CH	0.36	0.43	0.07	(-0.06, 0.20)
2_CH	0.46	0.61	0.15	(-0.05, 0.35)
3_CH	1.38	1.13	-0.25	(-0.47, -0.03)
4_CH	0.50	0.15	-0.35	(-0.52, -0.18)
5_CH	0.28	0.44	0.16	(0.05, 0.27)
6_CH	0.10	0.68	0.58	(0.41, 0.75)
7_CH	0.53	0.73	0.20	(0.06, 0.34)
8_CH	0.34	0.58	0.24	(0.13, 0.35)

Table 4 **Results for 50% Late-Request Customers and 33% waiting time - Average number of requests not served**

(a) Comparison (SP - AI) with 50 SP plans

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	1.10	1.12	0.02	(-0.15, 0.19)
2_He	1.00	1.07	0.07	(-0.11, 0.25)
3_He	2.13	2.13	0.00	(-0.25, 0.25)
4_He	0.57	0.32	-0.25	(-0.42, -0.08)
5_He	0.77	0.89	0.12	(-0.01, 0.25)
6_He	0.61	0.83	0.22	(0.09, 0.35)
7_He	0.96	1.10	0.14	(-0.05, 0.33)
8_He	1.25	1.06	-0.19	(-0.40, 0.02)
1_CH	2.47	1.79	-0.68	(-0.94, -0.42)
2_CH	2.20	1.48	-0.72	(-1.04, -0.40)
3_CH	3.55	2.67	-0.88	(-1.28, -0.48)
4_CH	1.66	1.25	-0.41	(-0.69, -0.13)
5_CH	1.33	1.35	0.02	(-0.17, 0.21)
6_CH	1.13	1.40	0.27	(0.07, 0.47)
7_CH	1.60	1.43	-0.17	(-0.40, 0.06)
8_CH	1.68	1.27	-0.41	(-0.70, -0.12)

(b) Comparison (SP - AI) with 75 SP plans

Set	AI	SP	Δ SP - AI	SP - AI CI
1_He	1.10	1.08	-0.02	(-0.18, 0.14)
2_He	1.00	1.08	0.08	(-0.11, 0.27)
3_He	2.13	2.30	0.17	(-0.10, 0.44)
4_He	0.57	0.22	-0.35	(-0.53, -0.17)
5_He	0.77	0.86	0.09	(-0.03, 0.21)
6_He	0.61	0.84	0.23	(0.09, 0.37)
7_He	0.96	1.00	0.04	(-0.16, 0.24)
8_He	1.25	0.98	-0.27	(-0.46, -0.08)
1_CH	2.47	1.72	-0.75	(-1.03, -0.47)
2_CH	2.20	1.42	-0.78	(-1.13, -0.43)
3_CH	3.55	2.88	-0.67	(-1.05, -0.29)
4_CH	1.66	1.25	-0.41	(-0.69, -0.13)
5_CH	1.33	1.25	-0.08	(-0.29, 0.13)
6_CH	1.13	1.31	0.18	(-0.02, 0.38)
7_CH	1.60	1.32	-0.28	(-0.51, -0.05)
8_CH	1.68	1.31	-0.37	(-0.64, -0.10)

choose an action (Table 5). For AI, an action consists of the decision of which new requests to insert and if and where to wait. For SP, the decision concerns which new requests to insert, the generation of new plans and the determination of the distinguished plan. Given that comparable times were obtained for all the different classes of instances, we report, for the sake of brevity, only the results related to datasets _He and _CH. Action selection for AI is nearly instantaneous and is omitted as it would be reported as 0.0. Our results show that SP takes, on the average, 3.4, 7.0, and 10.7 seconds in the case of 25, 50, and 75 plans, respectively. These numbers show how the computing times increase as the number of SP plans increases. While these times are likely feasible in a real-time decision-making environment, the results presented in Tables 3 and 4 suggest that the greater effort is worthwhile only for companies operating in environments with a high degree of dynamism.

6. Conclusions

In this paper, we study a dynamic and stochastic routing problem in which a single, uncapacitated vehicle serves a set of known customers locations, but in which some customers request service

Table 5 Average times (in seconds) to choose an SP action. Results with 50% late-request customers and 33% waiting time

Set	Time 25 SP Plans	Time 50 SP Plans	Time 75 SP Plans
1_He	3.3	6.6	10.3
2_He	4.2	8.6	13.6
3_He	5.1	10.4	16.2
4_He	3.7	7.6	12.0
5_He	2.1	4.4	6.7
6_He	2.4	4.9	7.8
7_He	2.2	4.4	6.7
8_He	2.1	4.3	6.6
1_CH	4.3	8.9	13.4
2_CH	5.1	10.3	15.2
3_CH	6.2	12.6	18.8
4_CH	3.4	6.6	10.6
5_CH	2.3	4.7	7.2
6_CH	3.3	6.5	10.2
7_CH	2.5	5.2	7.8
8_CH	2.7	5.4	8.4

while the vehicle is en route. We compare anticipatory insertion and sample-scenario planning to asses the value that can be gained by allowing a solution free of restriction. In the process of our comparison, we first show that a waiting strategy first developed for anticipatory insertion can be used with sample-scenario planning. The comparison between anticipatory insertion and sample-scenario planning also shows that, when the degree of dynamism is relatively low, sample-scenario planning requires many more samples in the construction of the distinguished tour to outperform anticipatory insertion. A similar observation can be made in the case in which the probability of dynamic service requests is low.

These results suggest that the computational burden of sample-scenario planning has greater value as an instance's degree of dynamism increases and as the likelihood of dynamic customer requests increases. Yet, even then, the value of the sample-scenario planning approach may be limited as the magnitude of the performance improvement is small. As noted earlier though, our results demonstrate a best case improvement for the SP heuristic averaging only 0.2 more customers. Further, achieving the improved performance requires significant computing power in order to achieve decision times that are reasonable for real-time implementation.

Thus, the trade-off between the relatively small performance improvement and the cost of computing hardware becomes an important managerial consideration. To gain insight into this trade-off, we contacted a driver and fleet operations manager, who wished to remain anonymous, at a major U.S. less-than-truckload carrier. Our contacted noted that the addition of one stop to a tour per day represents an average of a 5% improvement under current practice. While such an improvement would be impressive given the thin margins of the less-than-truckload industry, our results show SP is capable of generating only a fraction of that improvement relative to AI.

The work in this paper suggests several areas for future work. First, the work presented here does not consider time windows. With time windows, the solution space would be greatly reduced and likely leading to improved performance for sample-scenario planning. In addition, this work considered only a single vehicle. Work examining a fleet of vehicles may also demonstrate different results from those presented here. Third, as noted previously, the sample-scenario planning heuristic could potentially benefit from fixing an initial route for the early decision epochs and then transitioning to the dynamic tours. Fourth, because of the difficulty in solving the subproblem associated with generating the routing plans for sample-scenario planning, we were forced to use a heuristic to generate the routing plans. It would be interesting to apply sample-scenario planning in circumstances in which the quality of the subproblem solutions could be assessed. Such an analysis would indicate the value associated with the quality of the subproblem solutions. Finally, as discussed in Section 2, the relationship between the AI heuristic and a priori routes suggests the possibility that the route structure of the AI-generated solutions might offer some level of consistency to the routes when considered from day to day. Work that seeks to answer this question could be valuable.

Acknowledgments
This research was partially supported by the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) of Italy. This support is gratefully acknowledged. The authors also thank three anonymous referees for their useful comments, which helped to improve the paper.

References

Bent, R., P. Van Hentenryck. 2004. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research* **52** 977–987.

Benton, W. C., Manuel D. Rosetti. 1992. The vehicle scheduling problem with intermittent customer demands. *Computers and Operations Research* **19** 521–531.

Branke, J., M. Middendorf, G. Noeth, M. Dessouky. 2005. Waiting strategies for dynamic vehicle routing. *Transportation Science* **39** 298–312.

Dumas, Y., J. Desrosiers, E. Gelinas, M.M. Solomon. 1995. An optimal algorithm for the traveling salesman problem with time windows. *Operations Research* **43** 367–371.

Feo, T.A., M.G.C. Resende. 1995. Greedy randomized adpative search procedures. *Journal of Global Optimization* **6** 109–134.

Gendreau, M., F. Guerten, J.-Y. Potvin, É. Taillard. 1999. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science* **33** 381–390.

Ghiani, G., E. Manni, A. Quaranta, C. Triki. 2009. Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E* **45** 96–106.

Goodson, J., J.W. Ohlmann, B.W. Thomas. 2010. New rollout policies for approximate dynamic programming with application to the vehicle routing problem with stochastic demand and duration limits. Submitted for publication.

Haughton, Michael A. 2000. Quantifying the benefits of route reoptimisation under stochastic customer demand. *Journal of the Operational Research Society* **51** 320–332.

Haughton, Michael A. 2002. Route reoptimization's impact on delivery efficiency. *Transportation Research - Part E* **38** 53–63.

Hvattum, L.M., A. Løkketangen, G. Laporte. 2006. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science* **40**(4) 421–438.

Ichoua, S., M. Gendreau, J.-Y. Potvin. 2000. Diversion issues in real-time vehicle dispatching. *Transportation Science* **34** 426–438.

Ichoua, S., M. Gendreau, J.-Y. Potvin. 2006. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science* **40** 211–225.

Kilby, P., P. Prosser, P. Shaw. 1998. Dynamic VRPs: A study of scenarios. Tech. Rep. APES-06-1998, Department of Computer Science, Strathclyde University.

Larsen, A., O.B.G. Madsen, M.M. Solomon. 2002. Partially dynamic vehicle routing - models and algorithms. *Journal of the Operational Research Society* **53** 637–646.

Larsen, A., O.B.G. Madsen, M.M. Solomon. 2004. The a-priori dynamic traveling salesman problem with time windows. *Transportation Science* **38** 459–472.

Law, A.M., W.D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd ed. McGraw-Hill, Boston.

Manni, E. 2007. Topics in real-time fleet management. Ph.D. thesis, Università Della Calabria.

Manni, E., J.W. Ohlmann, B.W. Thomas. 2009. Variable neighborhood search with scheduled penalty. Working Paper.

Meisel, S., U. Suppa, D. Mattfeld. 2009. GRASP based approximate dynamic programming for dynamic routing of a vehicle. S. Voss, M. Caserta, eds., *Proceedings of MIC 2009: The VIII Metaheuristics International Conference*. Hamburg, Germany, to appear.

Mitrović-Minić, S., R. Krishnamurti, G. Laporte. 2004. Double-horizon based heurisitcs for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B* **38** 669–685.

Mitrović-Minić, S., G. Laporte. 2004. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B* **38** 635–655.

Nahmias, S. 2001. *Production and Operations Analysis*. 4th ed. Irwin/McGraw-Hill, Boston.

Ohlmann, J.W., B.W. Thomas. 2007. A compressed annealing approach to the traveling salesman problem with time windows. *INFORMS Journal on Computing* **19** 80–90.

Powell, W.B., P. Jaillet, A. Odoni. 1995. Stochastic and dynamic networks and routing. M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, eds., *Network Routing*, *Handbooks in Operations Research and Management Science*, vol. 8. Elsevier Science, Amsterdam, 141–295.

Psaraftis, H.N. 1988. Dynamic vehicle routing problems. B.L. Golden, A.A. Assad, eds., *Vehicle Routing: Methods and Studies*. North-Holland, Amsterdam, 223–248.

Psaraftis, H.N. 1995. Dynamic vehicle routing: Status and prospects. *Annals of Operations Research* **61** 143–164.

Savelsbergh, Martin W. P., M. Goetschalckx. 1995. A comparison of the efficiency of fixed versus variable vehicle routes. *Journal of Business Logistics* **46** 474–490.

Secomandi, N. 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research* **27** 1201–1225.

Secomandi, N. 2001. A rollout policy for the vehicle routing policy with stochastic demands. *Operations Research* **49** 796–802.

Secomandi, N., F. Margot. 2009. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research* **57**(1) 214–230.

Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with time windows. *Operations Research* **35** 254–265.

Stefik, M. 1981. Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence* **16** 111–139.

Thomas, B.W. 2007. Waiting strategies for anticipating service requests from known customer locations. *Transportation Science* **41**(3) 319–331.

van de Klundert, J., L. Wormer. 2010. ASAP: The after-salesman problem. *Manufacturing & Service Operations Management* DOI: 10.1287/msom.1100.0292.

van Hemert, J.I., J.A. La Poutré. 2004. Dynamic routing with fruitful regions: Models and evolutionary computation. X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tino, A. Kabáan, H.-P. Schwefel, eds., *Parallel Problem Solving from Nature VIII*. Springer-Verlag, 690–699.

Waters, C. D. J. 1989. Vehicle scheduling problems with uncertainty and omitted customers. *Journal of the Operational Research Society* **40** 1099–1108.