A Look-ahead Partial Routing Framework for the Dynamic Vehicle Routing Problem

Han Zou and Maged M. Dessouky

Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California,

3715 McClintock Avenue, GER240, Los Angeles, CA 90089-0193, United States

Han Zou hanzou@usc.edu, Maged M. Dessouky maged@usc.edu

Abstract

In this paper, we study the vehicle routing problem with dynamic customers, where a portion of the customer requests are known in advance and the rest arrive in real time. We propose an optimization-based look-ahead dynamic routing framework that involves request forecasting, partial planning, and dynamic real-time routing of the fleet. This framework has the capabilities for adjustments in response to routing environments with different levels of uncertainties. Through extensive numeral simulations, we exam its performance in routing environments with various levels of uncertainties. We demonstrate the efficiency and robustness of the proposed solution by benchmarking against two other routing strategies. This paper fills the gap in the literature on studying the relationship between the level of route planning in the solution approach and the quality of the solution under various system conditions.

Key Words: Dynamic vehicle routing problem, Look-ahead dynamic routing, Re-optimization, Partial routing, Waiting time adjustment

1 Introduction

Many industries deal with the task of transporting goods or delivering services in a timely, reliable, and cost-effective manner, including manufacturing, food, e-commerce, public transit, etc. Logistics has become the backbone that enables the productivity and mobility of these industries [16]. Indeed,

growth in the transportation sector recently has been on par with the growth in the Gross Domestic Product (GDP) in the United States. According to statistics from the 2013 National Transportation Statistics report [17], expenditure on transportation activities amounted to 1,426 billion dollars in 2012, representing nearly 9 percent of the total US GDP. The transportation industry, like many others, has undergone significant changes in the last decade through the introduction of information technologies. Examples include vehicle tracking, such as global-positioning-systems (GPS), wireless communication via satellite, cellular and paging networks, which enable 2-way communication with mobile fleets, and real-time information services that allow for dynamic estimation of travel times. Whereas in the past it was difficult for a logistics company to control or route vehicles once they left the depot, these technologies make accurate dynamic real-time routing a very real possibility.

However, most of the developed techniques and models for planning, routing and scheduling assume "known" static data as their input, and have yet to take full advantage of the technological advances mentioned above. For instance, in the Vehicle Routing Problem (VRP) the customer demands, travel costs, and travel times are known in advance. In this case, the fundamental problem is to determine the optimal route that minimizes a certain objective such as fleet size and total travel distance. The built-in assumption of these approaches is that there will be small deviations on the realization of the demand and travel times from the plan so that the pre-determined routes form a basis for either the pickup or delivery schedule. In the real world, however, operations in any transportation network contain a fairly high level of uncertainties including variable waiting and travel times due to traffic congestion, arrival of new service requests, cancellation of existing requests, unknown demand sizes, etc. That is why human operators (dispatchers) still play a major role in route planning and vehicle scheduling in the trucking industry.

The problem of routing a fleet of vehicles in real time to serve a set of customers under changing and gradually revealed information falls in the scope of the Dynamic Vehicle Routing Problem (DVRP). The DVRP derives from the VRP when some element of the problem becomes non-deterministic. The DVRP has emerged as an active and intense area of research, both due to industry needs, but also due to technological advances, including map databases, location determination technology (e.g., GPS), wireless communication and mobile computing. In some highly uncertain environments, information concerning the randomness in the problem may not be available and pre-planned optimal routes are no longer of practical use. A reactive approach must be adopted to constantly re-route the fleet in light of newly revealed information. In other cases,

some stochastic information on network conditions and customer requests may be obtained from historical data. For these situations, it is widely expected that the use of information technology in transportation systems narrows the gap between highly uncertain systems in reality and the perfectly known static systems in theory.

As discussed above, for static systems, where the network parameters are known and fixed, the well-established routing and scheduling algorithms lead to optimum solutions. On the other hand, in a highly uncertain system where no stochastic information on the randomness of the problem is available, the reactive routing approach is the only option. Therefore, there exists a gap in the literature for situations that are in between these two extreme cases, where some stochastic information concerning the random system is available. An ideal approach to solve these situations should have the flexibility to adjust the level of route planning in the solution based on the level of uncertainties in the system. To address this gap, there is a need to study the relationship between the amount and quality of information available in the dynamic routing problem and the level of route planning that should be implemented in order to generate an efficient and reliable solution.

In this paper, we focus on studying a category of the dynamic vehicle routing problem with dynamic customers. The objective of this research is to develop a routing technique that involves partial routing and has the capabilities for adjustments in response to problems with different levels of uncertainties. We propose an optimization-based look-ahead dynamic routing framework that employes time-efficient heuristic algorithms. In order to tackle problems with various levels of uncertainties, the behavior of the proposed model can be adjusted by changing multiple parameter settings. We conduct extensive numerical simulations to find the desirable level of route planning in the solution approach in response to different levels of uncertainties in the problem for the best performance. Our analysis sheds insights into how dynamic real-time routing would narrow the gap between highly uncertain systems in reality and the perfectly known static system in theory. This paper also fills the gap in the literature on studying the relationship between the level of route planning in the solution approach and the quality of the solution under various system conditions.

The rest of the paper is organized as follows. In Section 2, a literature review of the dynamic vehicle routing problem is presented. Section 3 formally defines the problem and illustrates the solution framework. Section 4 presents the experimental setup and results. We conclude in Section 5.

2 Literature Review

The traditional vehicle routing problem (VRP) has always received significant attention in the literature ever since its first introduction by Dantzig and Ramser in 1959 [5]. A traditional VRP is based on a graph, with a special node representing the depot, and the remaining nodes representing customers. A cost matrix is defined on the arcs to represent the travel costs (usually proportional to travel distance) between corresponding locations. A fleet of vehicles originally located at the depot are routed to service the customers. The objective is to find a feasible routing schedule that visits each customer exactly once with minimum total travel cost. Feasibility is often defined with respect to side constraints, which may include vehicle capacity constraint, time window constraint, service level constraint, etc.

The DVRP differentiates from the VRP in that some element of the problem is random, and is not known with certainty at the time the vehicle routes must be planned. The problem arises naturally from a broad spectrum of real-world applications, including courier routing [8, 13], service scheduling [4, 3], Dial-a-Ride systems [7, 2, 23, 22], etc. Depending on which element or elements of the problem become dynamic, numerous variations of the DVRP exist. For example, the set of customers that needs to be serviced may not be known in advance. Instead, new customers may arrive in real time throughout the planning horizon [13, 9, 23]. In some cases, the demand of a customer may not be known when the service request is made and when routing decisions have to be made. Instead, the actual size of the demand may only be revealed when the vehicle reaches the customer [25, 27, 14]. The cost matrix can also be random, reflecting random travel times between customer locations due to varying traffic conditions or uncertainties in operations [29, 30]. There are many other potential sources of randomness that could make a problem dynamic. The variation of DVRP that is particularly of interest is the vehicle routing problem with dynamic customers, where new customers arrive in real time throughout the planning horizon. In the dynamic environment, critical problem information is revealed over time, meaning that the complete realization of randomness is only known at the end of the planning horizon. As a consequence, the initial solution can only be constructed based on partial information at the beginning of the planning horizon. The set of routes must be updated (if possible) in real time as new information becomes available. This cannot be done without the help of real-time vehicle positioning and communication technologies. Due to recent advances in these technologies, they can now be implemented at lower costs and at a larger scale [19]. One can refer to [12, 6, 19] for complete reviews on the recent

DVRP literature.

Solution approaches for the DVRP can be classified into three categories, namely static routing, local dynamic routing, and look-ahead dynamic routing [4]. In the static routing approach, *a priori* vehicle routes or routing policies are constructed with limited information at the beginning of the planning horizon, before vehicles begin to travel. As new information becomes available, existing routes adapt automatically according to pre-defined rules. In the local dynamic approach, route planners react to new information by explicitly incorporating them into decision making. Thus vehicles often need to be diverted and re-routed during the planning horizon. In the look-ahead dynamic approach, route planners not only react to new information, but also forecast future events and the fleet status, and explicitly use predictions to help design vehicle routes. Forecasts are usually made based on historical information. The latter two approaches rely on real-time vehicle positioning technologies and real-time communication systems between each vehicle and route planners. In this paper, we propose to develop a look-ahead dynamic routing approach.

One dynamic routing technique that has received significant attention in the literature is re-optimization. The intuition behind re-optimization is to repeatedly and sequentially formulate static vehicle routing problems based on newly revealed information throughout the planning horizon. And solve these problems using well-studied static VRP algorithms. Depending on when static problems are formulated and solved, this approach involves into either the periodic re-optimization approach or the continuous re-optimization approach. In periodic re-optimization, an optimization procedure runs at the beginning of the planning horizon to construct an initial set of vehicle routes. Then, an optimization procedure is invoked periodically to solve the current state static problem, whenever new problem information becomes available, or at fixed intervals of time. Such fixed intervals are referred to as decision epochs or time slices in the literature [11, 4, 19].

The first periodic re-optimization technique was introduced by Psaraftis in 1980 [20]. The author utilized a local-dynamic approach to solve the vehicle routing problem with dynamic customers. In particular, a static VRP is formulated whenever a new customer requests service, and is solved to optimality by a dynamic programming algorithm. This approach inevitably suffers from the curse of dimensionality of dynamic programming, which prevents its application to large instances. Several streams of research follow the lead in developing periodic re-optimization frameworks embedded with exact algorithms. Chen and Xu considered a dynamic vehicle routing problem with hard time windows [4]. The authors assumed that the dispatcher does not have any deterministic or

probabilistic information on the location and the size of a customer order until it arrives. A periodic re-optimization framework embedded with a dynamic column-generation-based algorithm was developed. The approach showed its merits when compared with insertion-based heuristics on most problems.

Other researchers have focused on developing heuristic algorithms. Several Metaheuristics were proposed to be combined with the periodic re-optimization framework. Montemanni et al. developed an Ant Colony System (ACS) to solve the vehicle routing problem with dynamic customers [15]. One feature of the solution is to hold dynamic customers that arrive within a time period until the end of that period. This limitation is certainly not desirable in situations where an immediate or at least a timely response to customer requests is crucial. Secomandi and Margot studied a vehicle routing problem with stochastic demands [24]. The actual demand is only known when the vehicle arrives at the customer. The authors developed a finite-horizon Markov Decision Process (MDP) formulation for the single vehicle case. A partial re-optimization heuristic is proposed to solve the MDP. The authors compared multiple heuristics to embed in the re-optimization framework. They argued that their best approach outperforms existing heuristics.

One major limitation of the re-optimization approach lies in the fact that all optimization needs to be performed before the decision maker can update each vehicle with its new route, potentially causing delays in routing operations. One possible solution is to employ computationally fast heuristic algorithms instead of exact algorithms in the re-optimization framework.

3 Problem Definition and Solution Framework

In this section, we formulate and solve the vehicle routing problem with dynamic customer requests. In particular, we develop a look-ahead dynamic partial routing framework that involves demand forecasting, partial planning, dynamic real-time routing, and periodic re-optimization of the current schedule. In this section, we first formally define the problem and introduce the notations. We then illustrate components of the proposed framework together with details about how vehicles are routed dynamically based on partial routing schedules. This section concludes with a detailed explanation of all the heuristic algorithms used in the framework.

3.1 Problem Definition

Suppose that the operation consists of routing a fleet of capacitated vehicles to collect shipments from a set of customers and transport them to a central depot. The length of the planning horizon is T_{max} and can be discretized into time steps of unit length. There are N potential customers. Each customer has a fixed location, a known demand size, a known service time window and a service time of fixed length. The service time window specifies the earliest and latest times when service can be started at the corresponding customer and cannot be violated. Each customer requests service at most once during the planning horizon. The uncertainty lies in the fact that not all customers would request service. Some customers request service in advance (prior to the beginning of the planning horizon), and are called advance customers. These customers must be served. The rest of the customers are called dynamic customers, who may or may not request service during the planning horizon. We assume that the probability a dynamic customer requests service can be estimated from historical information. The time when a dynamic customer requests service is called its request time. It is also the time when it becomes certain that the customer needs to be served. Dynamic customer requests are not guaranteed to be accommodated due to potentially insufficient fleet capacity. The objective is two-fold: minimizing the total travel distance of all vehicles and minimizing the number of rejected dynamic customer requests.

The following notations are used for model parameters and decision variables. Generally, i and j are used to index customers, k to index vehicles/routes, and t to index time.

$\mathcal{N}$	total number of customers
$\mathbb{AC}$	set of advance customers
$\mathbb{DC}$	set of dynamic customers
d_i	demand of customer i
s_i	service time of customer i
u_i	request time of customer i
e_i	the earliest time that service can begin at customer i
l_i	the latest time that service can begin at customer i
$t_{i,j}$	distance (minimum travel time) between location i and j
$\mathcal{K}$	total number of vehicles
$\mathcal{C}$	capacity of each vehicle
$r_{k,t}$	partial routing schedule for vehicle k at time t

$n_{i,k,t}$	the i-th customer scheduled on vehicle k at time t
a_i	time of arrival at customer i
b_i	time of departure from customer i
$n_{0,k,t}$	the location from where vehicle k would start its new route if diverted at time t
$a_{0,k,t}$	the time when vehicle k would become available to start its new route if diverted at time t

It is assumed that all vehicles travel at unit speed. Thus, the travel time is equatable with travel distance between corresponding locations. It is also assumed that no preemption in vehicle routes is allowed, meaning that a vehicle cannot be diverted while en route to its current scheduled customer. The vehicle can only be diverted after it reaches and finishes service at its current customer. The request time u_i of dynamic customer i represents the time when it becomes certain that customer i needs to be serviced. u_i is modeled as a random variable taking values on the interval $[0, e_i]$. It means that the customer must make the decision on whether it needs to be serviced or not before the beginning of the its service time window. In addition, we assume that real-time two-way communication capability is established between the central decision making unit and each vehicle. At any point in time, the decision maker is aware of the complete fleet status including current locations, directions, and remaining capacities. This enables dynamic real-time routing of the vehicles.

There are two issues that are uncertain about dynamic customer requests. First, whether the customer requests service at all during the planning horizon. Second, when will the customer request service given that it will do so. From a historical perspective, the probability that a customer requests service on any day can be estimated by the proportion of days that the customer has requested service among all the days of operation. We use q_i to denote this probability. For the second issue, a distribution on request time can be estimated by the actual request times of the customer on the days when it actually requested service. By definition, this distribution is conditional on the fact that the customer requests service. Let $f_i(t)$ be the conditional probability density function of request time u_i. Recall that u_i is defined on $[0, e_i]$, thus we have $\int_0^{e_i} f_i(t)dt = 1, \forall i$. Given this setup, the probability that a dynamic customer i requests service during the time interval $[t_1, t_2], 0 \leq t_1 \leq t_2 \leq T_{max}$ on any day can be calculated as

$$P\left(i \text{ requests during } [t_1, t_2]\right) = P\left(i \text{ requests, } i \text{ requests during } [t_1, t_2]\right) \quad (1)$$

$$= P\left(i \text{ requests during } [t_1, t_2] \mid i \text{ requests}\right) * P\left(i \text{ requests}\right) \quad (2)$$

$$= \int_{t_1}^{t_2} f_i(t)dt * q_i \tag{3}$$

In a dynamic vehicle routing context, problem information are revealed gradually over time. In other words, the full set of customers cannot be know until the end of the planning horizon. At any time t in the planning horizon, only the set of advance customers and a subset of dynamic customers who have already requested service are known. A problem consisting of only partial information is called a partial vehicle routing problem P_t. The solution to a partial problem at time t is called a partial solution S_t, which consists of a collection of partial routing schedules, $S_t = \{r_{k,t}\}$ where $k = 1, \ldots, \mathcal{K}$. In the dynamic context, the sequence of customers alone does not uniquely determine an operational schedule. In addition, we need to specify the exact arrival and departure times at each location along the route. Let a_i and b_i denote the arrival and departure times at customer i respectively. A partial routing schedule for vehicle k specifies the sequence of customers scheduled for the vehicle, together with the arrival and departure times at each customer. $r_{k,t} = \{n_{1,k,t}, \ldots, n_{|r_{k,t}|,k,t}, n_{|r_{k,t}|+1,k,t}\}$ where $|r_{k,t}|$ denotes the total number of customers scheduled on route k at time t. $n_{|r_{k,t}|+1,k,t} = 0, \forall k, t$ is a dummy place holder variable representing the constraint that all vehicles must return to the depot by the end of the planning horizon.

$n_{0,k,t}$ denotes the location from where vehicle k would start its new route if it was diverted at time t. It can be loosely interpreted as the "available position" of vehicle k. At any time t, vehicle k must be in exactly one of the following two states. State I: serving or idling at some customer i. State II: en route to some customer i. In either case, if a new routing schedule were to be constructed at the moment and the vehicle is diverted, the new route must start at location i (no preemption assumption). Hence in either case, we have $n_{0,k,t} = i$. In fact, during the implementation of vehicle routes, the $n_{0,k,t}$ variable should be updated once the vehicle starts to travel to its next customer based on the no preemption rule.

Similarly, $a_{0,k,t}$ denotes the time when vehicle k would become available to start its new route if it were to be diverted at time t. It can be loosely interpreted as the "available time" of vehicle k. At any time t, if vehicle k is currently servicing customer i, then $a_{0,k,t} = a_i + s_i$; if the vehicle is idle, then $a_{0,k,t} = t$; if the vehicle is traveling to service customer i, then $a_{0,k,t} = a_i + s_i$.

3.2 Look-ahead Partial Vehicle Routing Framework

3.2.1 Customer States

At any time during the planning horizon, each customer belongs to exactly one of the following 5 customer states.

Unconfirmed customer $\mathbb{U}(t)$**.** Customers who have yet to request service, and are not anticipated to request in the near future. This is the initial state for all dynamic customers.

Confirmed customer $\mathbb{C}(t)$**.** Customers who have requested service and are accepted, but not yet serviced. This is the initial state for all advance customers and the state for a dynamic customer once it requests service and is accepted.

Serviced customer $\mathbb{S}(t)$**.** Customers who have been serviced.

Rejected customer $\mathbb{R}(t)$**.** Customers who have requested service, but have been rejected due to infeasibility in the routing schedule.

Anticipated customer $\mathbb{A}(t)$**.** Customers who have yet to request service, but are anticipated to do so soon. The set of anticipated customers is updated at each decision epoch (Section 3.2.2).

The state of each customer changes over time. State changes are triggered by certain events in the dynamic partial routing environment and will be introduced in the following sections.

3.2.2 Decision Epoch

Decision epochs are the key component of the re-optimization scheme. Figure 3.1 illustrates the time dynamic of events in the system. The entire planning horizon is divided equally into a number of time periods. The beginning of each time period is called a decision epoch. By construction, the first decision epoch occurs at time 0. At each decision epoch, four solution procedures are called sequentially to construct and solve a partial vehicle routing problem. The resulting partial routing schedule is implemented based on pre-defined rules until the next decision epoch (when the partial schedule is updated), or when the end of the planning horizon is reached.

Figure 3.2 illustrates how partial solutions are constructed at each decision epoch. In the first step, a partial vehicle routing problem consisting of both confirmed and anticipated customers is formulated. The set of anticipated customers is constructed by a request forecast procedure. Let $forecastHorizon$ be an adjustable parameter representing how far we forecast into the planning horizon. Suppose the current time is t^* and dynamic customer i has yet to request service. We want to calculate how likely it is for the customer to request service within the $forecastHorizon$.

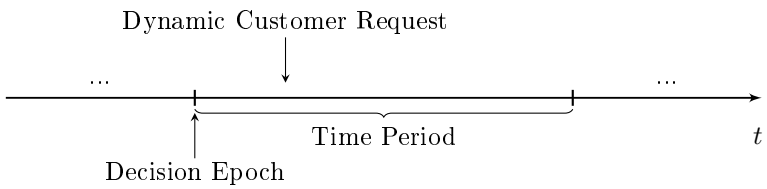

Figure 3.1: Time dynamic of events

Let p_{i,t^*} denote this probability. Then we have

$$p_{i,t^*} = P\left(i \text{ requests during } [t^*, t^* + forecastHorizon] \mid i \text{ hasn't requested till } t^*\right) \tag{4}$$

$$= \frac{P\left(i \text{ requests during } [t^*, t^* + forecastHorizon], i \text{ hasn't requested till } t^*\right)}{P\left(i \text{ hasn't requested till } t^*\right)} \tag{5}$$

$$= \frac{P\left(i \text{ requests during } [t^*, t^* + forecastHorizon], i \text{ hasn't requested till } t^*\right)}{1 - P\left(i \text{ requests during } [0, t^*]\right)} \tag{6}$$

The numerator represents the joint probability of events "i requests during $[t^*, t^* + forecastHorizon]$" and "$i$ hasn't requested till t^*". In fact, the first event completely contains the second event. Given the assumption that each dynamic customer requests service at most once during the planning horizon, the fact that customer i requests in time interval $[t^*, t^* + forecastHorizon]$ implies that the customer must have not requested prior to time t^*. Thus equation 6 can be written as

$$p_{i,t^*} = \frac{P\left(i \text{ requests during } [t^*, t^* + forecastHorizon]\right)}{1 - P\left(i \text{ requests during } [0, t^*]\right)} \tag{7}$$

$$= \frac{\int_{t^*}^{t^* + forecastHorizon} f_i(t)dt}{1 - \int_0^{t^*} f_i(t)dt} * q_i \tag{8}$$

where the last step follows from equation 3. Given p_{i,t^*} for each dynamic customer i who has yet to request service, we use a simply threshold rule to select the set of anticipated customers. Namely, customer i is placed in the anticipated set if and only if $p_{i,t^*} \geq threshold$, where $threshold$ is a tunable model parameter.

A parallel construction heuristic is implemented to construct an initial feasible solution to the partial vehicle routing problem. Both confirmed and anticipated customers are routed. The local search heuristic follows to iteratively improve the initial solution. Last but not least, the hybrid waiting time adjustment heuristic re-distributes slack time along each vehicle route to maximize the chance of accommodating dynamic customers when they actually request service. In essence,

a combination of push backward and push forward procedures is used to position the maximum amount of slack time possible immediately prior to the time when the vehicle must leave for the first anticipated customer on the route (if there is any). The hybrid heuristic ensures that (after finishing service at the previous customer) each vehicle waits for an anticipated customer to actually request service for the maximum amount of time possible while maintaining time window feasibility at all subsequent customers. Details of all heuristic algorithms are presented in Section 3.3.

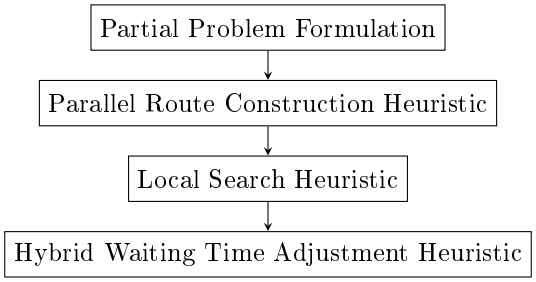

Figure 3.2: Partial solution construction at decision epochs

3.2.3 Dynamic Routing

In our model, vehicles are routed dynamically in real time based on partial routing schedules and newly revealed problem information. The schedules generated at a decision epoch are used until the next decision epoch, when the schedules are updated. A partial routing schedule specifies the sequence of customers scheduled on a particular vehicle together with the arrival and departure times at each customer. For example, the schedule may specify that vehicle k begins to travel to customer i at some future time t^*. When time t^* is reached, one of the following two cases must be true. Case I: customer i is a confirmed customer at t^*. In this case, the vehicle begins to travel to the customer and the customer's state is updated to "serviced" and $n_{0,k,t} = i$. Both updates are consequential to the no preemption rule. Case II: customer i is an anticipated customer at t^*. Since the customer has not requested service and the vehicle has waited for the maximum amount of time for it to do so (Sections 3.3.3), the anticipated customer is dropped from the route and the hybrid waiting time adjustment heuristic adjusts the waiting time for all remaining customers on the same route. Based on the new schedule, the vehicle either remains idle at its current location ($n_{0,k,t}$ is not updated) or immediately begins to travel to its next customer ($n_{0,k,t}$ and the customer's state are updated similarly as in Case I).

When a dynamic customer requests service, it may or may not have a reserved time slot in the

current schedule. The dynamic customer may have a reserved time slot because it was anticipated and routed at the previous decision epoch. In this case, the reserved slot is confirmed immediately and the customer becomes a confirmed customer. Otherwise, the customer does not have a reserved time slot in the schedule either because the customer was not anticipated and routed or because its time slot has expired (corresponding to case II discussed above). In this case, a cheapest insertion heuristic is used to route the customer. If no feasible schedule can be found, the customer is rejected.

3.3 Heuristics

3.3.1 Construction Heuristic

At each decision epoch, the construction heuristic generates an initial feasible solution to the partial vehicle routing problem consisting of both confirmed and anticipated customers. An iterative insertion-based heuristic that constructs all vehicle routes in parallel is implemented. This heuristic aims to minimize both total travel distance and the number of vehicles used. At each iteration, an impact measure is calculated to estimate the inconvenience of routing each customer at each feasible position in the partial solution. Then the customer with the lowest impact measure is placed at the corresponding position with minimum impact. The heuristic initiates with the set of confirmed customers. After all confirmed customers are routed, the procedure continues with the set of anticipated customers.

We have adopted the definition of the impact measure by Solomon and Ioannou [10, 28] as the insertion criteria. Let $impact\,(i, r_k, n_{j,k})$ denote the impact measure of placing customer i on route k at the position immediately prior to customer $n_{j,k}$. i belongs to the set of un-routed customers in the current iteration, $1 \leq k \leq \mathcal{K}$, $1 \leq j \leq |r_k|$. Note that we have omitted the time subscript of variables r_k and $n_{j,k}$ in this section for simplicity. $impact\,(i, r_k, n_{j,k})$ is calculated as a weighted average of marginal travel distance and other surrogate cost measures. In particular, we have

$$impact\,(i, r_k, n_{j,k}) = \alpha_{SI} SI\,(i, r_k, n_{j,k}) + \alpha_{EI} EI(i) + \alpha_{II} II\,(i, r_k, n_{j,k}) \tag{9}$$

where SI, EI, and II represent self-impact, external-impact, and internal-impact respectively. α_{SI}, α_{EI}, and α_{II} are external parameters representing the weight of each impact component. $\alpha_{SI} + \alpha_{EI} + \alpha_{II} = 1$. Definition of each impact component is presented below.

- $SI\,(i, r_k, n_{j,k})$ measures the coverage of the service time window of customer i. Let $\alpha_i\,(n_{j,k})$

denote the arrival time at customer i if the customer is scheduled immediately prior to customer $n_{j,k}$ on route k. The self-impact is calculated as the difference between the vehicle arrival time at customer i and the beginning of the time window of customer i.

$$SI(i, r_k, n_{j,k}) = \begin{cases} \max\left(e_i, a_{n_{j-1,k}} + s_{n_{j-1,k}} + t_{n_{j-1,k},i}\right) - e_i & \forall j > 1 \\ \max\left(e_i, a_{0,k,t} + t_{n_{0,k,t},i}\right) - e_i & j = 1 \end{cases} \tag{10}$$

- $EI(i)$ measures the inconvenience caused for the remaining un-routed customers as a result of routing customer i. Whenever we schedule a customer in the partial solution, it becomes more difficult to schedule other customers due to potential infeasibility in the time window and/or vehicle capacity. The external-impact quantifies this difficulty. Let UC denote the set of un-routed customers in the current iteration.

$$EI(i) = \frac{1}{|UC| - 1} \sum_{j \in UC \setminus \{i\}} \max\left(l_j - e_i - t_{i,j}, l_i - e_j, t_{i,j}\right) \tag{11}$$

- $II\left(i, r_k, n_{j,k}\right)$ denotes internal-impact, which is calculated as a weighted average of 3 sub-components. The first sub-component $c_1\left(i, r_k, n_{j,k}\right)$ measures the marginal travel distance of inserting customer i. The second sub-component $c_2\left(i, r_k, n_{j,k}\right)$ measures the maximum amount of delay in arrival time at subsequent customers. The third measure $c_3\left(i, r_k, n_{j,k}\right)$ is calculated as the time gap between the earliest possible arrival time at customer i and the beginning of the time window at customer i. This measure expresses the compatibility of customer i with the insertion position. β_1, β_2 and β_3 are external parameters representing the weight of each sub-component. $\beta_1 + \beta_2 + \beta_3 = 1$.

$$c_1\left(i, r_k, n_{j,k}\right) = t_{n_{j-1,k},i} + t_{i,n_{j,k}} - t_{n_{j-1,k},n_{j,k}} \tag{12}$$

$$c_2\left(i, r_k, n_{j,k}\right) = \max\left(e_{n_{j,k}}, a_i + s_i + t_{i,n_{j,k}}\right) + \max\left(e_{n_{j,k}}, a_{n_{j-1,k}} + s_{n_{j-1,k}} + t_{n_{j-1,k},n_{j,k}}\right) \tag{13}$$

$$c_3\left(i, r_k, n_{j,k}\right) = l_i - \left(a_{n_{j-1,k}} + s_{n_{j-1,k}} + t_{n_{j-1,k},i}\right) \tag{14}$$

$$II\left(i, r_k, n_{j,k}\right) = \beta_1 c_1\left(i, r_k, n_{j,k}\right) + \beta_2 c_2\left(i, r_k, n_{j,k}\right) + \beta_3 c_3\left(i, r_k, n_{j,k}\right) \tag{15}$$

3.3.2 Local Search

A Simulated Annealing (SA) algorithm embedded with well-known local search operators is developed to improve the initial partial solution at each decision epoch. These operators are widely used to solve a variety of vehicle routing problems [18, 1, 21, 26]. In particular, one of the following 7 local search operators is randomly selected at each iteration of the Simulated Annealing algorithm. The first group of methods operates on inter-route neighborhoods, meaning that two vehicle routes are changed simultaneously. The second set of methods operates on intra-route neighborhoods, meaning that only one vehicle route is changed.

Inter-route operators:

Crossover. One customer is randomly selected from each of the two routes. The sequence of customers following and including the chosen customers are switched between the two original routes to form two new routes.

Relocate. One customer is randomly selected and removed from the first route, and then inserted to a random position in the second route.

Relocate2. Similar to Relocate, except that a pair of two consecutive customers is relocated from one route to another.

Intra-route operators:

Reinsert. One customer is randomly selected and removed from one route. The customer is then inserted to a random position on the same route that it is removed from.

Or-opt2. Similar to Reinsert, except that a pair of two consecutive customers is reinserted to a random position on the same route that it is removed from.

Or-opt3. Similar to Reinsert, except that a sequence of three consecutive customers is reinserted to a random position on the same route that it is removed from.

2opt-exchange. Two customers are randomly selected from one route. The sequence of customers between and including the two chosen customers is reversed.

In each iteration, let S_t and S'_t denote the original and updated partial solution respectively. Let $cost\,(S_t)$ and $cost\,(S'_t)$ denote the corresponding total travel distance. If $cost\,(S'_t) < cost\,(S_t)$, S'_t is accepted as the current partial solution. Otherwise, S'_t is only accepted up to a probability determined by the acceptance rule of the simulated annealing algorithm. An exponential acceptance

probability function is implemented in our model, which specifies

$$P(S_t, S'_t, T) = p^* * \exp\left(-\frac{cost\,(S'_t) - cost\,(S_t)}{T}\right) \tag{16}$$

where $P(S_t, S'_t, T)$ represents the probability of accepting the updated partial solution S'_t given the current solution S_t and temperature T. T is an external variable which monotonically decreases as the number of iterations increase. p^* is an external parameter representing the maximum probability of acceptance. $P(S_t, S'_t, T) = p^*$ if and only if $cost\,(S'_t) = cost\,(S_t)$.

3.3.3 Waiting Time Adjustment

In a dynamic vehicle routing environment, the design of arrival and departure times directly affects the final total cost of a solution. Studies in the recent literature have shown that waiting can be a useful strategy in handling dynamic customer arrivals, especially those with time windows [19]. In this section, we first review two common waiting time adjustment strategies for the vehicle routing problem with time windows, namely the Push Backward and Push Forward heuristics. We then illustrate the hybrid waiting time adjustment heuristic we develop for the look-ahead partial routing framework.

Push Backward. All arrival and departure times are set based on the wait-first strategy. That is, when a vehicle finishes service at its current customer and becomes idle, it should first wait at its current location, and then travel to the next customer at the earliest time to ensure no waiting time at the next customer before it could start service. Let i^- and i^+ denote the predecessor and successor of customer i respectively. The Push Backward heuristic can be represented as follows.

$$a_i = \max\left(e_i, a_{i^-} + s_{i^-} + t_{i^-,i}\right)$$
$$b_i = a_{i^+} - t_{i,i^+}$$

Figure 3.3 illustrates an example of the Push Backward heuristic. Let g_i represent the time when service is finished at customer i. There are three customers. The beginning and ending of their service time windows are labeled by a set of single and double bars respectively. Solid arcs represent travel times, dashed arcs represent service times, and double-headed arrows show waiting times. Suppose that the vehicle arrives at customer 1 exactly at time e_1 and begins service right away. Service is finished at time g_1. The vehicle waits at customer 1 until time b_1 before traveling

to customer 2, so that it does not arrive earlier than the service time window at customer 2. The subsequent arrival and departure times are calculated accordingly.

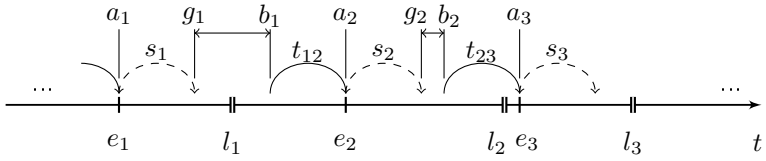

Figure 3.3: Example I: Push backward

Push Forward. In a process quite like the reverse of the Push Backward heuristic, all arrival and departure times are pushed towards the end of the planning horizon as much as possible. Equivalently speaking, when a vehicle finishes service at its current customer and becomes idle, it should first wait at its current location, and then travel to the next customer at the latest time possible, while maintaining time window feasibility at all subsequent customers. The heuristic can be represented as follows.

$$a_i = \min\left(l_i, a_{i^+} - t_{i,i^+} - s_i\right)$$

$$b_i = a_{i^+} - t_{i,i^+}$$

Figure 3.4 illustrates an example of the Push Forward heuristic. The set of customers and their service time windows are the same as in Example I.

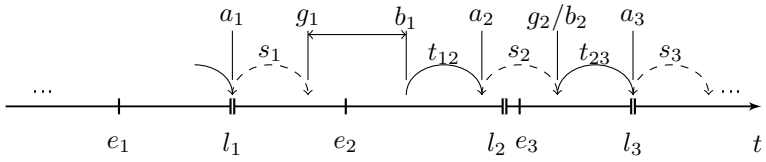

Figure 3.4: Example II: Push forward

Hybrid Waiting Time Adjustment. When solving the dynamic vehicle routing problem, we aim to minimize the total travel distance of all vehicles. Waiting is assumed to be cost-free. Thus it is more efficient to let an idle vehicle wait at its current location for more information on future customer requests to become available before letting it to travel to the next customer. This can be achieved by applying the Push Backward heuristic to all of the customers scheduled on each vehicle.

Additionally, we want to maximize the chance of accommodating dynamic (and possibly anticipated) customers when they request service. Recall that at each decision epoch, a new set of anticipated customers is selected and routed in the partial schedule. An anticipated customer remains in the solution until one of the following 3 cases happen. Case I: the customer requests service, at which time it becomes confirmed. Case II: it is time for the assigned vehicle to travel to the customer, at which time it has to be dropped form the schedule and becomes unconfirmed. If the same customer requests service after its reserved time slot has expired, it is routed using the cheapest insertion heuristic discussed. Case III: the next decision epoch is reached, at which time the customer becomes unconfirmed and a new set of anticipated customers is selected. Since time slots reserved for anticipated customers in the partial schedule have been optimized by the local search heuristic, case I is clearly a more desirable outcome than case II. Thus, when a vehicle has finished serviced at its current customer and the next customer scheduled on the route is an anticipated customer who has yet to request service, it is beneficial to let the vehicle wait for the customer to realize at its current location, for as long as possible before having to drop the customer out of the schedule. Equivalently, we want to place the maximum amount of slack time possible just prior to the anticipated customer on each route. This can be achieved by applying the Push Forward heuristic to all customers scheduled between (and including) the first anticipated customer on the route and the end of the route.

Figure 3.5 illustrates an example of the hybrid heuristic. The set of customers and their service time windows are the same as in Examples I and II. Suppose that customer 2 is an anticipated customer who has yet to request service, while customers 1 and 3 are confirmed customers. The hybrid heuristic locates the largest possible amount of waiting time between customer 1 and 2 as shown by the gap between g_1 and b_1.

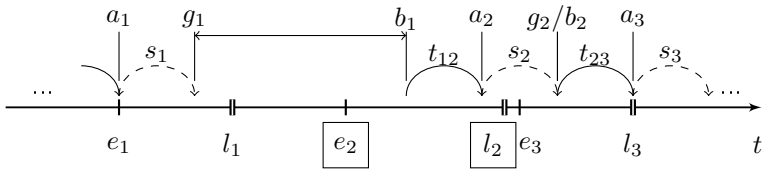

Figure 3.5: Example III: Hybrid waiting time adjustment

4 Experimental Results

4.1 Experiment Setup

Simulations are performed on a modified Solomon RC201 instance for the vehicle routing problem with time windows (VRPTW) [28]. The instance specifies all of the deterministic information on customer locations, demands, service time windows, and fleet capacity. There are 100 customers and the length of the planning horizon is 960 time steps. A dynamic vehicle routing problem can be constructed by specifying two parameters, namely the percentage of advance customers - $ACPercent$, and the probability that a dynamic customer requests service - $RequestProb$. These two parameters jointly determine the mixture between the number of advance customers and the expected number of realized dynamic customers in the problem. We assume that all dynamic customers have the same probability of requesting service. $q_i = RequestProb, \forall i$. We use a triangular distribution function to model $f_i(t)$, the conditional probability density function of request time u_i.

A dynamic vehicle routing instance constructed as above contains all deterministic and stochastic information of the problem. A realization of the problem specifies the actual set of advance customers, a group of dynamic customers who are to make requests, and the precise request times of these customers. A realization reflects the operation environment faced by decision makers. For each dynamic instance, we simulate 50 realizations and report the average results.

4.2 Routing Strategies and Parameters

For each realization of the dynamic vehicle routing problem, we simulate the following three routing strategies and record relevant performance measures for each strategy.

Static routing. A static vehicle routing problem with time windows is built and solved. The set of customers is the union of advance customers and dynamics customers who request service. It is assumed that all of these customers are known at the beginning of the planning horizon and must be served. We assume that the number of available vehicles is unlimited and solve this deterministic VRP by using the construction and the local search heuristics. We record the total distance $static\_Dist$ and number of vehicles used $static\_NumV$. This strategy assumes perfect problem information and thus provides a lower bound on the total travel distance if solved optimally.

Dynamic partial routing. The proposed look-ahead dynamic partial routing framework is used to solve the dynamic vehicle routing problem. The number of vehicles is set to be the same as

in the static solution. In the case that dynamic customers have to be rejected due to insufficient fleet capacity, extra vehicles are added. Thus the final solution accommodates all advance customers together with the dynamic customers who request service. This solution is called the dynamic partial routing solution. The corresponding total travel distance is denoted as $dynamic\_Dist$. Multiple model parameters can be changed to adjust the level of planning and partial routing in the proposed framework, including the number of decision epochs - $numEpoch$, the length of the forecast horizon - $forecastHorizon$, and the threshold value used to select the anticipated customers - $threshold$. In addition, each heuristic algorithm has its own adjustable parameters. Through preliminary experiments, we have identified and fixed the values of some parameters for better performance. In particular, we set $\alpha_{SI} = 0.33$, $\alpha_{EI} = 0.33$, $\alpha_{II} = 0.34$, $\beta_1 = 0.8$, $\beta_2 = 0.1$ and $\beta_3 = 0.1$.

Reactive routing. A reactive heuristic is used to solve the dynamic vehicle routing problem. This heuristic makes no forecast and no planning on dynamic customers. At the beginning of the planning horizon, the construction and local search heuristics are used to construct vehicle routes using only the known demand. No re-optimization is performed. Dynamic customer requests are handled in a reactive fashion by the cheapest insertion heuristic. Similarly as above, the number of vehicles is set to be the same as in the static case and additional vehicles are added when necessary. The final solution is called the reactive routing solution. The corresponding total travel distance is denoted as $reactive\_Dist$.

4.3 Dynamic Partial Routing in Base Case

As discussed in the introduction, reactive heuristics for solving the dynamic vehicle routing problem work reasonably well in fairly dynamic environments. The reactive routing strategy defined above falls into this category. In contrast, the dynamic partial routing framework proactively plans for dynamic customers and conducts partial routing of the vehicles. In this section, we study the performance of the partial routing strategy under various model parameter settings and benchmark its performance against the reactive routing strategy. We use the total travel distance as the measure of comparison and calculate the percentage savings in total travel distance of the dynamic partial routing as compared to the reactive routing. $Distance\ Saving = \frac{reactive\_Dist - dynamic\_Dist}{reactive\_Dist}$.

In the base case, we use $ACPercent = 0.25$ and $RequestProb = 0.75$ to construct a fairly dynamic operating environment. The number of advance customers is $100 * 0.25 = 25$ and the

expected number of dynamic customers is $100 * (1 - 0.25) * 0.75 \approx 57$. The values of the model parameters we test are $numEpochs = 1, 2, 5, 10$ and 20, $forecastHorizon = 24, 48, 96, 192, 480$ and 960, and $\frac{threshold}{RequestProb} = 0.2, 0.5$ and 0.8. With $T_{max} = 960$, each one of the $forecastHorizon$ values selected (except 24) corresponds to the time between two consecutive decision epochs with respect to one of the $numEpochs$ values. For example, when $T_{max} = 960$, $numEpochs = 10$, and $forecastHorizon = 48$, the time between decision epochs is 96 and at each decision epoch, the request forecasting procedure forecasts for 48 time units. The values of $threshold$ are relative to the $requestProb$ value of the instance. For example, the actual threshold values used in the base case simulations are $0.75 * 0.2 = 0.15$, $0.75 * 0.5 = 0.375$, and $0.75 * 0.8 = 0.6$, representing low, medium, and high levels of acceptance respectively.

Table 4.1 shows the distance savings of the dynamic partial routing in the base case. The $numEpochs$ is fixed at 5 and the table shows the sensitivity of the model with respect to different settings of the parameters $forecastHorizon$ and $threshold$.

$numEpochs = 5$						
$threshold$	$forecastHorizon$					
	24	48	96	192	480	960
0.15	18.61%	24.50%	28.26%	28.42%	28.09%	28.51%
0.375	16.25%	18.75%	25.21%	28.07%	28.23%	28.41%
0.6	14.37%	16.50%	21.50%	27.95%	28.19%	28.56%

Table 4.1: Distance savings of dynamic partial routing in base case

1. When holding the $threshold$ fixed, the distance savings increase as the length of the forecast horizon increases. The longer the forecast horizon, the higher the probability that a dynamic customer requests service within this time frame. Given the same $threshold$ value, more dynamic customers are anticipated and routed at each decision epoch. This in turn increases the chance that when a dynamic customer requests service, it would receive a reserved time slot from the current solution. Such time slots have been optimized by the local search and waiting time adjustment heuristics and hence tend to be more efficient than the ones generated by the myopic cheapest insertion procedure.

2. The increasing trend reaches its peak (or plateaus in some cases) at $forecastHorizon = 96$ with $threshold = 0.15$ and at $forecastHorizon = 192$ with $threshold = 0.375$ and 0.6. Given that $T_{max} = 960$ and $numEpochs = 5$, the time between two consecutive decision epochs

equals to $960/5 = 192$ time steps. We call 192 the breakpoint corresponding to $numEpochs = 5$. With $forecastHorizon = 192$, it is guaranteed that the entire planning horizon will be covered by the request forecast procedure. With $forecastHorizon = 96$ and $threshold = 0.15$, even though the planning horizon is not entirely covered, the low threshold value makes it more likely for dynamic customers to be anticipated. As a result, the distance saving of this setting is nearly as high as with $forecastHorizon = 192$. In general, when setting the $forecastHorizon$ value to be equal to the breakpoints, all of the stochastic information on possible future customer requests can be exploited. Further increases in the length of the forecast horizon would result in overlapping and not provide additional information about future requests, thus not leading to extra distance savings.

3. When holding the $forecastHorizon$ fixed, the distance savings decrease as the $threshold$ value increases. When the $threshold$ is higher, fewer dynamic customers would be anticipated and routed at each decision epoch. Consequently, less of the dynamic customers, when they request service, would be provided with pre-planned time slots. More of the dynamic customers would have to rely on the cheapest insertion procedure which is myopic and generally more costly.

We have established that for $numEpochs = 5$ and fixed $forecastHorizon$ value, the smallest threshold value at $threshold = 0.15$ will generate the highest distance savings. The same trend holds for all other $numEpochs$ values we have tested. Our next task is to analyze the sensitivity of the dynamic partial routing strategy with respect to the parameter $numEpochs$.

Table 4.2 shows the distance savings of the dynamic partial routing strategy with different parameter settings of $numEpochs$ and $forecastHorizon$. The value of $threshold$ is fixed to be 0.15 for all experiments. Again the savings are measured with respect to the reactive routing strategy.

$threshold = 0.15$						
$numEpochs$	$forecastHorizon$					
	24	48	96	192	480	960
1	5.24%	8.44%	13.35%	20.65%	27.93%	27.90%
2	9.19%	13.05%	18.77%	26.28%	28.30%	27.98%
5	18.61%	24.50%	28.26%	28.42%	28.09%	28.51%
10	25.22%	28.56%	28.99%	29.07%	29.22%	28.90%
20	27.40%	28.66%	29.20%	28.74%	29.22%	29.01%

Table 4.2: Sensitivity analysis of parameters $numEpochs$ and $forecastHorizon$

4. As an extension of observations 1 and 2, when holding $numEpochs$ fixed, the distance savings increase as the length of the forecast horizon increases. The increasing trend reaches

its peak (or plateaus in some cases) no later than when the value of $forecastHorizon$ reaches the breakpoints corresponding to the value of $numEpochs$. With $T_{max} = 960$, the breakpoints corresponding to $numEpochs = 1, 2, 5, 10$ and 20 are $960, 480, 192, 96$ and 48 respectively. Indeed the tables shows that with $numEpochs = 1$ or 2, the increasing trend peaks at $forecastHorizon = 480$. With $numEpochs = 5$, the cost saving plateaus at $forecastHorizon = 96$. Last but not least, with $numEpochs = 10$ or 20, the cost saving plateaus at $forecastHorizon = 48$. This observation confirms the claim that distance savings are generated by dynamic customers who receive optimized time slots in the existing schedule (observation 1). It also supports the claim that there exist a minimum level of $forecastHorizon$ that is sufficient to generate the maximum distance savings. This minimum level aligns with the breakpoint value corresponding to each $numEpochs$ value.

5. When holding $forecastHorizon$ fixed, distance savings increase as $numEpochs$ increases. Having more decision epochs corresponds to a higher level of partial routing since the demand forecast and re-optimization procedures are performed more frequently. This allows newly revealed information to be handled more promptly, leading to extra cost savings. This phenomenon is much more significant at lower settings of $forecastHorizon$ than at higher settings. When $forecastHorizon$ is small, the entire planning horizon is hardly covered by the forecast procedure even with the largest $numEpochs$ value we test. ($24 * 20 \ll 960$). Consequently, each increase in the number of epochs will significantly increase the proportion of the planning horizon that is covered, leading to a sizable increase in cost savings. In contrast, when $forecastHorizon$ is large, the majority of the planning horizon can be covered even with the least number of epochs. For example, at $forecastHorizon = 960$, only one epoch is needed at $t = 0$ to cover the entire planning horizon. In this case, increasing the number of decision epochs has only limited effects on the total cost savings, as reflected in the right-most column of Table 4.2.

We have now studied how the model parameters $numEpoch$, $forecastHorizon$, and $threshold$ individually and jointly affect the performance of the look-ahead partial routing algorithm. We have concluded that it is beneficial to set the threshold value based on $\frac{threshold}{RequestProb} = 0.2$, and set the number of decision epochs at $numEpoch = 20$. Besides, it is sufficient to set the length of the forecast horizon to be equal to the breakpoint value corresponding to the number of epochs. In this case, $forecastHorizon = 48$ is sufficient. These settings are used throughout the next section.

4.4 Sensitivity to the Expected Proportion of Realized Dynamic Customers

As discussed in the previous sections, the dynamic partial routing strategy lies in the middle of the spectrum that reflects the amount of problem information used in a solution approach. On one end of the spectrum is the static routing strategy that assumes a deterministic system and uses perfect problem information; on the other end lies the totally reactive routing strategy that does not make use of any stochastic information. In this section, we seek to explore the behavior of dynamic partial routing in problems with different expected proportions of realized dynamic customers and compare it to the benchmarking strategies lying at the ends of the spectrum. In our experiments, we use the parameters $ACPercent$ and $RequestProb$ to adjust the mixture between advance and realized dynamic customers. The expected proportion of realized dynamic customers among all realized customers can be calculated as $\frac{RequestProb*(1-ACPercent)}{RequestProb*(1-ACPercent)+ACPercent}$, which lies between 0 and 1 by definition. Intuitively, the higher the values of $ACPercent$, the smaller the expected proportion of realized dynamic customers and the less dynamic the problem is. Similarly, the higher the values of $RequestProb$, the higher the expected proportion of realized dynamic customers is. In our experiments, the expected proportion of realized dynamic customers ranges between 0.01 and 0.89.

Since the static routing strategy assumes a deterministic environment, it provides a lower bound on total travel distance for the other two strategies that solve the dynamic routing problem. Thus we report their distance penalties as compared to the static solution. For example, the distance penalty of dynamic partial routing is calculated as $Distance\ Penalty = \frac{dynamic\_Dist - static\_Dist}{static\_Dist}$. The smaller the penalty, the better the solution. The results are summarized in Table 4.3.

	$RequestProb$									
	0.1	0.25	0.5	0.75	0.9	0.1	0.25	0.5	0.75	0.9
$ACPercent$	Reactive Routing					Dynamic Partial Routing				
0.1	24.4%	33.2%	56.5%	57.3%	62.4%	14.3%	11.3%	6.9%	5.4%	4.4%
0.25	12.6%	17.6%	35.7%	47.1%	53.1%	11.2%	10.6%	6.6%	4.6%	3.1%
0.5	5.5%	4.8%	18.7%	29.5%	33.2%	6.8%	6.0%	4.5%	2.6%	2.3%
0.75	0.6%	4.2%	9.2%	13.4%	17.9%	3.7%	3.8%	3.0%	1.7%	1.5%
0.9	0.5%	4.0%	6.0%	5.5%	5.9%	2.0%	1.4%	1.4%	1.5%	1.9%

Table 4.3: Sensitivity to different expected proportions of realized dynamic customers

6. When holding the $ACPercent$ fixed at low to moderate levels (0.1 through 0.5), the distance

penalties increase for the reactive routing strategy as $RequestProb$ increases. This strategy does not make use of any stochastic information on dynamic customers, and thus it suffers more when there are more dynamic customers in the system. The trend becomes less significant when the $ACPercent$ is high. At $ACPercent = 0.9$, the reactive routing strategy shows relatively flat performance except for the case where $RequestProb = 0.1$. In this case, the expected proportion of realized dynamic customers is only $\frac{RequestProb*(1-ACPercent)}{RequestProb*(1-ACPercent)+ACPercent} = \frac{0.1*(1-0.9)}{0.1*(1-0.9)+0.9} \approx 0.01$ and the routing environment is practically static. The reactive and static routing strategies are expected to behave similarly as reflected by the 0.5% average distance penalty.

7. On the contrary, distance penalties decrease for the dynamic partial routing strategy as the $RequestProb$ increases when holding the $ACPercent$ fixed at 0.1 through 0.5. The partial routing strategy uses a threshold rule to forecast customer requests. When the $RequestProb$ is low, many time slots reserved for anticipated customers will not get confirmed and are wasted. The quality of the forecast is poor. These time slots take up both physical and temporal capacity of the fleet. As the $RequestProb$ increases, more of the time slots reserved for anticipated customers will get confirmed. The quality of the forecast increases, and the distance penalty becomes smaller. Similarly as for the reactive routing strategy, the penalty measures become flat across different values of $RequestProb$ for problems with higher $ACPercent$ values. Especially at $ACPercent = 0.9$, the number of dynamic customers is only 10 and is too small as compared to the number of advance customers, such that the exact number of realized dynamic customers barely affect the quality of the solution.

8. When holding $RequestProb$ fixed, both the reactive routing and dynamic partial routing strategies perform better for problems with more advance customers and lower expected proportions of realized dynamic customers. Even though the two strategies use different methods to handle randomness in the problem, they both suffer from the uncertainties in dynamic requests and both have to make routing decisions based on partial information.

9. We now compare the reactive routing and the dynamic partial routing across problems with different expected proportions of realized dynamic customers. Generally speaking, dynamic partial routing outperforms reactive routing for problems with low $ACPercent$ and high $RequestProb$ values. Among all the problems we have tested, the highest expected proportion

of realized dynamic customers is 0.89, corresponding to the instance with $ACPercent = 0.1$ and $RequestProb = 0.9$. The partial routing strategy shows only 4.4% in distance penalty and outperforms reactive routing by the largest margin among all instances. On the contrary, for problems with high $ACPercent$ and low $RequestProb$ values, reactive routing outperforms dynamic partial routing. This suggests that excessive planning for future requests could backfire in situations where the number of dynamic customers is small. Nevertheless, it is evident that the contrast between 0.5% and 2.0% in distance penalties in the case with the lowest expected proportion of realized dynamic customers is minimal as compared to the sharp difference between 62.4% and 4.4% in the case with the highest expected proportion of realized dynamic customers. It suggests that even though the quality of forecast is poor when there are few dynamic customers, the re-optimization scheme in the partial routing framework could promptly correct the errors in forecasting based on newly revealed information. This implies that dynamic partial routing is more flexible and robust than reactive routing across problems with different expected proportions of realized dynamic customers.

4.5 Unit Cost Analysis of the Different Routing Strategies

So far we have focused on benchmarking the total travel distance of serving all customers. Another metric of interest is the average travel distance per customer, which can be calculated by dividing the total travel distance by the total number of serviced customers (both advance and dynamic customers). This unit cost measure allows us to study the effect of economies of scale of serving additional customers for problems of different sizes. In the context of the vehicle routing problem with dynamic customers, the size of a problem can be interpreted in two dimensions. The first dimension relates to the total number of realized customers, and the second dimension concerns with the expected proportion of realized dynamic customers given the total number of customers. To this end, we now analyze the average travel distance per customer on a set of instances that reflect both dimensions of the size of the problem.

Table 4.4 reports the average travel distance per customer. Each row of the table contains the results corresponding to problems with the same expected number of customers, and the number is listed in the first column of the table. The rest of the columns contain the results of the corresponding routing strategy. In static routing, all of the customers are assumed to be advance customers. The mixture between advance and dynamic customers is irrelevant. For the other two strategies,

		Expected Proportion of Realized Dynamic Customers			
		High	Low	High	Low
Expected Number of Customers	Static Routing	Dynamic Partial Routing		Reactive Routing	
32.50	20.46	21.98	22.76	26.28	23.07
55.00	16.16	18.29	17.26	26.75	17.05
77.50	13.84	14.89	14.36	22.21	13.93
91.00	12.94	13.62	13.20	21.19	13.04

Table 4.4: Average travel distance per customer for different problem sizes

we report the results on two DVRP instances, one with a high expected proportion of realized dynamic customers (ranging between 0.69 and 0.89) and the other with a low expected proportion of realized dynamic customers (ranging between 0.01 and 0.23). Note that both instances have the same total expected number of customers.

10. For the static routing strategy, as the expected number of customers increases, more customers can be accommodated on the same vehicle based on proximity in their locations. Thus the proportion of dead heading miles is reduced and the average travel distance per customer decreases due to economies of scale on vehicle usage.

11. For the dynamic partial routing strategy, the average travel distance per customer is roughly the same in cases with high or low expected proportions of realized dynamic customers, given the same total expected number of customers. This suggests that the partial routing strategy is robust with respect to the mixture between advance versus dynamic customers. The re-optimization algorithm shows its advantage in quickly adapting to new information and easing out part of the inconvenience caused by uncertainties in dynamic requests. The average travel distance per customer in both the high and low expected proportions of realized dynamic customers cases are marginally higher than the corresponding measure of the static routing solution, which is consistent with the previous analysis. As the expected number of customers increases, the average travel distance per customer decreases similarly as in the static routing case due to the effect of economies of scale. This suggests that under the dynamic partial routing strategy, the system has the ability to accommodate additional dynamic customers without the risk of increasing the average travel distance per customer.

12. For the reactive routing strategy, the average travel distance per customer is significantly larger in cases with high expected proportions of realized dynamic customers than in cases

where the proportions are low, given the same total expected number of customers. This suggests that reactive routing is very sensitive to the number of dynamic customers in the problem because these customers are handled by the myopic cheapest insertion heuristic, which is globally suboptimal. This observation is consistent with the previous analysis which shows that the reactive routing strategy performs particularly poorly on instances where the value of *RequestProb* is high. In the cases with high expected proportions of realized dynamic customers, the average travel distance per customer are significantly higher than the corresponding measure of the static routing solution, suggesting that the benefit of economies of scale is out-weighted by the increase in total travel distance caused by the suboptimal routing strategy. Additional dynamic customers tend to cause the average travel distance per customer to increase dramatically.

5 Conclusion

In this article, we study the vehicle routing problem with dynamic customer requests. We model the uncertainties related to dynamic customer requests by assuming an underlying probability of request as well as a conditional likelihood function on the request time. We adopt a look-ahead dynamic routing approach to design a solution framework that proactively forecasts future customer requests. The hybrid waiting time adjustment heuristic strategically optimizes time slots in the current schedule in anticipation for potential requests. Dynamic real-time routing rules are developed to minimize the total travel distance of all vehicles as well as maximize the probability of accepting dynamic customer requests. The level of forecasting and route planning in our solution can be adjusted by changing the values of three model parameters, namely the *numEpochs*, $forecastHorizon$, and *threshold*. This partial routing capability positions our solution in the middle of the spectrum that reflects the amount of problem information used in a solution approach.

Through extensive numerical simulations, we first study the behavior of the proposed partial routing framework under different parameter settings. Using the base case network, where 25 customers are known in advance and each dynamic customer has a 75% chance of requesting service, we identify that the lowest *threshold* value generally leads to the best result. We also show that there exists a minimum value of the $forecastHorizon$ that is sufficient to exploit the benefit of forecasting dynamic requests. This value aligns with the time between two consecutive decision epochs given the choice of the *numEpochs*. We then compare and contrast the above mentioned

routing strategies in networks with various levels of uncertainties. The dynamic partial routing strategy is shown to be more reliable than reactive routing across problems with different expected proportions of realized dynamic customers. The analysis based on the average travel distance per customer shows that the dynamic partial routing strategy could benefit from the effect of economies of scale on vehicle usage in situations with both high and low levels of expected proportions of realized dynamic customers.

Acknowledgement

We acknowledge Metrans for its kind support of this research.

References

[1] Russell W. Bent and Pascal Van Hentenryck. Waiting and relocation strategies in online stochastic vehicle routing. In *International Joint Conferences on Artificial Intelligence*, pages 1816–1821. 2007 edition, 2007.

[2] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, 24(3):343–355, 2012.

[3] Yossi Borenstein, Nazaraf Shah, Edward Tsang, Raphael Dorne, Abdullah Alsheddy, and Christos Voudouris. On the partitioning of dynamic workforce scheduling problems. *Journal of Scheduling*, 13(4):411–425, 2010.

[4] Zhi-Long Chen and Hang Xu. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88, 2006.

[5] George B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[6] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[7] Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013.

[8] Michel Gendreau, François Guertin, Jean-Yves Potvin, and René Séguin. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C: Emerging Technologies*, 14(3):157–174, 2006.

[9] Gianpaolo Ghiani, Emanuele Manni, and Barrett W. Thomas. A comparison of anticipatory algorithms for the dynamic and stochastic Traveling Salesman Problem. *Transportation Science*, 46(3):374–387, 2012.

[10] George Ioannou, M. Kritikos, and G. Prastacos. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *Journal of the Operational Research Society*, 52(5):523–537, 2001.

[11] Philip Kilby, Patrick Prosser, and Paul Shaw. Dynamic VRPs: A study of scenarios. Technical report, University of Strathclyde, Glasgow, Scotland, 1998.

[12] Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[13] Allan Larsen, Oli B. G. Madsen, and Marius M. Solomon. The a priori dynamic traveling salesman problem with time windows. *Transportation Science*, 38(4):459–472, 2004.

[14] Yannis Marinakis and Magdalene Marinaki. Combinatorial expanding neighborhood topology particle swarm optimization for the vehicle routing problem with stochastic demands. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 49–56, 2013.

[15] Roberto Montemanni, Luca Maria Gambardella, Andrea Emilio Rizzoli, and Alberto V. Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4):327–343, 2005.

[16] Benoit Montreuil, Russell D. Meller, and Eric Ballot. Physical internet foundations. In Theodor Borangiu, Andre Thomas, and Damien Trentesaux, editors, *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, volume 472 of *Studies in Computational Intelligence*, pages 151–166. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[17] William H. Moore. National transportation statistics. Technical report, U.S. Department of Transportation, 2013.

[18] Rodrigo Moretti Branchini, Vinícius Amaral Armentano, and Arne Løkketangen. Adaptive granular local search heuristic for a dynamic vehicle routing problem. *Computers & Operations Research*, 36(11):2955–2968, 2009.

[19] Victor Pillac, Michel Gendreau, Christelle Guéret, and Andrés L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[20] Harilaos N. Psaraftis. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem, 1980.

[21] Martin W. P. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *INFORMS Journal on Computing*, 4(2):146–154, 1992.

[22] Hamid R. Sayarshad and Joseph Y. J. Chow. The non-myopic dynamic dial-a-ride and pricing problem. 2014.

[23] Michael Schilde, Karl F. Doerner, and Richard F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12):1719–1730, 2011.

[24] Nicola Secomandi and François Margot. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57(1):214–230, 2009.

[25] Geetha Shanmugam. Meta heuristic algorithms for vehicle routing problem with stochastic demands. *Journal of Computer Science*, 7(4):533–542, 2011.

[26] Zhihong Shen, Maged Dessouky, and Fernando Ordóñez. A two-stage vehicle routing model for large-scale bioterrorism emergencies. *Networks*, 54(4):255–269, 2009.

[27] Nagesh Shukla, M.K. Tiwari, and Darek Ceglarek. Genetic-algorithms-based algorithm portfolio for inventory routing problem with stochastic demand. *International Journal of Production Research*, 51(1):118–137, 2013.

[28] Marius M. Solomon. Algorithms for the vehicle-routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[29] Junlong Zhang, William H. K. Lam, and Bi Yu Chen. A stochastic vehicle routing problem with travel time uncertainty: Trade-Off between cost and customer service. *Networks and Spatial Economics*, 2013.

[30] Tao Zhang, W.A. Chaovalitwongse, and Yuejie Zhang. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. *Computers & Operations Research*, 39(10):2277–2290, 2012.