Restocking-Based Rollout Policies for the Vehicle Routing Problem with Stochastic Demand and Duration Limits

Justin C. Goodson   Barrett W. Thomas   Jeffrey W. Ohlmann

October 24, 2014

Abstract

We develop restocking-based rollout policies to make real-time, dynamic routing decisions for the *vehicle routing problem with stochastic demand and duration limits*. Leveraging dominance results, we develop a computationally tractable method to estimate the value of an optimal restocking policy along a fixed route. Embedding our procedure in rollout algorithms, we show restocking-based rollout out performs a priori-based rollout, demonstrating the value of explicitly considering preemptive capacity replenishment in a rollout approach for dynamic routing. We also demonstrate the effectiveness of basic local search versus more sophisticated mechanisms for the heuristic component of the rollout procedure.

1 Introduction

Vehicle routing problems (VRPs) with stochastic demand arise in a variety of logistics and supply chain management problems, such as less-than-truckload trucking (Bertsimas and Simchi-Levi, 1996) and vendor-managed distribution systems (Erera et al., 2009). In these and related problems, initial estimates of customer demand available at the time vehicle routes are planned often differ from the actual demands observed upon arrival to customer locations. These uncertainties in customer demands sometimes lead to costly *route failures*, situations where vehicle capacity is inadequate to fully serve demand, requiring the vehicle to return to a central depot to replenish capacity before continuing to service customers. The design of high-performance routing plans that mitigate the impact of route failures is a key issue in algorithm development for VRPs with stochastic demand.

Recent advances in communication technologies make it possible to move beyond static routing approaches (Campbell and Thomas, 2008) to dynamic solution methods that alleviate the effect of route failures by reoptimizing in response to observed customer demands. With operating margins between two and four percent in the trucking industry (American Trucking Association, 2009), even small improvements in productivity can be helpful in combating the flat rates and rising costs characteristic of today's trucking business (Wilson, 2011). In this paper, we focus on dynamic routing solutions for the *vehicle routing problem with stochastic demand and duration limits* (VRPSDL), where the objective is to maximize expected demand served subject to constraints on vehicle capacity and route duration.

To obtain solutions to the dynamic routing problem, we employ rollout procedures, a form of approximate dynamic programming in which decision rules are evaluated only for observed states by using heuristics to approximate the reward-to-go in the dynamic programming optimality equations (Bertsekas et al., 1997; Bertsekas, 2000; Goodson et al., 2013). A key challenge associated with rollout procedures is balancing the computational burden of the heuristic with the quality of the heuristic policy. An ideal heuristic facilitates real-time identification of high-quality dynamic routing policies.

Heuristic quality and computational requirements can be balanced in at least two ways. First, the search space can be restricted. In the context of Markov decision processes (MDPs), a restriction is equivalent to limiting the set of policies over which we search – we refer to this as restricting the policy class. Second, a heuristic's quality and computational requirement are tied to the method employed to find heuristic policies. For instance, when operating on a particular restricted policy class, one might be able to solve for an optimal policy within the class using a math programming technique. Alternatively, a greedy algorithm might accelerate the search, but may sacrifice policy quality.

In this paper, we address both the restricted policy class and the heuristic search method. To address the effect of the restricted policy class, we relax the policy class we use for the heuristic policies. Specifically, we relax the a priori fixed-route policy class of Goodson et al. (2013) by considering policies that permit preemptive capacity replenishment, a policy class referred to in the literature as *restocking* fixed-route policies. Notably, in contrast to the a priori-based heuristic of Goodson et al. (2013), the relaxation allows the heuristic to explicitly evaluate restocking actions. As a result, for a given fixed route, the value of the optimal restocking policy is at least as large as the value of the a priori policy along the same fixed route. The consequence is a better estimate of the expected demand served in a rollout policy and thus a potentially better rollout policy. However, it is computationally challenging to evaluate and search the space of restocking fixed-route policies. To address this challenge, we develop a computationally tractable dynamic programming procedure to identify an optimal restocking policy along a given fixed route and for a given sample of customer demands. We significantly reduce the computation required to solve the dynamic program by leveraging dominance results to prune the state-space graph. We apply the dynamic programming procedure across multiple samples to estimate the expected value of an optimal restocking policy along the fixed route.

To address the heuristic search method, in addition to the basic local search mechanism of Goodson et al. (2013), we consider a more sophisticated variable neighborhood search as the heuristic component of rollout algorithms. Surprisingly, despite the ability of the variable neighborhood search to identify better static fixed-route policies, the local search heuristic yields superior dynamic solutions. Our focus on the restricted policy class and heuristic search method establish our main contribution: the development of a real-time, dynamic solution approach for a multi-vehicle routing problem with stochastic demand that explicitly considers preemptive capacity replenishment.

The remainder of the paper proceeds as follows. We formally state the VRPSDL and provide a literature review in §2. In §3, we formulate the VRPSDL as a MDP. In §4, we describe our procedure to estimate the expected value of an optimal VRPSDL restocking policy along a given fixed route. We embed our estimation method in the heuristic lookahead mechanism for the rollout policies described in §5. As our computational results in §6 show, the restocking-based rollout policies for the VRPSDL yield better performance than the a priori-based rollout policies of Goodson et al. (2013). This performance improvement is the result of our approach's ability to effectively and efficiently estimate the rewards-to-go using restocking fixed-route policies. We conclude the paper in §7.

2 Problem Statement and Related Literature

We consider a fleet of capacitated vehicles serving a set of customers from a central depot. Before a vehicle arrives at a customer, customer demand is known only in distribution. Upon arriving at a customer, the realized demand is served to the fullest extent possible given the remaining capacity available on the vehicle. If the vehicle's capacity is consumed, the vehicle must return to the depot to replenish capacity. Otherwise, after serving demand or replenishing capacity, the vehicle's next destination is chosen. The vehicle can go to any customer who has not yet been visited or who has remaining demand, provided no vehicle is already en route or serving that customer. Further, the vehicle can travel to the depot to preemptively replenish capacity. This decision is made based on information known at the time of the decision and does not depend on any pre-planned tour. The objective of the problem is to maximize the expected demand served given a limit on the working day.

We use rollout algorithms to generate dynamic solutions to the VRPSDL. Rollout algorithms rely on iterative applications of heuristics to select actions at each decision epoch. Because in this paper our heuristic centers on restocking fixed-route policies, we focus much of our review on literature related to restocking policies for a given fixed route. This literature centers on obtaining restocking policies with an objective of minimizing some measure of travel cost. Although the methods we develop focus on maximizing demand served, there are methodological similarities between our work and the literature. For reviews of more general literature on stochastic routing, we refer the reader to Goodson et al. (2013) and Goodson et al. (2012).

Much of the literature on restocking policies is founded on the work of Bertsimas et al. (1995), which considers restocking policies for a single-vehicle routing problem with stochastic demand. For a given fixed route, Bertsimas et al. (1995) formulate a dynamic program to determine the expected length of the route that results from following an optimal restocking policy. Because Bertsimas et al. (1995) assume that customer demands are discrete, the dynamic program can be solved in polynomial time. We formulate a dynamic program to determine the demand served by following an optimal restocking policy for a given fixed route. However, because we model arrival times to customers as continuous (rather than discrete), we are unable to obtain a similar complexity result. In §4, we derive structural results and a forward dynamic programming procedure that overcome this difficulty. Our results apply to both discrete- and continuous-time models.

Yang et al. (2000) establish a key structural property for the dynamic program proposed by Bertsimas et al. (1995): an optimal restocking policy is to return to the depot if the available vehicle capacity is below a threshold or otherwise continue directly to the next customer. Because our objective is to maximize demand served, the threshold structure does not transfer to our work. Yang et al. (2000) also show that routing a single vehicle along a single fixed route is equivalent to using multiple vehicles unless additional constraints are imposed, such as the route duration constraints we consider. Although the structural result reduces the computation required to evaluate a fixed route as a restocking policy, Yang et al. (2000) find that searching for an optimal fixed route is still computationally prohibitive. To further reduce computation, Yang et al. (2000) develop a method to approximate the change in the expected length of a fixed route when it is modified by a local search procedure. Bianchi et al. (2006) embed this approximation strategy in metaheuristic procedures and demonstrate that the computational results of Yang et al. (2000) can be improved by considering more complex search methods. In our work, we employ a local search procedure to explore the space of restocking fixed-route policies, evaluating each policy via the methods we develop in §4.

Secomandi (2003) develops a rollout procedure to search for restocking policies for a single-vehicle routing problem with stochastic demand. Given an initial fixed route, the method is guaranteed to return a restocking fixed-route policy at least as good as the initial policy. In our work, we employ rollout to obtain solutions to the dynamic problem, using a restocking-based heuristic to estimate expected demand served. Drawing on the notion of a fortified rollout policy (Bertsekas et al., 1997; Goodson et al., 2013), we also guarantee weak improvement over an initial policy.

Tsirimpas et al. (2008) and Tatarakis and Minis (2009) consider three variations of the basic model proposed by Bertsimas et al. (1995) to evaluate the expected length of a given fixed route: the case of multi-product deliveries when each product is stored in its own compartment in the vehicle, the case of multi-product deliveries when all products are stored together in the vehicle's single compartment, and the case in which the vehicle picks up from and delivers a single product to each customer. The authors identify structural properties that aid in solving the dynamic programs. Demonstrating a threshold-type structure on the optimal policy for any integer-number of vehicle compartments, Pandelis et al. (2012) generalize the work of Tatarakis and Minis (2009), which applies to only two compartments. Again, because of differences in problem structure, the structural results do not translate to the problem discussed in this paper.

To make dynamic routing decisions for a single-vehicle routing problem with stochastic demand, the rollout procedures of Secomandi (2001) and Novoa and Storer (2009) employ restocking fixed-route policies along a pre-determined base sequence of customers. Novoa and Storer (2009) provide computational evidence that improving the base sequence can improve performance of the base policy. By construction, our approach also improves upon a base restocking policy. Further, we provide computational evidence of improvement resulting from consideration of restocking fixed-route policies versus a priori fixed-route policies, i.e., the benefit of explicitly considering preemptive capacity replenishment. While following an optimal restocking policy leads to performance at least as good as the performance of the a priori policy along the *same* fixed route, there is no guarantee the resulting rollout policy will yield improved solutions over a rollout policy based on a priori fixed-route policies. However, our computational results demonstrate improved reward-to-go estimates afforded by restocking fixed-route policies yield better rollout policy performance than the a priori fixed-route policy class of Goodson et al. (2013).

3 Problem Formulation

We formulate the problem of making dynamic routing decisions for the VRPSDL as a MDP. We present a summary of the MDP model in this section and refer the reader to the electronic companion of Goodson et al. (2013) for a detailed formulation. Let $G = (\mathcal{N}, \mathcal{E})$ be a complete graph where $\mathcal{N} = \{0, 1, \ldots, N\}$ is a set of $N + 1$ nodes and $\mathcal{E} = \{(n, n') : n, n' \in \mathcal{N}\}$ is the set of edges connecting the nodes. Node 0 represents a depot and nodes $1, \ldots, N$ represent customer locations. Travel times $t(n, n')$ associated with each edge (n, n') in $\mathcal{E}$ are known and assumed deterministic. Let $\mathcal{M} = \{1, \ldots, M\}$ be a set of M identical vehicles initially located at the depot. Let Q denote vehicle capacity and L a route duration limit (e.g., end of a working day) by which time all vehicles must return to the depot. Customer demands are random variables that follow a known joint probability distribution F with a support restricted to be a subset of $[0, \infty)^N$.

The state of the system is the tuple (l, t, q, d, x). For each vehicle, vectors l, t, and q store vehicle destinations, arrival times at vehicle destinations, and remaining vehicle capacities. The state of vehicle m in $\mathcal{M}$ is denoted (l_m, t_m, q_m). For each customer, vectors d and x store unserved customer demand and the observed demand, respectively, both of which are unknown for customers not yet visited. The state of demand at customer n in $\mathcal{N}$ is denoted (d_n, x_n).

An action is an assignment of the vehicles in $\mathcal{M}$ to locations in $\mathcal{N}$. We place the following restrictions on the available actions. Vehicles en route may not be diverted. If a vehicle's capacity will be depleted by serving customer demand at its current location, then the vehicle must return to the depot to replenish. We prohibit actions assigning more than one vehicle to a customer at a time, but allow multiple vehicles to simultaneously return to the depot. Over the horizon of the problem, a customer may be visited multiple times by different vehicles. Actions forcing a vehicle to return to the depot at a time greater than duration limit L are prohibited. Vehicles are not permitted to wait at locations, except at the depot when it is not feasible to collect demand from the remaining set of customers. We denote the pre-decision state at decision epoch k by s_k, an action in action space $\mathcal{A}(s_k)$ by a, and the post-decision state corresponding to action a by s_k^a (see Powell (2007) for a discussion of pre- and post-decision states).

The system dynamics proceed as follows. A decision epoch is triggered by the arrival of one or more vehicles at customer locations or at the depot (multiple vehicles may arrive at their respective locations simultaneously). Upon arrival to customer locations, actual demand is observed. In addition, vehicle capacities are replenished for vehicles arriving at the depot. These events are captured in the transition from post-decision state s^a_{k-1} to pre-decision state s_k. For vehicles at customer locations or at the depot (i.e., vehicles not en route), an action is selected indicating each respective location these vehicles will travel to next. Vehicles currently at customer locations then serve demand to the fullest extent given available vehicle capacity (total demand served in period k is recorded as reward $R_k(s_k, a)$) and the selected action, specifying the next destination for each vehicle, is executed. These events are captured in the deterministic transition from pre-decision state s_k to post-decision state s^a_k.

Let Π be the set of all Markovian deterministic policies for the VRPSDL. A policy π in Π is a sequence of decision rules: $\pi = (\delta^\pi_0, \delta^\pi_1, \dots, \delta^\pi_K)$, where each decision rule $\delta^\pi_k(s_k) : s_k \mapsto \mathcal{A}(s_k)$ is a function that specifies the action choice when the process occupies state s_k and follows policy π. We seek a policy π in Π that maximizes the total expected reward, conditional on initial state s_0: $\mathbb{E}[\sum_{k=0}^{K} R_k(s_k, \delta^\pi(s_k))|s_0]$. Denoting by $V(s_k)$ the expected reward-to-go from state s_k in epoch k through final decision epoch K, an optimal policy can be obtained by solving the optimality equation $V(s_k) = \max_{a \in \mathcal{A}(s_k)}\{R_k(s_k, a) + \mathbb{E}[V(s_{k+1})|s_k, a]\}$ for each epoch k and state s_k in state space $\mathcal{S}$, where $V(s_K) = 0$ for each absorbing state s_K.

4 A Restocking Fixed-Route Policy for the VRPSDL

Our solution method for the MDP of §3 relies on heuristic estimates of future rewards. We estimate the reward-to-go via restocking fixed-route policies. Because restocking fixed-route policies can be computationally challenging to evaluate, in this section we develop a method to estimate the value of an optimal restocking fixed-route policy along a given sequence of customers. We then employ these estimates in our rollout-based solution procedure.

In §4.1, we discuss the decision rule associated with a restocking policy. For a given state, the decision rule is a function that determines if a vehicle should travel directly to the next customer on the fixed route or first replenish capacity at the depot and then travel to the next customer. To evaluate the decision rule, we estimate the value of these two decisions via the expected value with perfect information, a computationally tractable upper bound on the value of an optimal restocking fixed-route policy along the given customer sequence. In §4.2, we introduce a deterministic dynamic program to facilitate estimation of the expected value with perfect information. We call this dynamic program the *auxiliary dynamic program* to distinguish it from the MDP model presented in the previous section. Given a sequence of customers and sampled customer demands, we solve a deterministic dynamic program to calculate the value of the optimal restocking policy for the given demands. As we demonstrate in §4.3, averaging the values of the auxiliary dynamic programs across multiple samples of customer demands is an unbiased estimate of the expected value with perfect information and facilitates estimation of the reward-to-go.

4.1 A Restocking Decision Rule for a Given State Along a Given Fixed Route

A fixed route specifies a static ordering of customers for a driver to visit. A vehicle is required to visit customers in the order specified by the fixed route. From a given customer on the fixed route, the decision rule determines whether the vehicle should travel directly to the next customer on the route or first replenish capacity at the depot.

We denote by v^m a fixed route for vehicle m in vehicle set $\mathcal{M}$, where $v^m = (v_1^m, v_2^m, \ldots, v_{B^m}^m)$ is a sequence of B^m locations such that v_1^m is l_m, the destination (or current location) of vehicle m in state s, and the remaining locations are customers in $\mathcal{N} \setminus \{0\}$. A fixed-route policy is characterized by $v = (v^m)_{m \in \mathcal{M}}$, a set containing a fixed route for each vehicle m in $\mathcal{M}$. In v, each customer with pending or unknown demand appears exactly once on exactly one route.

After a vehicle m serves demand at a customer v_b^m, vehicle m may continue directly to the next customer on its fixed route, v_{b+1}^m, or first replenish capacity at the depot before traveling to customer v_{b+1}^m. The choice between direct travel and capacity replenishment is decided by the action that leads to the largest estimate of expected demand served. If v_b^m is the final customer on the fixed route, or if continuing to v_{b+1}^m will result in a violation of the route duration limit, vehicle m is required to return to the depot. A return trip to the depot for the purpose of capacity replenishment is required in the event of a route failure.

To precisely describe a restocking fixed-route decision rule, denote by $\pi(v) = (\pi(v^m))_{m\in\mathcal{M}}$ the restocking fixed-route policy associated with the set of fixed routes v, which consists of a separate policy for each vehicle m in $\mathcal{M}$. At decision epoch k, suppose the process occupies state $s_k = (l, t, q, d, x)$. Denote by T_k the time at which decision epoch k occurs and by $\mathcal{M}'$ the set of vehicles that reach their destinations at time T_k. Let v_b^m be the first customer on the fixed route for vehicle m that will have pending or unknown demand after demand at the current location is served to the fullest extent possible. Customer v_b^m may be the current location if available capacity is insufficient to meet demand at the vehicle's current location. If demand at all customers on the fixed route will be served during the current period, then let v_b^m be the depot. Denote by $\widehat{V}^{\pi(v^m)}_{\text{replen}}(s_k)$ the estimated expected demand served with perfect information from customer v_b^m onward when the process occupies state s_k and capacity is replenished immediately prior to visiting v_b^m. Similarly, denote by $\widehat{V}^{\pi(v^m)}_{\text{direct}}(s_k)$ the estimated expected demand served with perfect information from customer v_b^m onward when the process occupies state s_k and the vehicle travels directly from its current location to v_b^m. We explain the calculation of $\widehat{V}^{\pi(v^m)}_{\text{replen}}(s_k)$ and $\widehat{V}^{\pi(v^m)}_{\text{direct}}(s_k)$ in §4.3. Then, the fixed-route decision rule for vehicle m is

$$\delta_k^{\pi(v^m)}(s_k) = \begin{cases} l_m, & m \notin \mathcal{M}', \\ 0, & m \in \mathcal{M}' \text{ and } q_m \leq d_{l_m}, \\ 0, & m \in \mathcal{M}' \text{ and } T_k + t(l_m, v_b^m) + t(v_b^m, 0) > L, \\ v_b^m, & m \in \mathcal{M}' \text{ and } l_m = 0, \\ 0, & m \in \mathcal{M}' \text{ and } \widehat{V}^{\pi(v^m)}_{\text{replen}}(s_k) > \widehat{V}^{\pi(v^m)}_{\text{direct}}(s_k), \\ v_b^m, & \text{otherwise.} \end{cases} \tag{1}$$

The first case in equation (1) requires a vehicle en route to continue to its destination. The second case assigns a vehicle to the depot if capacity will be depleted after serving demand at its current location. The third case assigns a vehicle to the depot if traveling to customer v_b^m will result in a violation of the route duration limit. The fourth case directs the vehicle to customer v_b^m if the vehicle is currently at the depot. The fifth case assigns the vehicle to the depot if preemptively replenishing capacity results in a larger estimate of future demand served than directly proceeding to the next customer v_b^m. The sixth case assigns the vehicle to location v_b^m. The decision rule for policy $\pi(v)$ consists of the decision rules for each fixed route composing v: $\delta_k^{\pi(v)} = (\delta_k^{\pi(v^m)})_{m \in \mathcal{M}}$.

We note that although both a priori and restocking policies are characterized by fixed sequences of customers, the decision rules associated with each policy type are different. Because decision rule (1) permits preemptive capacity replenishment, the restocking policy action space is a superset of the a priori policy action space. Thus, following an optimal restocking policy along a given fixed route results in a larger expected demand served than following an a priori policy along the same fixed route.

4.2 Auxiliary Dynamic Program

In §4.2.1, we formulate an auxiliary deterministic dynamic program to calculate the demand served by an optimal restocking policy when customer demands are known along fixed route $v^m = (v_1^m, v_2^m, \dots, v_{B^m}^m)$ for $m \in \mathcal{M}'$. Recall from §4.1 that v_b^m is the first customer on the fixed route that vehicle m will visit after the current location $v_1^m = l_m$ is served to the fullest extent possible. Therefore, the auxiliary dynamic program involves the sequence of customers $(l_m, v_b^m, v_{b+1}^m, \dots, v_{B^m}^m)$. To ease notation, in this section we refer to this sequence as the fixed route $v^m = (l_m, v_b^m, v_{b+1}^m, \dots, v_{B^m}^m) = (v_1^m, v_2^m, \dots, v_{C^m}^m)$. In §4.2.2, we establish structural properties which we then exploit in the solution method outlined in §4.2.3. Solving the dynamic program for each realization in a sample of customer demands facilitates estimation of the reward-to-go for a restocking policy and the calculation of the quantities $\widehat{V}_{\text{replen}}^{\pi(v^m)}(\cdot)$ and $\widehat{V}_{\text{direct}}^{\pi(v^m)}(\cdot)$ in equation (1).

4.2.1 Formulation

The auxiliary dynamic program has C^m stages, one for each customer on the fixed route. At stage c, the state variable is the pair $\tilde{s}_c = (Q_{v_c^m}, A_{v_c^m})$, which denotes vehicle capacity upon initial arrival to customer v_c^m and time of initial arrival to customer v_c^m, respectively. Actions available at stage

c are $\tilde{a}^r$ and $\tilde{a}^d$, representing preemptive capacity replenishment before traveling to customer v^m_{c+1} and direct travel to customer v^m_{c+1}, respectively. We denote the transition function via $\tilde{S}(\cdot) = (\tilde{Q}(\cdot), \tilde{A}(\cdot))$, which returns the next state as a function of the current state and the selected action. The stage-c reward is $Z_c(\tilde{s}_c)$, the demand served at customer v^m_c. We seek a policy that maximizes the sum of the demand served across all stages: $\sum_{c=1}^{C^m} Z_c(\tilde{s}_c)$. We provide additional details below.

The quantities $Q_{v^m_c}$ and $A_{v^m_c}$ depend on the state at stage $c-1$ and the action selected at stage $c-1$. As in §3, $t(n, n')$ denotes the time required to travel from location n to location n'. If action $\tilde{a}^d_{c-1}$ is selected at stage $c-1$, we separate the calculation of $\tilde{Q}(\tilde{s}_{c-1}, \tilde{a}^d_{c-1})$ and $\tilde{A}(\tilde{s}_{c-1}, \tilde{a}^d_{c-1})$ into three cases:

$$\tilde{Q}(\tilde{s}_{c-1}, \tilde{a}^d_{c-1}) = \begin{cases} Q_{v^m_{c-1}} - x_{v^m_{c-1}}, & x_{v^m_{c-1}} < Q_{v^m_{c-1}}, \\ Q, & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} = \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil, \\ \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil Q - x_{v^m_{c-1}} + Q_{v^m_{c-1}}, & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} < \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil, \end{cases} \tag{2}$$

and

$$\tilde{A}(\tilde{s}_{c-1}, \tilde{a}^d_{c-1}) = \begin{cases} A_{v^m_{c-1}} + t(v^m_{c-1}, v^m_c), & x_{v^m_{c-1}} < Q_{v^m_{c-1}}, \\ A_{v^m_{c-1}} + \left(\frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} + 1 \right) t(v^m_{c-1}, 0) & \\ \qquad + \left(\frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right) t(0, v^m_{c-1}) + t(0, v^m_c), & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} = \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil, \\ A_{v^m_{c-1}} + t(v^m_{c-1}, v^m_c) + \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil & \\ \qquad \times \left(t(v^m_{c-1}, 0) + t(0, v^m_{c-1}) \right), & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} < \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil, \end{cases} \tag{3}$$

where the boundary conditions $v^m_1 = l_m$, $Q_{v^m_1} = q_m$, and $A_{v^m_1} = t_m$ are given by the current state of vehicle m in the original MDP. In the first case, demand at customer v^m_{c-1} is less than vehicle capacity upon arrival to v^m_{c-1}, thus vehicle capacity is simply decremented by the amount of demand and the vehicle travels directly from v^m_{c-1} to v^m_c. The second and third cases account for situations where demand at customer v^m_{c-1} is greater than or equal to vehicle capacity upon arrival to v^m_{c-1}, thereby requiring return trips to the depot to replenish capacity. The required number of return trips is $\lceil (x_{v^m_{c-1}} - Q_{v^m_{c-1}})/Q \rceil$. In the second case, satisfying demand at v^m_{c-1} exactly depletes vehicle capacity, thus requiring the vehicle to replenish at the depot one additional time and travel directly to customer v^m_c with full capacity. In the third case, there is some capacity remaining after serving demand at customer v^m_{c-1}. After making the necessary return trips to the depot, the vehicle travels directly from v^m_{c-1} to v^m_c with the remaining capacity.

If action $\tilde{a}^r_{c-1}$ is selected at stage $c-1$, then capacity replenishment yields full vehicle capacity upon arrival to customer v^m_c: $\tilde{Q}(\tilde{s}_{c-1}, \tilde{a}^r_{c-1}) = Q$. We separate the calculation of the initial arrival time to customer v^m_c into the same three cases considered in equation (3):

$$\tilde{A}(\tilde{s}_{c-1}, \tilde{a}^r_{c-1}) = \begin{cases} A_{v^m_{c-1}} + t(v^m_{c-1}, 0) + t(0, v^m_c), & x_{v^m_{c-1}} < Q_{v^m_{c-1}}, \\ A_{v^m_{c-1}} + \left(\frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} + 1\right) t(v^m_{c-1}, 0) & \\ \quad + \left(\frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q}\right) t(0, v^m_{c-1}) + t(0, v^m_c), & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} = \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil, \\ A_{v^m_{c-1}} + t(v^m_{c-1}, 0) + t(0, v^m_c) + \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil & \\ \quad \times \left(t(v^m_{c-1}, 0) + t(0, v^m_{c-1}) \right), & \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} < \left\lceil \frac{x_{v^m_{c-1}} - Q_{v^m_{c-1}}}{Q} \right\rceil. \end{cases} \tag{4}$$

Equation (4) is similar to equation (3), except that in the first and third cases, vehicle capacity is preemptively replenished via an additional trip to the depot.

To compute $Z_c(\tilde{s}_c)$, we consider the route duration limit, noting violations of the route duration limit result in zero demand served. Four cases are considered in the calculation:

$$Z_c(\tilde{s}_c) = \begin{cases} 0, & A_{v^m_c} > L - t(v^m_c, 0), \\ \left\lfloor \frac{L - t(v^m_c, 0) - A_{v^m_c}}{t(v^m_c, 0) + t(0, v^m_c)} \right\rfloor Q + Q_{v^m_c}, & \left\lfloor \frac{L - t(v^m_c, 0) - A_{v^m_c}}{t(v^m_c, 0) + t(0, v^m_c)} \right\rfloor < \left\lceil \frac{x_{v^m_c} - Q_{v^m_c}}{Q} \right\rceil, \\ x_{v^m_c}, & x_{v^m_c} \le Q_{v^m_c} \text{ and } A_{v^m_c} \le L - t(v^m_c, 0), \\ x_{v^m_c}, & \left\lfloor \frac{L - t(v^m_c, 0) - A_{v^m_c}}{t(v^m_c, 0) + t(0, v^m_c)} \right\rfloor \ge \left\lceil \frac{x_{v^m_c} - Q_{v^m_c}}{Q} \right\rceil. \end{cases} \tag{5}$$

In the first case, zero demand is served because the route duration limit is violated. In the second case, only a portion of demand is served because the vehicle does not have enough time to make the return trips to the depot necessary to serve demand in full. In the third case, demand is served in full because the vehicle arrives prior to the route duration limit and with sufficient capacity. In the fourth case, demand is served in full because the vehicle has enough time to make any necessary replenishments.

Denote by $\tilde{V}_c(\tilde{s}_c; v^m, x)$ the reward-to-go at stage c when the state is $\tilde{s}_c$, the fixed route is v^m, and customer demands are given by the vector x. An optimal policy can be obtained by solving for all possible states $\tilde{s}_c$

$$\tilde{V}_c(\tilde{s}_c; v^m, x) = Z_c(\tilde{s}_c) + \max_{a \in \{a_c^d, a_c^r\}} \left\{ \tilde{V}_{c+1}\left(\tilde{s}_{c+1} = \tilde{S}(\tilde{s}_c, a)\right)\right\}, \tag{6}$$

for $c = 1, \ldots, C^m - 1$ and $\tilde{V}_{C^m}(\tilde{s}_{C^m}; v^m, x) = Z_{C^m}(\tilde{s}_{C^m})$. In §4.2.2 and §4.2.3, we propose methods to solve these optimality equations.

4.2.2 Structural Properties

A sequence of states from stage 1 to stage C^m in the auxiliary dynamic program represents one possible sequence of arrival time and capacity upon arrival to each customer $v_1^m, \ldots, v_{C^m}^m$ on the fixed route. Consider a sequence of states $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{c'}, \ldots, \tilde{s}_{C^m}$, where c' is the first stage such that the demand at customer $v_{c'}^m$ is not fully served, i.e., $Z(\tilde{s}_{c'}) < x_{v_{c'}^m}$. By equation (5), it must be that demand is not fully served because doing so violates the route duration limit. Assuming the triangle inequality holds for travel times, then because we require demand to be served in full at customer $v_{c'}^m$ before proceeding to customer $v_{c'+1}^m$, it is not possible to serve any demand at customers $v_{c'+1}^m, v_{c'+2}^m, \ldots, v_{C^m}^m$ because doing so violates the route duration limit. We formalize this observation in Proposition 1 (see Appendix A for the proof). In §4.2.3, we use this result to prune the state space in a forward dynamic programming solution approach.

Proposition 1. *Assume travel times $t(\cdot, \cdot)$ satisfy the triangle inequality. Consider a sequence of states $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{C^m}$. Let c' be the smallest c such that $Z_c(\tilde{s}_c) < x_{v_c^m}$. If c' exists, then:*

(i). $Z_c(\tilde{s}_c) = 0$ *for* $c = c' + 1, c' + 2, \ldots, C^m$*; and*

(ii). $Z_c(\tilde{s}_c) = x_{v_c^m}$ *for* $c = 1, 2, \ldots, c' - 1$.

Proposition 2 shows the demand served from a given state $\tilde{s}_c$ through $\tilde{s}_{C^m}$ is non-decreasing as the capacity upon arrival $Q_{v_c^m}$ increases and as the arrival time $A_{v_c^m}$ decreases (see Appendix A for the proof). We use this dominance relationship, in conjunction with Proposition 1, to further prune the state space in the forward dynamic programming approach we develop in §4.2.3.

Proposition 2. *For* $\tilde{s}_c = (Q_{v_c^m}, A_{v_c^m})$ *and* $\tilde{s}'_c = (Q'_{v_c^m}, A'_{v_c^m})$ *such that* $0 \leq Q_{v_c^m} \leq Q'_{v_c^m}$ *and* $0 \leq A'_{v_c^m} \leq A_{v_c^m}$, $\tilde{V}_c(\tilde{s}_c; v^m, x) \leq \tilde{V}_c(\tilde{s}'_c; v^m, x)$.

4.2.3 Solution Approach

Because we do not require travel times to be discrete, the state space of the auxiliary dynamic program is infinite, thereby rendering the standard backward dynamic programming procedure computationally intractable. Instead, we develop a forward dynamic programming approach that only considers states that may actually occur as a result of a given initial state. Our approach mirrors the well-known reaching algorithm for dynamic programming (Denardo, 2003) and leverages the structural results of §4.2.2 to prune the state-space graph. Our forward dynamic programming procedure is able to obtain the values of optimal restocking policies for large fixed routes. Without the pruning afforded by our structural results, our forward approach enumerates the entire solution space. We note our structural results and forward solution approach are also applicable to the discrete-time case.

Our solution approach utilizes a graph structure. Each node in the graph is labeled by a state, the demand served in that state, and the total demand served so far. Thus, a node associated with a state $\tilde{s}_c$ is represented by the tuple $(\tilde{s}_c = (Q_{v_c^m}, A_{v_c^m}), Z_c(\tilde{s}_c), \lambda_c = \sum_{i=1}^{c} Z_i(\tilde{s}_i))$. Nodes are connected by arcs representing actions. An arc from a node $\tilde{s}_{c-1}$ to a node $\tilde{s}_c$ represents an action $\tilde{a}_{c-1} \in \{\tilde{a}^d_{c-1}, \tilde{a}^r_{c-1}\}$ denoting the decision to travel directly from v^m_{c-1} to v^m_c or to first replenish capacity at the depot. The graph is constructed in stages, one for each customer on the fixed route. We refer to the set of nodes belonging to stage c as Λ_c. Stage 1 of the graph is constructed via $\tilde{s}_1$, the given initial state of the vehicle. Stage 2 is constructed by *extending* the initial node corresponding to state $\tilde{s}_1$, i.e., by generating the states $\tilde{S}(\tilde{s}_1, \tilde{a}^d)$ and $\tilde{S}(\tilde{s}_1, \tilde{a}^r)$ that result by taking actions $\tilde{a}^d_1$ and $\tilde{a}^r_1$ from the initial state. Any stage $c + 1$ is constructed in a similar manner by extending the nodes in stage Λ_c. The node in Λ_{C^m} that achieves the largest λ_{C^m} indicates the value of the optimal restocking policy for the given fixed route. The optimal policy is represented by the sequence of actions leading to this node.

The total number of nodes in the graph is $2^{C^m} - 1$, where C^m is the number of customers on the fixed route. The total number of node sequences from stage 1 through stage C^m, each of which represents a policy, is $2^{C^m - 1}$. In our computational experience, storing the graph in memory becomes problematic as C^m approaches 25. Further, because the local search on restocking policies we discuss in §5 requires us to obtain optimal restocking policies for many fixed routes, evaluating fixed routes can be computationally prohibitive even when C^m is much smaller.

These computational issues can be mitigated by exploiting Propositions 1 and 2. Consider a partial path through the graph, which we denote by the sequence of states $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{c'}$, where c' is the first stage along this path such that the demand at customer $v^m_{c'}$ is not fully served. By Proposition 1, any extension of this path will result in zero demand served at customers $v^m_{c'+1}, v^m_{c'+2}, \ldots, v^m_{C^m}$, i.e., $Z_c(\tilde{s}_c) = 0$ for $c = c' + 1, c' + 2, \ldots, C^m$. Thus, to obtain the value of the optimal restocking policy, it is only necessary to extend nodes that fully serve demand. Thus, at stage c, we only extend nodes in the set $\Lambda'_c = \{(\tilde{s}_c, Z_c(\tilde{s}_c), \lambda_c) : Z_c(\tilde{s}_c) = x_{v^m_c}\}$.

Additional pruning is possible by using the result of Proposition 2 to further refine Λ'_c. For any two nodes $(\tilde{s}_c = (Q_{v^m_c}, A_{v^m_c}), Z_c(\tilde{s}_c), \lambda_c)$ and $(\tilde{s}'_c = (Q'_{v^m_c}, A'_{v^m_c}), Z_c(\tilde{s}'_c), \lambda_c)$ in Λ'_c, if $Q_{v^m_c} \leq Q'_{v^m_c}$ and $A'_{v^m_c} \leq A_{v^m_c}$, then it is not necessary to extend $\tilde{s}_c$ because the total demand served by extending $\tilde{s}'_c$ will be at least as large. More formally, let $\Lambda''_c = \{(\tilde{s}_c, \cdot, \cdot) \in \Lambda'_c : \nexists\ (\tilde{s}'_c, \cdot, \cdot) \in \Lambda'_c$ such that either $Q_{v^m_c} < Q'_{v^m_c}, A'_{v^m_c} \leq A_{v^m_c}$ or $Q_{v^m_c} \leq Q'_{v^m_c}, A'_{v^m_c} < A_{v^m_c}\}$ be the set of *non-dominated* nodes in Λ'_c. Proposition 2 guarantees that an optimal policy will be obtained by extending only the non-dominated nodes in the set $\Lambda''_c \subseteq \Lambda'_c$.

Algorithm 1 details our forward dynamic programming procedure. The EVALUATE$(v^m, x, \tilde{s}_1)$ procedure in Algorithm 1 takes as input a fixed route v^m, customer demands x, and initial state $\tilde{s}_1$ at customer v^m_1. It returns $\tilde{V}_1(\tilde{s}_1; v^m, x)$, the value of the optimal restocking policy for fixed route v^m when customer demands are x and the initial state at customer v^m_1 is $\tilde{s}_1$. The procedure begins on line 2 by initializing Λ_1 with the given initial state $\tilde{s}_1$, $Z_1(\tilde{s}_1)$, and $\lambda_1 = Z_1(\tilde{s}_1)$, the demand served in state $\tilde{s}_1$. For $c = 2, \ldots, C^m$, Λ_c is initialized to the empty set. Line 4 begins the process of identifying nodes in Λ_c to extend. Per Proposition 1, states not fully serving demand at customer v_c^m need not be extended. Thus, $\Lambda'_c \subseteq \Lambda_c$ restricts attention to nodes such that $Z_c(\tilde{s}_c) = x_{v_c^m}$. If Λ'_c is empty, then, by Proposition 1, it is not necessary to extend any nodes because doing so will not increase the total demand served. Line 6 accomplishes this by exiting the for-loop. In line 7, the set of nodes to be extended is further refined by applying Proposition 2. When identifying nodes for inclusion in Λ''_c, it may be that several nodes are identical, and therefore none of the nodes dominates the others. In such cases, we extend one of these nodes, provided it is not dominated by another node. Lines 9 and 10 construct Λ_{c+1} by extending the nodes in Λ''_c. Line 11 identifies the non-empty node set with the largest index, $\bar{c}$. Finally, the procedure returns the maximum demand served by the nodes in $\Lambda_{\bar{c}}$.

Algorithm 1 Valuation of Optimal Restocking Policy

1: **procedure** EVALUATE($v^m, x, \tilde{s}_1$)
2: $\quad \Lambda_1 \leftarrow \{(\tilde{s}_1, Z_1(\tilde{s}_1), \lambda_1 = Z_1(\tilde{s}_1))\}$, $\Lambda_c \leftarrow \emptyset$ for $c = 2, 3, \ldots, C^m$
3: $\quad$ **for** $c = 1$ **to** $C^m - 1$ **do**
4: $\quad\quad \Lambda'_c \leftarrow \{(\tilde{s}_c, Z_c(\tilde{s}_c), \lambda_c) \in \Lambda_c : Z_c(\tilde{s}_c) = x_{v_c^m}\}$
5: $\quad\quad$ **if** $\Lambda'_c = \emptyset$ **then**
6: $\quad\quad\quad$ break
7: $\quad\quad \Lambda''_c \leftarrow \{(\tilde{s}_c, \cdot, \cdot) \in \Lambda'_c : \nexists\, (s'_c, \cdot, \cdot) \in \Lambda'_c$ such that $Q_{v_c^m} \leq (<) Q'_{v_c^m}, A'_{v_c^m} < (\leq) A_{v_c^m}\}$
8: $\quad\quad$ **for** $(\tilde{s}_c, \lambda_c) \in \Lambda''_c$ **do**
9: $\quad\quad\quad \Lambda_{c+1} \leftarrow \Lambda_{c+1} \cup \{(\tilde{s}_{c+1} = (\tilde{Q}(\tilde{s}_c, \tilde{a}_c^d), \tilde{A}(\tilde{s}_c, \tilde{a}_c^d)), Z_{c+1}(\tilde{s}_{c+1}), \lambda_c + Z_{c+1}(\tilde{s}_{c+1})\}$
10: $\quad\quad\quad \Lambda_{c+1} \leftarrow \Lambda_{c+1} \cup \{(\tilde{s}_{c+1} = (\tilde{Q}(\tilde{s}_c, \tilde{a}_c^r), \tilde{A}(\tilde{s}_c, \tilde{a}_c^r)), Z_{c+1}(\tilde{s}_{c+1}), \lambda_c + Z_{c+1}(\tilde{s}_{c+1})\}$
11: $\quad \bar{c} \leftarrow$ largest $c \in \{1, 2, \ldots, C^m\}$ such that $\Lambda_c \neq \emptyset$
12: $\quad$ **return** $\max\{\lambda_{\bar{c}} : (\tilde{s}_{\bar{c}}, Z_{\bar{c}}(\tilde{s}_{\bar{c}}), \lambda_{\bar{c}}) \in \Lambda_{\bar{c}}\}$

The example in Table 1 and Figure 1 illustrates Algorithm 1. Table 1 displays data for a fixed route $v^m = (23, 22, 13, 15, 2)$ from problem instance R101(25) of Solomon (1987) with vehicle capacity $Q = 50$ and route duration limit $L = 171.681$. The column labeled "$x_{v_c^m}$" displays the sampled demand at customer v_c^m. The horizontal and vertical coordinates of each customer

Table 1: Example Data

c	v^m_c	$x_{v^m_c}$	x-coord	y-coord
1	23	39	55	5
2	22	18	45	10
3	13	23	30	25
4	15	8	30	5
5	2	7	35	17

are given in the columns labeled “x-coord” and “y-coord,” respectively. The depot is located at $(35, 35)$ and one time unit is equal to one unit of distance, as measured by the Euclidian metric.

Figure 1 depicts the full graph for the fixed route in Table 1. For convenience, each node is numbered in the upper-left corner. We demonstrate Algorithm 1 by stepping through the construction of the graph in Figure 1. Given $v^m = (23, 22, 13, 15, 2)$, $x = (39, 18, 23, 8, 7)$, and $\tilde{s}_1 = (50, 36.0555)$, the procedure begins by calling $\text{EVALUATE}(v^m, x, \tilde{s}_1)$. Node set Λ_1 is initialized via $\tilde{s}_1$, $Z_1(\tilde{s}_1) = 39$, and $\lambda_1 = 39$. Because $\Lambda_1 = \Lambda'_1 = \Lambda''_1 = \{1\}$, node 1 is extended to create $\Lambda_2 = \{2, 3\}$. Because $\Lambda_2 = \Lambda'_2 = \Lambda''_2$, nodes 2 and 3 are extended to create $\Lambda_3 = \{4, 5, 6, 7\}$. All four nodes in Λ_3 fully serve demand at customer $v^m_3 = 13$, thus $\Lambda'_3 = \Lambda_3$. However, node 5 is dominated by node 7, thus we only extend the nodes in $\Lambda''_3 = \{4, 6, 7\}$ to create $\Lambda_4 = \{8, 9, 12, 13, 14, 15\}$. Only node 12 fully serves demand at customer $v^m_4 = 15$, thus $\Lambda'_4 = \Lambda''_4 = \{12\}$. Extending node 12 results in $\Lambda_5 = \{24, 25\}$. The procedure concludes by returning $\max\{89, 88\} = 89$, which is the value of the optimal restocking policy for fixed route v^m when demand is x and the initial state is $\tilde{s}_1$. The optimal policy is represented by the sequence of nodes 1, 3, 6, 12, 24, which corresponds to an optimal sequence of actions $\tilde{a}^r_1, \tilde{a}^d_2, \tilde{a}^d_3, \tilde{a}^d_4$.

In this example, Algorithm 1 decreases the number of nodes in the graph from 31 to 15. In our experience, for large fixed routes, Algorithm 1 can decrease the number of nodes by several orders of magnitude, thereby making it computationally feasible to solve the dynamic program required to obtain an optimal restocking policy for a given fixed route. For smaller problem instances where the sequence of customers in a fixed route is typically shorter, we observe runtime reductions up to 75 percent when embedding Algorithm 1 in the local search heuristic we discuss in §5.

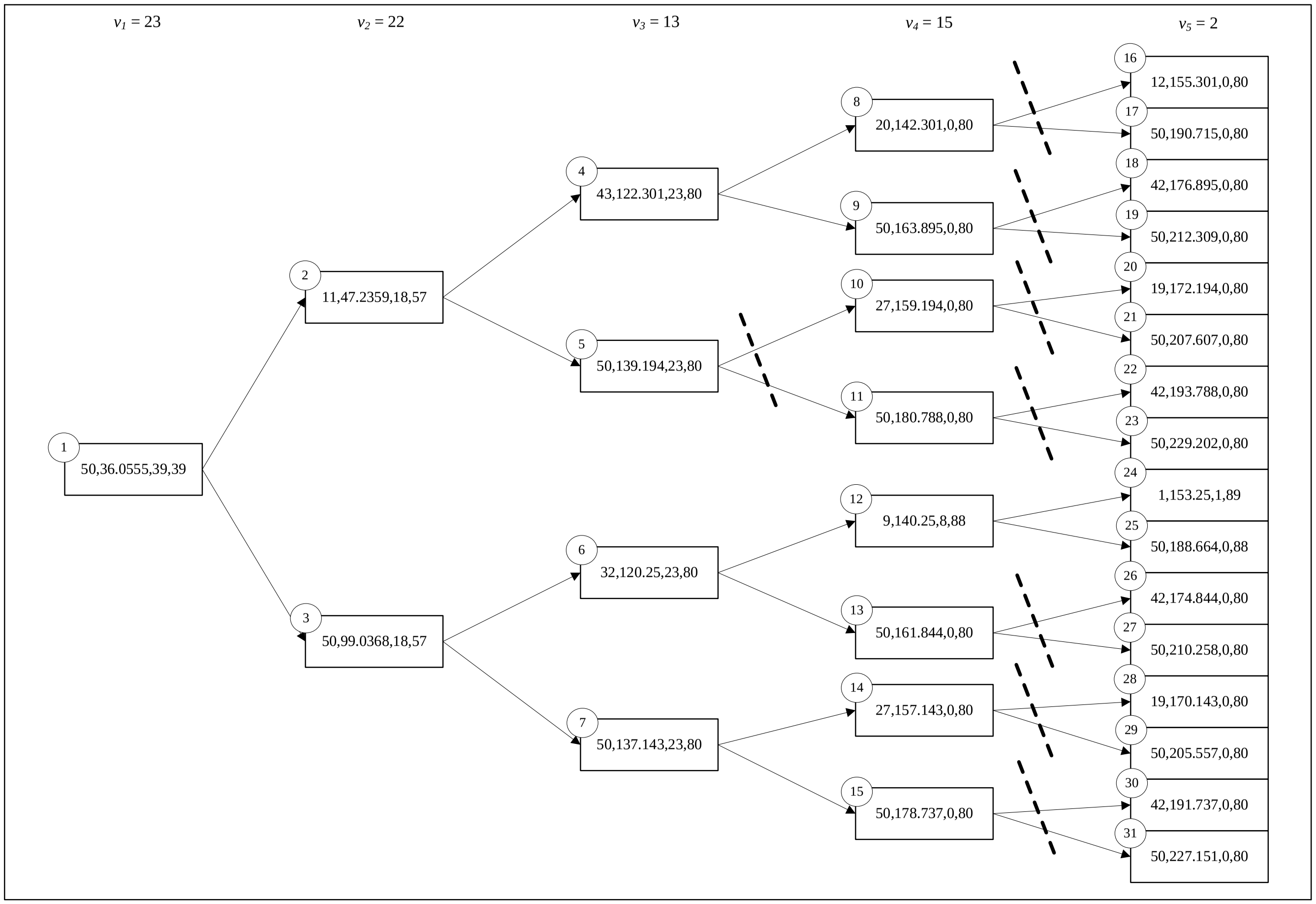

Figure 1: Example Dynamic Program

4.3 Estimation of Policy Value

Using the notation of §4.2.1, we can now define the quantities $\widehat{V}^{\pi(v^m)}_{\text{replen}}(\cdot)$ and $\widehat{V}^{\pi(v^m)}_{\text{direct}}(\cdot)$ used in the decision rule of equation (1) in the original MDP. Recall $\widehat{V}^{\pi(v^m)}_{\text{replen}}(\cdot)$ is the estimated expected demand served with perfect information when preemptively restocking capacity prior to continuing to the next customer and $\widehat{V}^{\pi(v^m)}_{\text{direct}}(\cdot)$ is the estimated expected demand served with perfect information when traveling directly to the next customer. Let $\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^P$ be P customer demand vectors generated randomly from distribution function F. Then,

$$\widehat{V}^{\pi(v^m)}_{\text{replen}}(s_k) = \frac{1}{P}\sum_{p=1}^{P} \tilde{V}_2\left(\tilde{S}\left((q_m, t_m), \tilde{a}^r\right); v^m, \hat{x}^p\right) \tag{7}$$

and

$$\widehat{V}^{\pi(v^m)}_{\text{direct}}(s_k) = \frac{1}{P}\sum_{p=1}^{P} \tilde{V}_2\left(\tilde{S}\left((q_m, t_m), \tilde{a}^d\right); v^m, \hat{x}^p\right), \tag{8}$$

Recalling the notation for fixed route $v^m = (l_m, v^m_b, v^m_{b+1}, \ldots, v^m_{B^m}) = (v^m_1, v^m_2, \ldots, v^m_{C^m})$, the quantity $\tilde{V}_2(\tilde{S}((q_m, t_m), \tilde{a}^r); v^m, \hat{x}^p)$ is the demand served from customer v^m_b onward when customer demands are $\hat{x}^p$, vehicle capacity at location l_m is q_m, time of arrival to l_m is t_m, and capacity is replenished immediately prior to visiting v^m_b. The quantity $\tilde{V}_2(\tilde{S}((q_m, t_m), \tilde{a}^d); v^m, \hat{x}^p)$ is analogous, except the vehicle travels directly from l_m to v^m_b, foregoing the opportunity to restock at the depot. We also note estimating expected demand served with perfect information is equivalent to the perfect information relaxation with zero penalty (Brown et al., 2010).

In §5, we estimate the value of restocking policies as part of local search and rollout procedures. Denote by

$$\widehat{V}^{\pi(v^m)}(s) = \frac{1}{P}\sum_{p=1}^{P} \tilde{V}_1\left((q_m, t_m); v^m, \hat{x}^p\right) \tag{9}$$

the estimated expected demand served from state s onward by restocking fixed-route policy $\pi(v^m)$. When the original MDP occupies a pre-decision state s_k, demand is known at current vehicle locations l. In this case, equation (9) is related to equations (7) and (8) via the period-k reward in the original MDP:

$$\widehat{V}^{\pi(v^m)}(s_k) = R_k(s_k, \cdot) + \max\left\{\widehat{V}^{\pi(v^m)}_{\text{direct}}(s_k), \widehat{V}^{\pi(v^m)}_{\text{replen}}(s_k)\right\}. \tag{10}$$

Finally, denote by

$$\widehat{V}^{\pi(v)}(s) = \sum_{m \in \mathcal{M}} \widehat{V}^{\pi(v^m)}(s) \tag{11}$$

the estimated expected demand served from state s onward by policy $\pi(v)$, the collection of restocking fixed-route policies for the vehicles comprising $\mathcal{M}$.

5 Rollout Policies

We employ rollout procedures to dynamically adjust routing plans for the VRPSDL. Rollout algorithms are heuristic versions of policy iteration for dynamic programs and employ the concept of forward dynamic programming where decision rules are calculated only for observed states (Bertsekas et al., 1997; Bertsekas, 2000). From a current state, the reward-to-go is approximated by heuristic policies, and these approximations are then used to guide action selection in the current state.

Using the rollout policy framework of Goodson et al. (2013), Goodson et al. (2013) develop a priori-based rollout procedures for the VRPSDL. We utilize the same framework, but seek to improve upon the a priori-based heuristic via a restocking-based heuristic. The restocking-based heuristic relaxes the a priori restricted policy class of Goodson et al. (2013) by giving explicit consideration to preemptive capacity replenishment actions. Further, because the expected demand served by a restocking policy is at least as large as the expected demand served by an a priori policy along the same fixed route, a restocking-based heuristic can potentially lead to better action selection in the rollout procedure. We describe three restocking fixed-route heuristics in §5.1 and the rollout policies in §5.2.

5.1 Restocking Fixed-Route Heuristics

We consider three heuristic search mechanisms for restocking fixed-route policies: a *random heuristic*, a *local search heuristic*, and a *variable neighborhood search heuristic*. We use these three heuristics in §6 to study the effect of more sophisticated search mechanisms.

The random heuristic randomly selects a set of fixed routes from $\mathcal{V}(s)$, the space of all fixed routes when the process occupies pre- or post-decision state s. The policy returned by the random heuristic is the restocking policy associated with the selected set of fixed routes.

As in Goodson et al. (2013), the local search heuristic obtains heuristic policies via a first-improving local search on 1-relocation neighborhoods of $\mathcal{V}(s)$. The 1-relocation neighborhood of a set of fixed routes v can be obtained by relocating a customer v_i^m after another customer $v_j^{m'}$ such that $i \neq 1$. The condition $i \neq 1$ ensures vehicles' current destinations, given by state s, are fixed for all solutions in the neighborhood.

Each iteration of the local search proceeds as follows. Given a current set of restocking fixed routes v, a set of restocking fixed routes $\bar{v}$ is randomly selected from the relocation neighborhood of v. Recalling the notation of §4.3, if the estimated value of policy $\pi(\bar{v})$ is greater than the estimated value of policy $\pi(v)$ (i.e., $\widehat{V}^{\pi(\bar{v})}(s) > \widehat{V}^{\pi(v)}(s)$), then the current set of fixed routes is updated to be $\bar{v}$ and the process repeats. Otherwise, the search continues by randomly selecting another set of fixed routes from the relocation neighborhood of v. The procedure terminates and returns fixed-route policy $\pi(v)$ if there does not exist a set of fixed routes $\bar{v}$ in the relocation neighborhood of v such that $\widehat{V}^{\pi(\bar{v})}(s) > \widehat{V}^{\pi(v)}(s)$.

When calculating $\widehat{V}^{\pi(\bar{v})}(s)$, it may not be necessary to execute the full auxiliary dynamic program. The structures of the state-space graphs associated with routes v^m and $\bar{v}^m$ for a given demand sample are identical up until the point of insertion or removal of a customer. Consequently, if the same demand samples are used to calculate $\widehat{V}^{\pi(v^m)}(s)$ and $\widehat{V}^{\pi(\bar{v}^m)}(s)$, then the state-space graphs used to calculate $\widehat{V}^{\pi(\bar{v}^m)}(s)$ need only be reconstructed from the point of insertion or removal onward.

More generally, let $\bar{v}$ be obtained by relocating a customer v_i^m after customer $v_j^{m'}$. If $m = m'$ and $i < j$, then it is only necessary to reconstruct the state-space graphs for route m from stage

i onward. For a given demand sample, this is accomplished by beginning Algorithm 1 at the i^{th} iteration of line 3, which begins construction of the nodes in stage i by extending the nodes in stage $i-1$, the largest stage common to the state-space graphs for routes v^m and $\bar{v}^m$. If $m = m'$ and $i \geq j$, then it is only necessary to reconstruct the state-space graphs for route m from stage $j+1$ onward. Similarly, if $m \neq m'$, then it is only necessary to reconstruct the state space graphs from stage i onward on route m and from stage $j+1$ onward on route m'. Our computational experience suggests updating state-space graphs in this fashion significantly reduces the computation required to evaluate restocking fixed-route policies in the relocation neighborhood of a current policy.

The variable neighborhood search heuristic follows Figure 1 of Hansen and Mladenović (2001). We employ the first-improving search on the 1-relocation neighborhood in the local search phase and extend the 1-relocation neighborhood to a k-relocation neighborhood for the shaking phase. A k-relocation neighbor for a set of fixed routes is a set of fixed routes obtained by moving a sequence of k locations to new positions. The variable neighborhood search proceeds as follows. Given an initial solution and beginning with k set to 1, we repeat the following steps until k reaches 3. First, randomly select a set of fixed routes from the k-relocation neighborhood of the current solution. Second, apply the first-improving local search method to the selected set of fixed routes. Third, if the resulting set of fixed routes yields a better restocking policy than the incumbent solution, update the incumbent and reset k to 1; otherwise increment k. Once k reaches 3, the incumbent solution is compared to the initial solution. If the incumbent solution is better than the initial solution, the three-step procedure is repeated with the incumbent set as the initial solution; otherwise, the search terminates.

5.2 One-Step, Post-Decision, and Pre-Decision Rollout

Figure 2 provides a graphical depiction of our rollout policies. Figure 2a depicts the MDP formulation of the VRPSDL as a decision tree: square nodes represent pre-decision states, solid arcs are feasible actions, circle nodes symbolize post-decision states, and dashed arcs signify observed customer demands. Figure 2b displays one-step rollout within the decision tree framework. From a current state s_k, one-step rollout estimates the reward-to-go for selecting action a by first mak-

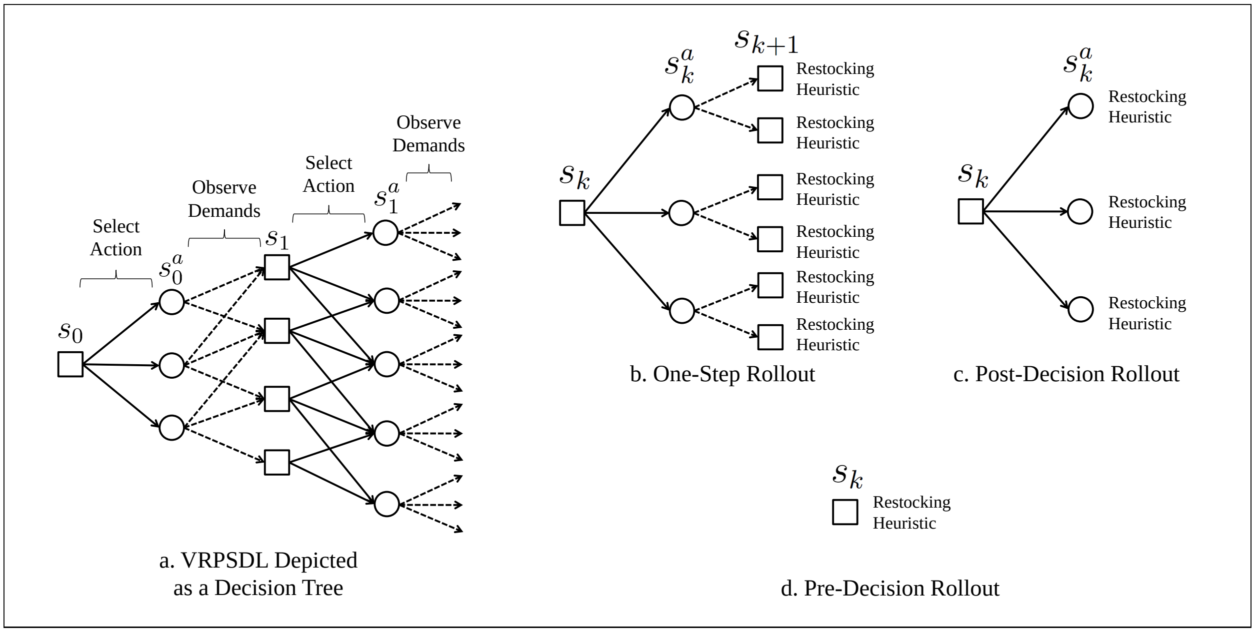

Figure 2: Rollout Policies in the Context of Decision Trees

ing transitions to all pre-decision states s_{k+1} that may be possible at the next decision epoch as a result of each action a and observing customer demands. From each state s_{k+1}, we execute one of the restocking fixed-route heuristics of §5.1, which we denote by "Restocking Heuristic." The expected value of the policies returned by the executions of the restocking fixed-route heuristic serve as the estimate of the reward-to-go for selecting action a. One-step rollout selects an action that maximizes the sum of the current-period reward plus the estimated reward-to-go.

In the post-decision rollout policy depicted in Figure 2, we transition to the post-decision state s_k^a associated with an action a rather than to all possible pre-decision states at the next decision epoch. The estimate of the reward-to-go when selecting action a in state s_k is the estimated expected value of the restocking fixed-route policy obtained when executing a heuristic from post-decision state s_k^a. As in one-step rollout, post-decision rollout selects an action that maximizes the sum of the current-period reward plus the estimated reward-to-go.

The pre-decision rollout policy shown in Figure 2 is in the spirit of a rolling horizon procedure and selects an action by executing a restocking fixed-route heuristic from the current state s_k and using the resulting policy to select an action. To guarantee weak improvement over a set of benchmark policies, we fortify each of our rollout policies (see Bertsekas et al. 1997 and Goodson et al.

2013 for a discussion).

In the a priori-based rollout procedures of Goodson et al. (2013), the restricted policy class does not include some actions that preemptively replenish vehicle capacities. Consequently, the a priori-based pre-decision rollout procedure of Goodson et al. (2013) is unable to explicitly consider preemptive capacity replenishment actions. To overcome this drawback, Goodson et al. (2013) combine pre- and post-decision rollout to form a hybrid rollout policy that explicitly considers restocking. In this work, we relax the a priori fixed-route policy class of Goodson et al. (2013) via a restocking fixed-route policy class. Because this class of policies includes actions that preemptively replenish vehicle capacities, a hybrid rollout policy is unnecessary. For instance, consider action 1 associated with the example in Table 1 and Figure 1 of Goodson et al. (2013). Goodson et al. (2013) show action 1, which preemptively replenishes the capacity of all vehicles, cannot be selected by an a priori fixed-route policy. Setting the restricted policy class to restocking fixed-route policies makes it possible for heuristics to explicitly consider such preemptive capacity replenishment actions.

Our computational experience indicates our restocking-based one-step and post-decision rollout policies become computationally prohibitive to implement for moderately-sized problem instances (e.g., 50 customers). In these cases, pre-decision rollout is an attractive alternative. We also consider the static decomposition scheme of Fan et al. (2006) and the dynamic decomposition scheme of Goodson et al. (2013) as additional methods to overcome computational roadblocks. These decomposition schemes reduce computation by applying rollout policies to single-vehicle problems and using the resulting policies to select actions for a multi-vehicle fleet. Both schemes restrict the action space over which a rollout policy is executed by disallowing actions that assign vehicles to customers outside the single-vehicle problems. The static decomposition partitions customers into permanent groups at the beginning of the time horizon, whereas dynamic decomposition re-partitions customers at each decision epoch via a single execution of the restocking fixed-route heuristic. Both decomposition schemes lead to computational tractability for larger problem instances. We refer to Goodson et al. (2013) for a more in-depth explanation of the decomposition schemes.

6 Computational Results

In this section, we report on computational experiments that implement the restocking-based VRPSDL rollout policies described in §5. Our experiments examine how three factors impact the quality of the rollout policies. First, we investigate how the sophistication of the heuristic search mechanism and the quality of the initial policy influence rollout policy quality. Then, by comparing our restocking-based rollout policies to the a priori-based rollout policies of Goodson et al. (2013), we explore the effect of relaxing the restricted policy class.

In §6.1, we detail the generation of problem instances used in our experiments. In §6.2, we describe upper bounds and benchmarks for our rollout policies. A discussion of our computational results is provided in §6.3. We implement our procedures in C++ and execute all computational experiments on 2.8GHz Intel Xeon processors with 12-48GB of RAM and the CentOS 5.3 operating system (we do not utilize parallel processing). The total computing time required to carry out the experiments is 2.94 CPU years.

6.1 Problem Instances

To facilitate a comparison with the results of Goodson et al. (2013), we conduct experiments on the same problem instances. We summarize the problem instances here and refer the reader to Goodson et al. (2013) for a more detailed discussion. Goodson et al. (2013) modify eight problems derived from the instances of Solomon (1987), ignoring the time windows. The problems include R101 (randomly dispersed customers) and C101 (clustered customers), each with 25, 50, 75, and 100 customers. They vary vehicle capacity (*small*, *medium*, and *large*), impose route duration limits (*short*, *medium*, and *long*), and vary customer demand variability (*low*, *moderate*, and *high*) to yield a total of 216 problem instances. In Appendix B, we detail these values along with the number of vehicles in each instance. For each instance, Goodson et al. (2013) randomly generate 500 realizations according to symmetric, unimodal probability distributions for customer demand (a total of 108,000 realizations). We use these same realizations in the computational experiments presented in this paper.

6.2 Bounds and Benchmarks

As an upper bound on the value of an optimal policy, we sum the expected demand at all customers. We label this bound "Expected Demand." This is a valid upper bound for any policy because it is impossible to serve more demand than the total available demand. For several problem instances with short duration limits, it is impossible to reach some customers within the route duration limit (i.e., via a singleton route from the depot, to the customer, and back to the depot). When calculating the upper bound for these problem instances, we do not include the expected demand of impossible-to-reach customers.

We consider three benchmarks for our rollout policies. First, we use the value of the high-quality (static) a priori fixed-route policies obtained by Goodson et al. (2013), which do not permit preemptive capacity replenishment. We label this benchmark "A Priori Fixed Routes." Second, we use the value of high-quality (static) restocking fixed-route policies within the class of policies described in §4. We label this benchmark "Restocking Fixed Routes." Because fixed-route policies are commonly implemented in industry, these benchmark policies serve as a practice-based standard for our rollout policies. The electronic companion of Goodson et al. (2013) describes the simulated annealing procedure employed to search the space of fixed-route policies, both a priori and restocking. As a third benchmark, we compare the values of our restocking-based rollout policies to the values of the best a priori-based rollout policies obtained by Goodson et al. (2013). We label this benchmark "A Priori-Based Rollout." This benchmark allows us to gauge any improvement due to the use of the restocking fixed-route heuristic to explicitly consider preemptive capacity replenishment.

6.3 Results and Discussion

Table 2 provides aggregate results of computational experiments exploring the effect of initial policy choice and heuristic search mechanism on rollout policy quality. We consider the three heuristics described in §5.1: *Random*, *Local Search*, and *Variable Neighborhood Search*. We consider two initial policies: the benchmark policies described in §6.2 (*High Quality*) and a randomly selected policy (*Low Quality*).

Each portion of Table 2 displays, for a different combination of heuristic and initial policy, average demand served and the average number of CPU seconds required to select an action at each decision epoch. These results are aggregated over all problem parameters for 25-, 50-, 75-, and 100-customer problem instances. Thus, each entry in Table 2 represents results for 54 of the 216 problem instances. Aggregating results in this fashion provides a concise overview of the computational experiments. In Appendix B, we provide disaggregated results for each method. The disaggregated results indicate consistent performance across the problem characteristics we consider.

Tables 3 and 4 display aggregate results by heuristic and by initial policy, respectively, showing percent improvement over the benchmark restocking fixed routes. In these tables, the choice of benchmark is arbitrary. Comparison to a priori fixed routes or to a rollout policy would yield similar insights.

We glean three insights from Tables 2, 3, and 4. First, beginning with a high quality initial policy is better than beginning with a low quality initial policy. From Table 4, a low quality initial policy leads to average improvement over restocking fixed routes of -9.5 percent. When the initial policy is of high quality, the figure increases to 0.64 percent. Novoa and Storer (2009) reach a similar conclusion for a single-vehicle routing problem with stochastic customer demand.

Second, regardless of the quality of the initial policy, both local search and variable neighborhood search heuristics perform better than the random heuristic. From Table 3, the random heuristic posts average improvement over restocking fixed routes of -10.98 percent, whereas the local search and variable neighborhood heuristics perform much better with percent improvements of -0.88 and -1.05, respectively. Notably, however, when beginning with a high quality initial policy, Table 2 shows even the random heuristic improves over the benchmark restocking fixed routes. This observation adds to the body of literature suggesting fortified rollout is a powerful tool to yield high quality dynamic policies.

Third, as demonstrated in Table 3, the local search heuristic is comparable to the variable neighborhood search heuristic. In separate computational tests, even with different random num-

Table 2: Aggregate Results: Heuristics and Initial Policies

	25 Customers		50 Customers		75 Customers		100 Customers	
Method	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**
Random Heuristic with Low Quality Initial Policy								
One-Step Rollout	279.08	1.45	–	–	–	–	–	–
Post-Decision Rollout	275.04	0.20	–	–	–	–	–	–
Pre-Decision Rollout	232.13	0.02	397.10	0.05	712.96	0.13	886.48	0.31
Random Heuristic with High Quality Initial Policy								
One-Step Rollout	318.88	0.86	–	–	–	–	–	–
Post-Decision Rollout	319.04	0.24	–	–	–	–	–	–
Pre-Decision Rollout	317.98	0.02	623.89	0.12	1054.91	0.27	1385.31	0.52
Local Search Heuristic with Low Quality Initial Policy								
One-Step Rollout	309.89	138.42	–	–	–	–	–	–
Post-Decision Rollout	310.12	27.66	–	–	–	–	–	–
Pre-Decision Rollout	302.15	0.47	592.75	4.84	1023.76	26.87	1339.79	79.84
Local Search Heuristic with High Quality Initial Policy								
One-Step Rollout	321.78	0.12	–	–	–	–	–	–
Post-Decision Rollout	321.73	0.02	–	–	–	–	–	–
Pre-Decision Rollout	320.81	0.00	630.01	0.01	1066.05	0.02	1402.99	0.05
Variable Neighborhood Search Heuristic with Low Quality Initial Policy								
One-Step Rollout	309.57	924.16	–	–	–	–	–	–
Post-Decision Rollout	308.94	179.07	–	–	–	–	–	–
Pre-Decision Rollout	304.91	2.29	593.21	25.96	1020.43	152.51	1332.22	480.31
Variable Neighborhood Search Heuristic with High Quality Initial Policy								
One-Step Rollout	321.66	241.47	–	–	–	–	–	–
Post-Decision Rollout	321.11	49.51	–	–	–	–	–	–
Pre-Decision Rollout	320.34	1.43	628.36	17.61	1062.35	108.95	1396.82	355.49

Table 3: Percent Improvement Over Restocking Fixed Routes by Heuristic

Method	25 Customers	50 Customers	75 Customers	100 Customers	Average
		Random Heuristic			
One-Step Rollout	-6.26	–	–	–	-6.26
Post-Decision Rollout	-6.96	–	–	–	-6.96
Pre-Decision Rollout	-15.51	-22.17	-19.30	-21.92	-19.72
		Local Search Heuristic			
One-Step Rollout	-0.59	–	–	–	-0.59
Post-Decision Rollout	-0.56	–	–	–	-0.56
Pre-Decision Rollout	-2.00	-2.01	-0.92	-0.98	-1.48
		Variable Neighborhood Search Heuristic			
One-Step Rollout	-0.66	–	–	–	-0.66
Post-Decision Rollout	-0.85	–	–	–	-0.85
Pre-Decision Rollout	-1.63	-2.11	-1.26	-1.49	-1.62

Table 4: Percent Improvement Over Restocking Fixed Routes by Quality of Initial Policy

Method	25 Customers	50 Customers	75 Customers	100 Customers	Average
		Low Quality Initial Policy			
One-Step Rollout	-6.08	–	–	–	-6.08
Post-Decision Rollout	-6.60	–	–	–	-6.60
Pre-Decision Rollout	-13.58	-18.18	-14.74	-16.75	-15.81
		High Quality Initial Policy			
One-Step Rollout	0.96	–	–	–	0.96
Post-Decision Rollout	0.91	–	–	–	0.91
Pre-Decision Rollout	0.63	0.60	0.62	0.73	0.64

ber streams, 95 percent of the time variable neighborhood search yields restocking fixed-route policies at least as good as the policies returned by the local search heuristic. Despite the ability of the variable neighborhood search heuristic to potentially identify better restocking fixed-route policies than the local search heuristic, these higher quality heuristic policies do not to lead to improved dynamic solutions. These results suggest the model error associated with a restricted policy class may lead to the phenomenon that finding a better policy within the restricted class does not result in a better policy for the original problem. Thus, for the VRPSDL, efforts to identify better dynamic solutions are unlikely to benefit from more sophisticated heuristic search mechanisms for restocking fixed-route policies. Instead, we believe a more promising avenue is to focus on restricted policy classes that more accurately approximate future rewards, e.g., policy classes permitting cooperation among vehicles.

Having identified a high-quality initial policy and a local search heuristic as important ingredients for a rollout policy, Table 5 provides aggregate results of computational experiments comparing our restocking-based rollout policies to the a priori-based rollout policies of Goodson et al. (2013). Rollout policies in these experiments begin with the high quality initial fixed-route policies described in §6.2 and employ the local search heuristic of §5.1. Table 5 is structured similar to Table 2 – results are aggregated over all problem parameters for 25-, 50-, 75-, and 100-customer problem instances. In Appendix B, we provide disaggregated results for each method. The disaggregated results indicate consistent performance across the problem characteristics we consider.

The first portion of Table 5 displays the benchmarks and bounds described in §6.2. The remaining portions of Table 5 display results for our rollout policies without decomposition, with static decomposition, and with dynamic decomposition. The average demand served values reported for "A Priori Fixed Routes" are averages of the true expected demand served (using the analytical evaluation method of Goodson et al. 2013) across 54 problem instances. The average demand served values reported for "Expected Demand" are the average of the expected demands across 54 problem instances. Average demand served values for the remaining methods are averages of estimates of the expected demand served for each of 500 realizations in each of 54 problem instances. We note that we list the per epoch computation time for "Restocking Fixed Routes" as 0.00 for all

Table 5: Aggregate Results: Restocking- vs. A Priori-Based Rollout

	25 Customers		**50 Customers**		**75 Customers**		**100 Customers**	
Method	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**	**Avg. Demand Served**	**Avg. Per Epoch CPU**
			Benchmarks and Bounds					
A Priori Fixed Routes	308.87	N/A	609.19	N/A	1038.89	N/A	1362.01	N/A
Restocking Fixed Routes	317.71	0.00	623.64	0.00	1054.55	0.00	1384.87	0.00
A Priori-Based Rollout	319.89	0.57	627.67	42.37	1060.83	0.59	1395.34	3.31
Expected Demand	372.67	N/A	752.17	N/A	1209.50	N/A	1610.67	N/A
			No Decomposition					
One-Step Rollout	321.78	0.12	–	–	–	–	–	–
Post-Decision Rollout	321.73	0.02	–	–	–	–	–	–
Pre-Decision Rollout	320.81	0.00	630.01	0.01	1066.05	0.02	1402.99	0.05
			Static Decomposition					
One-Step Rollout	318.63	0.00	625.07	0.01	1056.19	0.02	1387.89	0.02
Post-Decision Rollout	318.65	0.00	625.14	0.00	1056.32	0.00	1388.02	0.00
Pre-Decision Rollout	317.71	0.00	623.64	0.00	1054.55	0.00	1384.87	0.00
			Dynamic Decomposition					
One-Step Rollout	320.80	0.00	630.00	0.01	1066.14	0.04	1403.09	0.07
Post-Decision Rollout	320.87	0.00	630.01	0.01	1066.20	0.02	1403.16	0.05
Pre-Decision Rollout	320.80	0.00	629.96	0.01	1066.06	0.02	1403.05	0.05

Table 6: Percent Improvement Over Restocking Fixed Routes

Method	25 Customers	50 Customers	75 Customers	100 Customers	Average
		No Decomposition			
One-Step Rollout	1.28	–	–	–	1.28
Post-Decision Rollout	1.27	–	–	–	1.27
Pre-Decision Rollout	0.98	1.02	1.09	1.31	1.10
		Dynamic Decomposition			
One-Step Rollout	0.97	1.02	1.10	1.32	1.10
Post-Decision Rollout	1.00	1.02	1.10	1.32	1.11
Pre-Decision Rollout	0.97	1.01	1.09	1.31	1.10

cases. This time indicates some computation may be required at an epoch, notably the evaluation of equations (7) and (8). However, in our experiments, the time was insignificant.

Although the average per epoch computation times reported in Table 5 indicate shorter CPU times for restocking-based rollout than for a priori-based rollout, these times are not directly comparable due to differences in code architecture that handicap the a priori-based rollout implementation of Goodson et al. (2013). We also note the restocking-based one-step and post-decision rollout policies are computationally prohibitive to implement for larger problems because the size of the auxiliary dynamic programs required to evaluate restocking policies grows exponentially with problem size. Thus, making real-time dynamic routing decision in large-scale operations using our framework can only be accomplished via pre-decision rollout or via rollout with decomposition. In the latter case, dynamic decomposition yields better results than static decomposition.

Tables 6, 7, and 8 compare the performance of one-step, post-decision, and pre-decision rollout, with dynamic decomposition and without decomposition, to methods from the literature. Specifically, we compare our rollout policies using a restocking heuristic to restocking fixed routes in Table 6, to a priori-based rollout in Table 7, and to restocking-based pre-decision rollout without decomposition in Table 8. We omit a comparison to rollout with static decomposition because rollout with dynamic decomposition yields superior performance. Each entry in these tables is the percent improvement over the corresponding method from the literature. As in Table 5, the values in Tables 6, 7, and 8 are aggregated over all problem parameters for 25-, 50-, 75-, and 100-customer problems.

The results in Table 6 demonstrate that a rollout policy, resulting from re-application of the re-

Table 7: Percent Improvement Over A Priori-Based Rollout

Method	25 Customers	50 Customers	75 Customers	100 Customers	Average
		No Decomposition			
One-Step Rollout	0.59	–	–	–	0.59
Post-Decision Rollout	0.58	–	–	–	0.58
Pre-Decision Rollout	0.29	0.37	0.49	0.55	0.42
		Dynamic Decomposition			
One-Step Rollout	0.28	0.37	0.50	0.56	0.43
Post-Decision Rollout	0.31	0.37	0.51	0.56	0.44
Pre-Decision Rollout	0.29	0.36	0.49	0.55	0.42

Table 8: Percent Improvement Over Pre-Decision Rollout Without Decomposition

Method	25 Customers	50 Customers	75 Customers	100 Customers	Average
		No Decomposition			
One-Step Rollout	0.30	–	–	–	0.30
Post-Decision Rollout	0.29	–	–	–	0.29
		Dynamic Decomposition			
One-Step Rollout	0.00	0.00	0.01	0.01	0.00
Post-Decision Rollout	0.02	0.00	0.01	0.01	0.01
Pre-Decision Rollout	0.00	-0.01	0.00	0.00	0.00

stocking heuristic at each state, improves on the performance of the statically-implemented benchmark restocking fixed routes. On average, the rollout policies improve on the restocking fixed routes by 1.27 percent.

Table 7 demonstrates the improved performance of restocking-based rollout over a priori-based rollout policies. The best performance is achieved by one-step and post-decision rollout policies without decomposition. On average, these two methods improve upon the performance of a priori-based rollout by 0.58 percent. For problem instances with more than 25 customers, restocking-based pre-decision rollout is a computationally attractive alternative, on average improving upon the performance of a priori-based rollout by 0.42 percent. Further, restocking-based pre-decision rollout yields superior performance over all the a priori-based rollout procedures considered by Goodson et al. (2013) on problem instances with as many as 100 customers. Restocking-based one-step and post-decision rollout with dynamic decomposition offer comparable performance.

While demonstrating the ability of restocking-based rollout to outperform a-priori-based rollout, the results offer an additional conclusion. As noted previously, an optimal restocking policy along a fixed route is guaranteed to weakly improve upon the performance of an a priori policy along the same fixed route. Applied in a rollout procedure, this means a restocking policy offers a better estimate of future rewards than the a priori policy along the same fixed route. Although there is no guarantee this improved estimate leads to better action selection, Table 7 suggests better estimates matter, at least for the VRPSDL. Because our restocking heuristic employs the same local search heuristic of Goodson et al. (2013), improvements over a priori-based rollout are due to the relaxation of the restricted policy class permitting preemptive capacity replenishment.

Table 8 demonstrates the lookahead mechanism of one-step and post-decision rollout is beneficial, improving on the pre-decision restocking-based rollout by 0.3 and 0.29 percent, respectively. When using the restocking heuristic to estimate the reward-to-go, however, one-step and post-decision rollout are limited to problems with at most 25 customers. Regardless, these results demonstrate that applying the restocking heuristic in a rollout algorithm improves the performance of the heuristic. In contrast to the results for a priori-based rollout in Goodson et al. (2013), Table 8 indicates dynamic decomposition combined with restocking-based rollout policies does not yield significant performance improvement over the corresponding pre-decision rollout policy. This result suggests the improved estimates of the restocking heuristic are more important than the decomposition's ability to facilitate post-decision and one-step lookahead.

We conclude this section with a brief discussion of the statistical significance of our results. Table 9 presents six 90-percent paired confidence intervals. The first set of three intervals compare restocking-based rollout with a priori-based rollout and the second set of three intervals make comparisons among restocking-based rollout methods. In each case, the confidence interval is on the mean difference of the demand served by the first method less the demand served by the second method. All problem parameters are considered in aggregate and rollout policies begin with a high quality initial policy and utilize the local search heuristic. Because the first three confidence intervals exclude zero, we conclude relaxation of the a priori fixed-route policy class to the restocking fixed-route policy class results in a statistically significant improvement. Intervals four through six suggest rollout based on the restocking fixed-route policy class benefits from rollout decision rules that look further ahead. In particular, both restocking-based one-step and post-decision rollout yield statistically significant improvement over restocking-based pre-decision rollout. Because zero is included at the edge of the fourth interval, we conclude at the 90 percent confidence level that restocking-based one-step and post-decision rollout yield comparable performance.

Table 9: Confidence Intervals on Rollout Performance

Rollout Method Comparison	90-Percent Paired Confidence Interval
One-Step Restocking vs. One-Step A Priori	[1.80, 1.99]
Post-Decision Restocking vs. Post-Decision A Priori	[2.17, 2.36]
Pre-Decision Restocking vs. Pre-Decision A Priori	[6.61, 6.79]
One-Step Restocking vs. Post-Decision Restocking	[0.00, 0.11]
One-Step Restocking vs. Pre-Decision Restocking	[0.91, 1.04]
Post-Decision Restocking vs. Pre-Decision Restocking	[0.86, 0.99]

7 Conclusion

We develop restocking-based rollout policies to make dynamic routing decisions for the vehicle routing problem with stochastic demand and duration limits. Our contributions center around improving the heuristic component of the rollout policies developed by Goodson et al. (2013). To achieve this, we relax the a priori fixed-route policy class of Goodson et al. (2013) by considering restocking fixed-route policies that permit preemptive capacity replenishment. We develop a sampling-based procedure to estimate the value of a restocking fixed-route policy. Embedding this procedure within rollout policies, we demonstrate benefits of restocking-based rollout policies versus applying a restocking fixed route policy generated at the beginning of the horizon. We also show improvement over the results of Goodson et al. (2013). These results demonstrate that, for the VRPSDL, there is value in improving the estimate of the reward-to-go when implementing a rollout algorithm. We also identify investigation of additional restricted policy classes as a more promising future research direction than seeking to improve the heuristic search component of the rollout policy. Because of the nature of the heuristic estimates in this paper, the improvement results could only be demonstrated computationally. An interesting direction for future research would be to consider the circumstances in which an improved estimate of future rewards guarantees improved rollout policies.

References

American Trucking Association (2009). Allowance allocation policies in climate legislation: assisting consumers, investing in a clean energy future, and adapting to climate change. Statement made before the House Committee on Energy and Commerce Subcommittee on Energy and Environment. `http://www.truckline.com/AdvIssues/Energy/Cap%20and%20Trade%20Documents/Testimony%20on%20Allowance%20Allocation%20Policies%20in%20Climate%20Legislation.pdf`, Accessed on August 15, 2011.

Bertsekas, D. (2000). *Dynamic programming and optimal control* (2nd ed.), Volume I. Belmont, MA: Athena Scientific.

Bertsekas, D., J. Tsitsiklis, and C. Wu (1997). Rollout algorithms for combinatorial optimization. *Journal of Heuristics 3*(3), 245–262.

Bertsimas, D., P. Chervi, and M. Peterson (1995). Computational approaches to stochastic vehicle routing problems. *Transportation Science 29*(4), 342–352.

Bertsimas, D. and D. Simchi-Levi (1996). A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research 44*(2), 286–304.

Bianchi, L., M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-doria, and T. Schiavinotto (2006). Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms 5*, 91–110.

Brown, D., J. Smith, and P. Sun (2010). Information relaxations and duality in stochastic dynamic programs. *Operations Research 58*(4), 785–801.

Campbell, A. and B. Thomas (2008). Challenges and advances in a priori routing. In B. Golden, S. Raghavan, and E. Wasil (Eds.), *The vehicle routing problem: latest advances and new challenges*. New York, NY: Springer.

Denardo, E. (2003). *Dynamic programming: models and applications*. Mineola, NY: Dover Publications.

Erera, A., M. Savelsbergh, and E. Uyar (2009). Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks 54*(4), 270–283.

Fan, J., X. Wang, and H. Ning (2006). A multiple vehicles routing problem algorithm with stochastic demand. In *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Volume 1, pp. 1688–1692.

Goodson, J., J. Ohlmann, and B. Thomas (2012). Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. *European Journal of Operational Research 217*(2), 312–323.

Goodson, J., J. W. Ohlmann, and B. Thomas (2013). Rollout policies for dynamic solutions to the multi-vehicle routing problem with stochastic demand and duration limits. *Operations Research 61*(1), 138–154.

Goodson, J., B. W. Thomas, and J. W. Ohlmann (2013). A generalized rollout policy framework for stochastic dynamic programming. Working paper, Saint Louis University, `http://www.slu.edu/~goodson/papers/GoodsonRolloutFramework.pdf`, Accessed on January 13, 2013.

Hansen, P. and N. Mladenović (2001). Variable neighborhood search: principles and applications. *European Journal of Operational Research 130*, 449–467.

Novoa, C. and R. Storer (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research 196*(2), 509–515.

Pandelis, D., C. Kyriakidis, and T. Dimitrakos (2012). Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research 217*, 324–332.

Powell, W. (2007). *Approximate Dynamic Programming*. New York, NY: John Wiley and Sons.

Secomandi, N. (2001). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research 49*(5), 796–802.

Secomandi, N. (2003). Analysis of a rollout approach to sequencing problems with stochastic routing applications. *Journal of Heuristics 9*(4), 321–352.

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

Tatarakis, A. and I. Minis (2009). Stochastic vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research 197*, 557–571.

Tsirimpas, P., A. Tatarakis, I. Minis, and E. Kyriakidis (2008). Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research 187*, 483–495.

Wilson, R. (2011). 22nd annual state of logistics report. Council of Supply Chain Management Professionals.

Yang, W., K. Mathur, and R. Ballou (2000). Stochastic vehicle routing problem with restocking. *Transportation Science 34*(1), 99–112.

Acknowledgements

The authors wish to thank two anonymous referees and an Associate Editor for their useful comments. The authors also thank Information Technology Services at the University of Iowa and at Saint Louis University for access to high performance computing clusters.

A Proofs

A.1 Proof of Proposition 1

We first prove property (i). Since $Z_{c'}(\tilde{s}_{c'}) < x_{v^m_{c'}}$, we have one of two cases. In the first case, $Z_{c'}(\tilde{s}_{c'}) = 0$, implying by equation (5) that $A_{v^m_{c'}} + t(v^m_{c'}, 0) > L$. By the triangle inequality, this implies that for any $A_{v^m_{c'+1}}$ that may result from equations (3) or (4), $A_{v^m_{c'+1}} + t(v^m_{c'+1}, 0) > L$. Thus, $Z_{c'+1}(\tilde{s}_{c'+1}) = 0$. The same argument applies for $c = c' + 2, \dots, C^m$. In the second case, $0 < Z_{c'}(\tilde{s}_{c'}) < x_{v^m_{c'}}$, implying by equation (5) that $\lfloor (L - t(v^m_{c'}, 0) - A_{v^m_{c'}})/(t(v^m_{c'}, 0) + t(0, v^m_{c'})) \rfloor < \lfloor (x_{v^m_{c'}} - Q_{v^m_{c'}})/Q \rfloor$, meaning the number of replenishments required to satisfy demand in full is greater than the number of replenishments possible before violating the route duration limit L. Let t' be the time at which demand at customer $v^m_{c'}$ is served in full. It must be that $t' + t(v^m_{c'}, 0) > L$. Then, by the triangle inequality, $t' + t(v^m_{c'}, v^m_{c'+1}) + t(v^m_{c'+1}, 0) > L$ and $t' + t(v^m_{c'}, 0) + t(0, v^m_{c'+1}) + t(v^m_{c'+1}, 0) > L$, meaning that $A_{v^m_{c'+1}} + t(v^m_{c'+1}, 0) > L$ regardless of the action selected at decision epoch c'. Thus, $Z_{c'+1}(\tilde{s}_{c'+1}) = 0$. Then, using the arguments presented for the first case, $Z_c(\tilde{s}_c) = 0$ for $c = c' + 2, \dots, C^m$.

We prove property (ii) by contradiction. Suppose there exists some $c \in \{1, 2, \dots, c' - 1\}$ such that $Z_c(\tilde{s}_c) < x_{v^m_c}$. Then, by property (i), $Z_j(\tilde{s}_j) = 0$ for $j = c + 1, c + 2, \dots, C^m$. Yet, by assumption, $Z_j(\tilde{s}_j) = x_{v^m_j}$ for $j = 1, 2, \dots, c' - 1$.

A.2 Proof of Proposition 2

The proof is by induction. First, note that if $\tilde{V}_{C^m}(\tilde{s}_{C^m}; v^m, x) < x_{v^m_{C^m}}$, then $\tilde{V}_c(\tilde{s}_{C^m}; v^m, x)$ increases as $Q_{v^m_{C^m}}$ increases and as $A_{v^m_{C^m}}$ decreases. If $\tilde{V}_{C^m}(\tilde{s}_{C^m}; v^m, x) = x_{v^m_{C^m}}$, then $\tilde{V}_{C^m}(\tilde{s}_{C^m})$ is constant as it is bounded above by $x_{v^m_{C^m}}$. Thus, the result holds for stage C^m. Assume the result holds for stages $C^m - 1, C^m - 2, \dots, c + 1$. At stage c, it follows from equation (5) that $Z_c(\tilde{s}_c)$ increases as $Q_{v^m_c}$ increases and as $A_{v^m_c}$ decreases. As in stage C^m, $Z_c(\tilde{s}_c)$ is constant if $Z_c(\tilde{s}_c) = x_{v^m_c}$. By the induction hypothesis, the reward-to-go, $\tilde{V}_{c+1}(\tilde{s}_{c+1}; v^m, x)$, also increases as $Q_{v^m_c}$ increases and as $A_{v^m_c}$ decreases. Because the value function at stage c is the sum of two functions that increase as $Q_{v^m_c}$ increases and as $A_{v^m_c}$ decreases, the result holds at stage c.

B Disaggregated Results of Computational Results

In this appendix, we report the disaggregated results of our computational experiments. Table 10 details problem instance parameters. Table 11 displays the expected demand served by the benchmark restocking fixed-route policies. Table 12 shows the expected demand for each problem instance; expected demand does not change with variability in customer demand. Tables 13, 14, and 15 depict disaggregated results for our rollout policies without decomposition. Tables 16, 17, and 18 present disaggregated results for our rollout policies with static decomposition. Tables 19, 20, and 21 display disaggregated results for our rollout policies with dynamic decomposition. Table 22 is similar to Table 5, except Table 22 displays the average number of CPU seconds required to execute each method for an entire instance; Table 5 only presents the average number of CPU seconds per decision epoch. Tables 23-37 display disaggregate results for our second set of computational experiments with various heuristic mechanisms and initial policies. In Table 11, Tables 13-21, and Tables 23-37, each entry is the average demand served by the respective method across 500 realizations of the problem instance. The values in Table 12 are exact calculations.

Table 10: Problem Parameters

Problem	Vehicles	Duration Limits (short, medium, long)	Capacities (small, medium, large)
R101 (25)	4	(85.575, 142.625, 199.675)	(25, 50, 75)
C101 (25)	5	(67.875, 113.125, 158.375)	(25, 50, 75)
R101 (50)	8	(89.475, 149.125, 208.775)	(25, 50, 75)
C101 (50)	9	(67.875, 113.125, 158.375)	(25, 50, 75)
R101 (75)	11	(103.05, 171.75, 240.45)	(25, 50, 75)
C101 (75)	14	(99.45, 165.75, 232.05)	(25, 50, 75)
R101 (100)	15	(94.875, 158.125, 221.375)	(25, 50, 75)
C101 (100)	19	(95.325, 158.875, 222.425)	(25, 50, 75)

Table 11: Average Demand Served by Restocking Fixed Routes

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.15	221.93	278.66	236.90	325.38	332.01	308.01	331.98	332.04
C101 (25)	222.21	298.74	320.02	317.07	454.56	459.44	391.09	459.16	459.64
R101 (50)	354.92	501.20	632.47	520.38	716.59	721.04	661.69	720.78	721.15
C101 (50)	336.73	541.30	606.40	543.51	801.22	859.28	676.47	858.92	859.87
R101 (75)	593.76	831.77	1010.24	837.70	1077.39	1078.20	1034.26	1076.78	1078.23
C101 (75)	756.57	1097.16	1294.36	1067.83	1357.99	1359.55	1299.86	1359.45	1360.05
R101 (100)	803.42	1146.33	1376.46	1140.98	1456.67	1458.09	1383.68	1455.94	1458.47
C101 (100)	918.79	1349.53	1648.26	1304.85	1801.67	1810.42	1629.44	1808.90	1810.35
				Moderate Variability					
R101 (25)	147.51	215.87	273.03	223.22	312.90	330.76	287.98	331.09	331.61
C101 (25)	207.58	292.59	319.06	300.68	425.49	457.50	372.25	455.67	459.01
R101 (50)	329.95	486.24	610.00	494.30	690.85	719.14	619.45	718.78	720.22
C101 (50)	322.04	505.71	581.77	507.68	764.40	846.26	644.36	850.36	856.92
R101 (75)	558.08	791.92	962.29	793.13	1059.74	1075.92	975.12	1074.56	1076.54
C101 (75)	711.85	1046.57	1252.74	1015.24	1334.17	1356.72	1228.61	1357.86	1359.02
R101 (100)	755.10	1094.05	1326.28	1078.36	1435.19	1455.73	1313.87	1454.19	1455.65
C101 (100)	842.98	1292.65	1580.31	1228.41	1748.60	1805.63	1539.07	1804.63	1808.80
				High Variability					
R101 (25)	140.81	202.59	261.95	208.50	291.96	328.16	270.53	328.45	331.53
C101 (25)	184.90	272.76	306.58	280.31	395.29	448.86	347.76	447.75	458.80
R101 (50)	309.47	458.77	571.84	461.48	651.86	712.95	581.60	712.59	716.59
C101 (50)	303.09	474.69	559.07	469.85	694.75	818.23	602.67	820.53	854.34
R101 (75)	508.81	752.50	902.11	747.06	1024.07	1070.85	923.77	1070.62	1075.16
C101 (75)	656.07	978.99	1178.08	951.21	1290.25	1355.54	1144.80	1355.09	1359.74
R101 (100)	697.32	1020.06	1240.32	1006.87	1379.15	1449.18	1235.95	1447.70	1450.07
C101 (100)	778.98	1207.66	1481.80	1149.89	1634.19	1792.21	1439.30	1786.81	1803.95

Table 12: Expected Demand

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
R101 (25)	332.00	332.00	332.00	332.00	332.00	332.00	332.00	332.00	332.00
C101 (25)	320.00	320.00	320.00	460.00	460.00	460.00	460.00	460.00	460.00
R101 (50)	721.00	721.00	721.00	721.00	721.00	721.00	721.00	721.00	721.00
C101 (50)	630.00	630.00	630.00	860.00	860.00	860.00	860.00	860.00	860.00
R101 (75)	1079.00	1079.00	1079.00	1079.00	1079.00	1079.00	1079.00	1079.00	1079.00
C101 (75)	1300.00	1300.00	1300.00	1360.00	1360.00	1360.00	1360.00	1360.00	1360.00
R101 (100)	1438.00	1438.00	1438.00	1458.00	1458.00	1458.00	1458.00	1458.00	1458.00
C101 (100)	1690.00	1690.00	1690.00	1810.00	1810.00	1810.00	1810.00	1810.00	1810.00

Table 13: Average Demand Served by One-Step Rollout without Decomposition

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.26	221.97	278.66	238.59	324.85	332.00	313.31	331.97	331.68
C101 (25)	223.95	299.10	320.06	317.41	455.14	459.41	391.97	459.38	459.63
				Moderate Variability					
R101 (25)	150.00	217.63	273.27	228.73	315.50	331.00	297.36	331.36	331.51
C101 (25)	217.11	293.65	319.50	308.03	434.09	458.03	383.97	456.92	458.96
				High Variability					
R101 (25)	144.78	209.63	262.09	216.37	298.84	329.59	282.83	331.35	331.56
C101 (25)	206.19	281.06	312.30	299.57	406.90	454.30	374.02	453.56	459.47

Table 14: Average Demand Served by Post-Decision Rollout without Decomposition

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.15	221.98	278.66	238.43	326.13	331.96	313.25	331.99	331.99
C101 (25)	223.95	298.81	320.06	317.55	455.22	459.53	391.99	459.16	459.51
				Moderate Variability					
R101 (25)	150.05	217.25	273.25	228.64	315.76	331.25	297.70	331.46	331.44
C101 (25)	217.42	293.51	319.60	309.09	434.16	457.82	383.98	456.40	459.03
				High Variability					
R101 (25)	144.24	209.61	262.10	216.28	298.55	329.57	282.73	331.26	331.60
C101 (25)	204.12	280.42	313.28	298.80	406.95	453.89	372.92	453.72	459.44

Table 15: Average Demand Served by Pre-Decision Rollout without Decomposition

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.26	221.93	278.66	237.90	325.84	332.01	311.81	332.02	332.04
C101 (25)	223.96	299.12	320.02	317.33	455.06	459.64	391.46	459.20	459.64
R101 (50)	360.36	501.27	633.27	523.93	716.84	721.05	665.88	721.12	721.15
C101 (50)	339.03	541.44	606.72	545.14	801.94	859.51	679.49	859.30	859.90
R101 (75)	595.10	833.97	1011.36	841.58	1077.78	1078.36	1038.53	1078.33	1078.40
C101 (75)	760.68	1099.12	1296.29	1073.93	1359.58	1359.95	1302.97	1360.04	1360.05
R101 (100)	809.01	1147.16	1378.49	1151.03	1457.79	1458.51	1394.95	1458.51	1458.60
C101 (100)	923.41	1351.02	1653.20	1307.92	1803.54	1810.80	1641.42	1810.73	1810.84
				Moderate Variability					
R101 (25)	149.56	217.19	273.16	226.34	314.28	331.14	295.52	331.55	331.61
C101 (25)	216.41	293.81	319.37	306.84	432.93	457.81	379.98	457.14	459.23
R101 (50)	334.09	488.27	611.12	502.21	695.38	720.29	636.21	720.39	720.48
C101 (50)	325.95	511.14	582.92	526.61	769.22	849.40	661.69	854.21	858.51
R101 (75)	566.23	794.36	966.11	808.58	1067.28	1077.22	997.48	1076.89	1077.49
C101 (75)	733.76	1057.77	1263.94	1040.89	1343.56	1357.70	1262.63	1360.01	1359.30
R101 (100)	772.73	1104.63	1329.75	1104.23	1443.13	1457.21	1348.52	1457.17	1457.35
C101 (100)	874.44	1306.45	1602.74	1275.16	1772.34	1808.53	1588.98	1809.20	1809.90
				High Variability					
R101 (25)	142.26	204.10	262.11	215.13	297.57	329.66	280.96	331.29	331.70
C101 (25)	197.18	280.90	310.75	293.78	402.11	454.25	369.23	453.41	459.42
R101 (50)	318.30	465.63	577.69	482.19	662.14	715.88	611.38	715.40	716.93
C101 (50)	314.26	481.58	567.60	497.25	714.24	823.78	637.02	838.74	854.95
R101 (75)	527.42	760.55	912.62	775.51	1043.07	1075.38	967.65	1075.26	1075.63
C101 (75)	698.65	1014.63	1194.86	1004.76	1323.92	1359.87	1219.95	1358.94	1360.58
R101 (100)	726.77	1043.24	1255.35	1055.71	1410.60	1452.19	1308.35	1452.35	1452.86
C101 (100)	823.89	1240.74	1519.21	1228.86	1690.16	1800.91	1540.75	1803.63	1806.43

Table 16: Average Demand Served by One-Step Rollout with Static Decomposition

Duration		short			medium			long	
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	156.15	221.92	278.66	237.48	324.61	332.02	312.44	332.00	332.04
C101 (25)	223.21	299.03	320.02	317.15	454.97	459.62	391.12	459.16	458.63
R101 (50)	361.06	501.20	632.47	524.79	716.61	720.82	665.96	720.61	721.12
C101 (50)	337.89	541.34	606.40	544.09	801.50	859.40	677.36	858.79	859.79
R101 (75)	595.27	833.98	1010.95	841.91	1077.74	1078.25	1038.58	1078.05	1078.32
C101 (75)	759.02	1099.32	1293.81	1069.68	1359.04	1359.75	1300.23	1359.84	1359.39
R101 (100)	808.56	1146.77	1376.57	1151.22	1457.41	1458.44	1390.46	1457.07	1457.70
C101 (100)	921.66	1351.45	1648.26	1305.83	1802.51	1810.70	1633.05	1808.60	1810.75
				Moderate Variability					
R101 (25)	149.14	216.45	273.03	225.74	313.84	331.36	291.35	331.38	331.57
C101 (25)	213.35	293.13	319.30	301.30	428.45	457.56	372.77	455.80	458.79
R101 (50)	330.76	487.71	610.45	497.77	692.82	719.85	627.05	719.11	720.21
C101 (50)	322.62	506.19	581.84	512.61	765.82	846.61	650.12	850.88	858.04
R101 (75)	559.28	793.63	964.07	796.98	1056.98	1076.67	979.45	1075.99	1077.09
C101 (75)	717.12	1050.76	1255.85	1018.50	1333.84	1357.71	1230.97	1358.42	1359.18
R101 (100)	762.49	1101.67	1327.59	1083.46	1436.94	1454.81	1322.92	1455.44	1456.10
C101 (100)	848.78	1296.81	1585.45	1236.61	1749.06	1806.95	1542.51	1806.46	1808.22
				High Variability					
R101 (25)	141.72	202.59	261.95	211.48	295.84	329.49	273.91	330.99	331.70
C101 (25)	184.92	273.26	306.92	282.41	396.96	449.91	350.63	447.88	458.88
R101 (50)	311.61	460.42	573.24	467.87	652.49	713.76	587.35	713.44	716.50
C101 (50)	303.60	474.69	559.41	472.80	695.75	818.90	605.78	820.74	854.02
R101 (75)	511.15	755.01	902.86	750.05	1025.99	1074.12	927.64	1073.57	1075.21
C101 (75)	659.67	980.17	1177.51	952.99	1292.50	1357.29	1147.05	1356.11	1359.84
R101 (100)	703.45	1024.66	1245.35	1013.06	1381.31	1449.82	1242.80	1449.52	1451.27
C101 (100)	782.55	1212.71	1482.56	1155.57	1637.88	1794.12	1445.07	1789.60	1805.35

Table 17: Average Demand Served by Post-Decision Rollout with Static Decomposition

Duration		short			medium			long	
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	156.15	221.93	278.66	237.48	325.70	332.02	312.44	331.94	332.04
C101 (25)	223.21	299.03	320.02	317.15	454.98	459.58	391.12	459.12	458.57
R101 (50)	361.06	501.20	632.47	524.79	716.61	720.82	665.96	720.99	720.62
C101 (50)	337.89	541.34	606.40	544.09	801.38	859.50	677.37	859.02	859.76
R101 (75)	595.27	833.98	1010.95	842.56	1077.66	1078.29	1038.79	1078.15	1078.27
C101 (75)	759.02	1099.32	1293.78	1069.94	1359.09	1359.98	1300.37	1358.44	1360.03
R101 (100)	808.56	1146.77	1376.60	1151.18	1457.37	1458.16	1390.30	1457.50	1458.49
C101 (100)	921.66	1351.45	1648.07	1305.73	1802.29	1810.66	1633.37	1808.98	1810.75
				Moderate Variability					
R101 (25)	149.14	216.46	273.03	225.74	313.84	331.36	291.35	331.29	331.61
C101 (25)	213.35	293.13	319.30	301.99	428.41	457.56	372.73	456.17	458.83
R101 (50)	330.80	487.55	610.44	498.23	692.82	720.00	627.06	719.60	720.09
C101 (50)	322.62	506.20	581.84	512.55	766.04	846.51	650.26	851.05	858.06
R101 (75)	559.30	793.62	963.81	797.00	1060.46	1076.91	979.57	1075.92	1077.17
C101 (75)	716.97	1050.79	1256.26	1018.58	1334.31	1357.44	1231.02	1358.52	1358.87
R101 (100)	762.46	1101.73	1327.62	1083.40	1437.01	1454.85	1322.90	1456.24	1456.87
C101 (100)	848.76	1297.51	1585.52	1236.66	1749.13	1806.78	1543.16	1805.93	1808.94
				High Variability					
R101 (25)	141.61	202.59	261.95	211.53	295.83	329.48	274.35	330.83	331.65
C101 (25)	184.92	273.26	306.66	282.58	396.98	449.91	350.04	447.94	458.71
R101 (50)	311.61	460.05	573.25	468.02	652.28	713.81	587.48	713.93	716.66
C101 (50)	303.61	474.69	559.41	473.06	695.80	818.85	605.93	821.91	854.47
R101 (75)	511.88	755.04	902.75	750.08	1026.37	1073.19	927.87	1074.06	1075.40
C101 (75)	660.28	980.08	1177.96	952.96	1292.65	1357.15	1147.23	1355.91	1359.87
R101 (100)	703.36	1024.52	1245.39	1013.38	1381.14	1450.04	1242.27	1449.50	1452.09
C101 (100)	782.91	1213.56	1482.57	1155.47	1638.17	1794.57	1445.17	1789.79	1805.59

Table 18: Average Demand Served by Pre-Decision Rollout with Static Decomposition

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.15	221.93	278.66	236.90	325.38	332.01	308.01	331.98	332.04
C101 (25)	222.21	298.74	320.02	317.07	454.56	459.44	391.09	459.16	459.64
R101 (50)	354.92	501.20	632.47	520.38	716.59	721.04	661.69	720.78	721.15
C101 (50)	336.73	541.30	606.40	543.51	801.22	859.28	676.47	858.92	859.87
R101 (75)	593.76	831.77	1010.24	837.70	1077.39	1078.20	1034.26	1076.78	1078.23
C101 (75)	756.57	1097.16	1294.36	1067.83	1357.99	1359.55	1299.86	1359.45	1360.05
R101 (100)	803.42	1146.33	1376.46	1140.98	1456.67	1458.09	1383.69	1455.94	1458.47
C101 (100)	918.79	1349.53	1648.26	1304.85	1801.67	1810.42	1629.44	1808.90	1810.35
				Moderate Variability					
R101 (25)	147.51	215.87	273.03	223.22	312.90	330.76	287.98	331.09	331.61
C101 (25)	207.58	292.59	319.06	300.68	425.49	457.50	372.25	455.67	459.01
R101 (50)	329.95	486.24	610.00	494.30	690.85	719.14	619.45	718.78	720.22
C101 (50)	322.04	505.71	581.77	507.68	764.40	846.26	644.36	850.36	856.92
R101 (75)	558.08	791.92	962.29	793.13	1059.74	1075.92	975.12	1074.56	1076.54
C101 (75)	711.85	1046.57	1252.74	1015.24	1334.17	1356.72	1228.61	1357.86	1359.02
R101 (100)	755.10	1094.05	1326.28	1078.36	1435.19	1455.73	1313.87	1454.19	1455.65
C101 (100)	842.98	1292.65	1580.31	1228.41	1748.60	1805.63	1539.07	1804.63	1808.80
				High Variability					
R101 (25)	140.81	202.59	261.95	208.50	291.96	328.16	270.53	328.45	331.53
C101 (25)	184.90	272.76	306.58	280.31	395.29	448.86	347.76	447.75	458.80
R101 (50)	309.47	458.77	571.84	461.48	651.86	712.95	581.60	712.59	716.59
C101 (50)	303.09	474.69	559.07	469.85	694.75	818.23	602.67	820.53	854.34
R101 (75)	508.81	752.50	902.11	747.06	1024.07	1070.85	923.77	1070.62	1075.17
C101 (75)	656.07	978.99	1178.08	951.21	1290.25	1355.54	1144.80	1355.09	1359.74
R101 (100)	697.32	1020.06	1240.32	1006.87	1379.15	1449.18	1235.95	1447.70	1450.07
C101 (100)	778.98	1207.66	1481.80	1149.89	1634.19	1792.21	1439.30	1786.81	1803.95

Table 19: Average Demand Served by One-Step Rollout with Dynamic Decomposition

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.26	221.93	278.66	237.91	324.65	332.00	312.06	332.01	332.04
C101 (25)	223.96	299.13	320.02	317.34	455.06	459.39	391.48	459.22	459.64
R101 (50)	360.62	501.27	633.27	524.58	717.17	720.82	665.42	720.64	721.09
C101 (50)	338.90	541.54	606.70	545.29	802.21	859.47	679.27	859.24	859.88
R101 (75)	595.93	833.95	1011.37	842.86	1077.53	1078.36	1038.96	1078.29	1078.41
C101 (75)	760.24	1099.10	1296.55	1073.61	1358.71	1359.95	1302.98	1360.04	1359.87
R101 (100)	809.26	1146.99	1378.23	1152.07	1457.97	1458.49	1395.44	1458.42	1458.47
C101 (100)	924.38	1351.11	1652.68	1308.30	1802.98	1810.80	1641.20	1809.85	1810.84
				Moderate Variability					
R101 (25)	149.49	217.08	273.16	226.86	314.36	331.17	295.05	331.52	331.57
C101 (25)	215.94	293.74	319.37	306.74	433.03	457.86	380.40	457.17	459.14
R101 (50)	334.04	488.17	611.01	502.73	695.75	720.12	637.41	720.33	720.45
C101 (50)	326.30	510.72	583.03	526.61	769.64	849.31	662.60	854.34	858.48
R101 (75)	566.63	794.68	966.15	808.57	1066.94	1077.20	998.59	1076.89	1077.51
C101 (75)	733.12	1057.90	1262.07	1041.38	1343.45	1358.43	1262.33	1359.95	1359.28
R101 (100)	773.95	1104.09	1329.71	1104.91	1443.21	1457.12	1350.17	1456.78	1457.26
C101 (100)	874.35	1307.04	1604.17	1276.08	1771.71	1808.70	1588.60	1809.21	1809.32
				High Variability					
R101 (25)	142.77	204.63	262.10	215.22	297.58	329.67	280.81	331.21	331.71
C101 (25)	197.51	279.81	310.26	294.43	402.09	454.42	370.05	453.42	459.13
R101 (50)	318.58	465.76	577.74	481.49	661.06	715.64	611.88	715.34	716.78
C101 (50)	314.75	480.92	566.90	497.17	714.13	823.67	636.53	838.27	854.87
R101 (75)	527.53	760.43	913.53	776.31	1043.61	1075.30	967.97	1075.28	1075.60
C101 (75)	698.54	1013.12	1196.75	1005.66	1323.96	1359.76	1220.91	1358.80	1360.57
R101 (100)	728.58	1043.41	1261.03	1053.92	1410.86	1451.80	1309.60	1452.25	1452.65
C101 (100)	822.45	1239.57	1520.27	1227.91	1688.28	1800.60	1540.32	1803.11	1806.46

Table 20: Average Demand Served by Post-Decision Rollout with Dynamic Decomposition

Duration		short			medium			long	
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.26	221.93	278.66	237.91	325.84	332.00	311.63	331.99	332.04
C101 (25)	223.96	299.13	320.02	317.38	455.06	459.63	391.46	459.18	459.64
R101 (50)	359.02	501.27	633.27	524.66	717.17	720.83	665.88	721.07	721.15
C101 (50)	339.09	541.50	606.69	545.36	801.51	859.55	679.14	859.33	859.92
R101 (75)	595.84	833.94	1011.38	842.77	1077.51	1078.40	1039.19	1078.35	1078.39
C101 (75)	760.51	1099.63	1296.00	1073.95	1359.68	1359.98	1303.15	1360.04	1360.05
R101 (100)	809.53	1146.83	1378.29	1151.75	1457.83	1458.52	1394.86	1458.23	1458.60
C101 (100)	923.33	1351.62	1652.40	1308.04	1803.19	1810.77	1641.75	1810.08	1810.84
				Moderate Variability					
R101 (25)	149.61	217.27	273.13	226.70	314.25	331.26	295.36	331.50	331.61
C101 (25)	216.23	293.76	319.37	307.00	433.40	457.77	380.83	456.67	459.10
R101 (50)	334.25	487.99	611.33	503.13	695.85	720.29	636.56	720.38	720.39
C101 (50)	326.39	510.74	583.04	526.77	769.51	849.54	661.61	854.37	858.52
R101 (75)	566.48	794.78	965.82	809.52	1067.65	1077.28	998.05	1076.97	1077.52
C101 (75)	733.84	1057.77	1262.30	1040.91	1343.26	1357.55	1263.19	1359.97	1359.30
R101 (100)	774.60	1104.69	1329.85	1105.01	1442.80	1457.00	1349.28	1457.16	1457.33
C101 (100)	874.77	1307.01	1603.55	1276.48	1771.09	1808.20	1589.01	1808.83	1809.89
				High Variability					
R101 (25)	142.74	204.30	262.06	214.86	297.55	329.60	280.97	331.21	331.71
C101 (25)	197.82	281.34	310.66	293.88	402.64	454.85	369.72	453.37	459.39
R101 (50)	318.41	465.72	577.62	482.15	661.29	715.81	611.32	715.35	716.91
C101 (50)	314.61	481.46	567.79	497.38	713.01	824.45	636.96	838.04	854.95
R101 (75)	528.21	760.66	913.60	776.42	1043.18	1075.35	968.30	1075.28	1075.59
C101 (75)	698.57	1013.16	1196.04	1005.96	1323.39	1359.85	1220.65	1358.81	1360.58
R101 (100)	727.21	1043.20	1258.31	1056.54	1412.06	1451.90	1309.02	1452.23	1452.83
C101 (100)	823.56	1239.84	1518.80	1230.54	1689.25	1800.98	1541.30	1803.36	1806.46

Table 21: Average Demand Served by Pre-Decision Rollout with Dynamic Decomposition

Duration		short			medium			long	
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.26	221.93	278.66	237.90	325.84	332.01	311.75	332.02	332.04
C101 (25)	223.96	299.13	320.02	317.46	455.06	459.63	391.47	459.22	459.64
R101 (50)	359.21	501.27	633.27	524.09	716.86	721.05	665.75	721.08	721.15
C101 (50)	339.47	541.44	606.69	545.18	801.94	859.51	679.33	859.31	859.90
R101 (75)	595.51	833.92	1011.44	841.94	1077.82	1078.35	1038.42	1078.28	1078.41
C101 (75)	760.65	1099.16	1296.11	1073.93	1359.56	1359.95	1302.97	1360.04	1360.05
R101 (100)	809.33	1147.04	1378.25	1151.18	1457.89	1458.50	1395.29	1458.53	1458.60
C101 (100)	923.63	1351.49	1653.11	1307.86	1803.39	1810.77	1642.03	1810.73	1810.84
				Moderate Variability					
R101 (25)	149.39	216.77	273.16	226.17	314.29	331.39	295.31	331.54	331.61
C101 (25)	216.38	293.81	319.35	306.87	433.79	457.76	379.54	457.04	459.23
R101 (50)	333.92	488.15	611.06	502.65	696.01	720.36	635.98	720.39	720.47
C101 (50)	326.27	511.21	582.85	526.21	769.39	849.37	661.36	854.33	858.52
R101 (75)	566.21	794.35	966.37	808.44	1067.50	1077.27	997.67	1076.78	1077.49
C101 (75)	733.26	1058.98	1263.53	1040.80	1343.30	1357.73	1263.66	1360.02	1359.30
R101 (100)	773.23	1104.25	1329.93	1103.67	1443.20	1457.20	1348.84	1457.18	1457.34
C101 (100)	873.76	1306.69	1602.97	1275.28	1772.29	1808.49	1589.20	1809.13	1809.89
				High Variability					
R101 (25)	142.18	204.14	262.06	214.97	297.53	329.69	280.67	331.30	331.70
C101 (25)	197.57	281.16	310.74	294.00	401.96	454.70	368.89	453.32	459.39
R101 (50)	318.66	465.91	577.92	481.72	661.64	715.80	610.90	715.50	716.93
C101 (50)	314.21	480.98	567.51	497.66	714.06	824.24	636.36	837.93	854.92
R101 (75)	527.92	761.19	912.93	775.31	1042.71	1075.15	967.25	1075.28	1075.63
C101 (75)	698.64	1014.01	1194.19	1004.33	1324.46	1359.92	1219.75	1358.74	1360.58
R101 (100)	727.80	1043.31	1256.91	1053.96	1411.22	1452.23	1308.12	1452.36	1452.86
C101 (100)	824.09	1240.64	1519.84	1228.25	1690.97	1800.92	1539.82	1803.69	1806.45

Table 22: Average CPU Seconds Per Instance

Method	25 Customers Avg. CPU	50 Customers Avg. CPU	75 Customers Avg. CPU	100 Customers Avg. CPU
		No Decomposition		
One-Step Rollout	42.70	–	–	–
Post-Decision Rollout	9.14	–	–	–
Pre-Decision Rollout	0.45	4.39	26.02	76.35
		Static Decomposition		
One-Step Rollout	0.85	5.29	18.42	30.82
Post-Decision Rollout	0.25	0.89	3.52	6.11
Pre-Decision Rollout	0.01	0.02	0.02	0.14
		Dynamic Decomposition		
One-Step Rollout	1.27	9.60	43.49	104.81
Post-Decision Rollout	0.53	5.29	29.70	81.75
Pre-Decision Rollout	0.41	4.94	27.94	80.53

Table 23: Average Demand Served by One-Step Rollout with Random Heuristic and High Quality Initial Policy

Duration		short			medium			long	
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	156.15	221.93	278.66	236.94	325.38	332.02	309.90	332.02	332.02
C101 (25)	222.73	298.98	320.02	317.07	454.86	459.62	391.13	459.17	459.37
				Moderate Variability					
R101 (25)	147.80	216.29	273.14	223.96	312.88	331.02	290.00	331.42	331.56
C101 (25)	210.10	292.63	319.24	301.72	429.32	457.58	375.99	457.67	459.03
				High Variability					
R101 (25)	141.69	202.92	262.09	210.27	293.16	328.92	274.64	330.58	331.40
C101 (25)	185.77	273.41	309.12	283.66	399.24	452.93	360.26	450.94	459.37

Table 24: Average Demand Served by Post-Decision Rollout with Random Heuristic and High Quality Initial Policy

Duration		short			medium			long	
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	156.15	221.93	278.66	237.22	325.39	332.01	309.94	332.01	332.04
C101 (25)	222.44	299.07	320.02	317.10	454.86	459.64	391.11	459.20	459.64
				Moderate Variability					
R101 (25)	147.83	216.34	273.18	223.85	313.10	331.14	290.11	331.43	331.61
C101 (25)	211.77	292.83	319.23	302.32	431.09	457.73	376.07	457.78	459.21
				High Variability					
R101 (25)	141.38	202.91	262.10	210.45	293.35	329.05	274.80	330.48	331.72
C101 (25)	186.01	274.23	308.90	284.19	399.34	453.02	360.41	451.21	459.57

Table 25: Average Demand Served by Pre-Decision Rollout with Random Heuristic and High Quality Initial Policy

Duration	short			medium			long		
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	156.15	221.93	278.66	236.91	325.40	332.01	310.46	332.00	332.04
C101 (25)	222.50	298.87	320.02	317.07	454.80	459.58	391.09	459.16	459.64
R101 (50)	355.49	501.20	632.47	520.38	716.59	721.04	662.16	720.93	721.15
C101 (50)	336.78	541.30	606.40	543.53	801.36	859.40	677.00	858.98	859.90
R101 (75)	593.76	831.77	1010.37	837.72	1077.56	1078.20	1034.31	1077.53	1078.27
C101 (75)	756.57	1097.16	1294.44	1067.95	1358.25	1359.85	1300.04	1359.80	1360.05
R101 (100)	804.14	1146.33	1376.46	1141.00	1456.94	1458.22	1383.87	1456.98	1458.53
C101 (100)	918.81	1349.53	1648.26	1304.86	1802.20	1810.47	1630.34	1809.80	1810.61
				Moderate Variability					
R101 (25)	147.75	216.06	273.03	223.32	312.92	330.94	288.49	331.43	331.61
C101 (25)	208.42	292.59	319.06	300.68	426.83	457.56	372.45	456.19	459.01
R101 (50)	329.98	486.24	610.11	494.38	690.87	719.51	620.26	719.44	720.33
C101 (50)	322.04	505.94	581.77	508.04	764.84	846.47	644.71	850.86	857.48
R101 (75)	558.12	791.91	962.35	793.13	1059.87	1076.42	975.88	1075.38	1077.00
C101 (75)	712.04	1046.65	1254.17	1015.28	1334.29	1357.07	1229.46	1358.65	1359.29
R101 (100)	755.12	1094.05	1326.28	1078.45	1435.32	1456.24	1314.21	1455.65	1456.35
C101 (100)	842.98	1293.06	1580.33	1229.09	1748.83	1806.67	1539.47	1805.71	1809.33
				High Variability					
R101 (25)	140.97	202.59	261.95	208.99	292.06	328.26	271.25	329.60	331.66
C101 (25)	184.85	272.76	306.81	280.73	396.49	449.63	348.90	447.74	458.80
R101 (50)	310.13	458.79	571.85	461.85	652.44	713.25	582.86	713.09	716.71
C101 (50)	303.07	474.69	559.07	470.09	694.69	818.60	603.12	821.78	854.49
R101 (75)	508.81	752.55	902.19	747.24	1024.29	1071.62	924.26	1072.75	1075.40
C101 (75)	656.74	979.14	1178.18	951.59	1291.56	1356.21	1145.70	1355.89	1360.20
R101 (100)	697.34	1020.06	1241.22	1007.47	1379.15	1449.76	1236.36	1449.04	1451.56
C101 (100)	778.99	1207.71	1481.94	1149.98	1635.31	1792.85	1440.16	1788.52	1804.77

Table 26: Average Demand Served by One-Step Rollout with Random Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	**small**	**medium**	**large**	**small**	**medium**	**large**	**small**	**medium**	**large**
				Low Variability					
R101 (25)	139.09	168.58	169.42	190.73	244.85	289.98	260.18	326.97	326.58
C101 (25)	189.90	247.86	262.84	276.43	373.01	417.69	354.41	456.08	459.62
				Moderate Variability					
R101 (25)	112.63	135.55	152.97	186.11	246.35	270.99	251.35	311.70	316.40
C101 (25)	154.81	243.66	276.90	266.50	379.16	431.17	340.90	445.31	458.88
				High Variability					
R101 (25)	100.95	170.81	179.24	188.72	245.36	257.88	241.03	287.83	323.12
C101 (25)	158.74	232.32	280.97	246.18	347.93	426.73	343.58	443.89	459.32

Table 27: Average Demand Served by Post-Decision Rollout with Random Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	126.12	137.59	193.97	188.98	253.47	278.00	255.24	295.16	318.62
C101 (25)	185.84	240.37	289.01	271.56	369.29	410.34	336.86	449.41	459.62
				Moderate Variability					
R101 (25)	91.18	127.39	148.32	192.46	245.93	246.68	259.27	310.94	323.72
C101 (25)	163.31	253.83	290.97	265.91	342.53	422.83	340.15	454.18	459.03
				High Variability					
R101 (25)	101.85	166.06	166.89	183.73	239.83	220.32	236.24	298.86	314.73
C101 (25)	159.38	236.36	263.47	249.39	356.71	417.08	340.84	444.31	457.93

Table 28: Average Demand Served by Pre-Decision Rollout with Random Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	132.49	77.98	104.23	163.12	216.48	224.66	216.99	272.08	245.45
C101 (25)	137.04	218.49	199.25	231.91	286.66	324.77	326.13	394.40	443.70
R101 (50)	187.98	216.38	265.55	360.56	389.52	491.52	445.27	546.92	623.13
C101 (50)	212.07	181.01	284.95	375.63	467.39	467.38	495.42	691.06	603.31
R101 (75)	308.03	480.77	435.88	473.39	695.16	685.40	685.02	924.14	969.37
C101 (75)	435.31	549.71	624.06	747.64	946.05	1044.58	892.97	1117.04	1160.22
R101 (100)	337.06	471.57	586.29	752.75	882.84	999.21	954.10	1190.77	1166.73
C101 (100)	474.52	726.02	805.69	824.94	1032.82	1138.58	1204.42	1453.58	1504.07
				Moderate Variability					
R101 (25)	81.07	74.75	111.93	166.40	220.70	179.95	233.24	284.11	297.09
C101 (25)	135.26	201.33	220.90	212.64	333.83	354.06	286.33	415.91	451.18
R101 (50)	171.99	236.66	203.52	342.53	427.08	479.16	461.59	571.25	591.48
C101 (50)	219.97	249.54	293.23	298.28	417.91	470.09	478.70	643.59	678.04
R101 (75)	268.20	372.73	395.50	567.14	657.78	653.89	744.66	826.53	905.17
C101 (75)	437.69	511.63	595.78	710.68	854.67	1005.56	862.55	1120.36	1261.56
R101 (100)	394.49	525.11	510.02	713.52	828.37	914.20	999.79	1113.42	1217.49
C101 (100)	523.86	617.50	726.19	804.05	1016.49	1054.15	1119.80	1338.11	1542.41
				High Variability					
R101 (25)	79.08	129.83	139.10	167.03	197.55	182.61	214.58	207.26	293.00
C101 (25)	110.70	170.16	209.18	207.85	296.98	358.54	276.35	385.09	433.66
R101 (50)	202.57	184.22	280.36	275.26	373.05	396.64	437.18	538.64	584.31
C101 (50)	162.30	244.79	279.43	332.22	437.33	464.68	464.95	586.77	658.86
R101 (75)	299.76	397.22	376.09	478.67	663.89	707.77	706.95	843.65	926.92
C101 (75)	357.11	570.48	595.11	641.05	951.79	902.88	881.27	1069.45	1203.04
R101 (100)	382.19	491.48	525.51	634.02	870.38	919.92	829.12	1075.07	1222.25
C101 (100)	444.63	646.63	682.38	795.93	906.18	1117.25	1080.73	1351.25	1430.15

Table 29: Average Demand Served by One-Step Rollout with Local Search Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	143.27	196.03	223.80	234.87	308.25	330.77	300.39	331.97	332.01
C101 (25)	205.03	298.21	299.93	314.54	434.60	459.46	385.14	458.64	459.63
				Moderate Variability					
R101 (25)	118.54	182.24	204.84	222.41	303.68	325.51	292.93	330.84	331.53
C101 (25)	177.14	280.73	300.48	303.80	422.86	457.47	374.57	458.70	458.83
				High Variability					
R101 (25)	107.49	195.91	231.09	215.13	292.32	312.10	276.51	329.93	331.67
C101 (25)	175.81	268.90	312.72	281.40	403.86	451.71	371.73	452.08	459.98

Table 30: Average Demand Served by Post-Decision Rollout with Local Search Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	142.79	194.18	229.62	236.85	310.32	331.39	304.89	331.79	331.99
C101 (25)	200.83	298.86	300.00	315.45	443.74	459.46	380.81	458.98	459.63
				Moderate Variability					
R101 (25)	119.71	183.46	209.10	225.48	303.95	324.44	292.20	331.11	331.45
C101 (25)	178.39	276.65	300.91	305.62	421.20	457.77	376.11	458.81	458.93
				High Variability					
R101 (25)	108.86	195.70	230.28	213.80	292.57	312.01	274.58	330.00	331.57
C101 (25)	175.06	266.28	311.26	281.27	402.46	451.32	370.07	452.58	459.97

Table 31: Average Demand Served by Pre-Decision Rollout with Local Search Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	139.12	196.21	209.85	216.80	284.09	316.94	295.26	331.98	332.01
C101 (25)	195.26	268.85	270.05	297.50	416.17	459.29	377.46	458.31	459.62
R101 (50)	293.22	408.92	542.69	495.48	681.97	694.70	623.17	720.31	720.87
C101 (50)	299.39	487.07	588.08	505.89	709.19	775.02	646.59	855.10	859.34
R101 (75)	521.87	728.68	864.26	777.35	1061.76	1059.13	964.02	1078.04	1074.09
C101 (75)	677.68	1011.09	1188.52	992.54	1343.31	1359.41	1252.69	1359.07	1360.05
R101 (100)	653.48	1005.22	1246.72	1049.08	1425.87	1456.45	1313.18	1458.41	1458.57
C101 (100)	754.43	1199.32	1452.48	1195.40	1691.50	1787.66	1537.69	1810.41	1810.77
				Moderate Variability					
R101 (25)	116.14	174.35	176.09	220.92	280.23	297.39	285.97	329.67	331.35
C101 (25)	172.50	279.19	297.26	298.10	414.40	455.48	368.39	455.81	458.81
R101 (50)	291.15	428.66	533.24	465.39	626.67	696.80	612.62	716.07	720.28
C101 (50)	309.06	443.90	537.97	482.63	698.81	789.58	638.66	844.36	858.20
R101 (75)	452.27	718.36	856.91	780.16	1033.13	1058.90	966.84	1076.18	1077.50
C101 (75)	660.71	967.78	1173.75	974.42	1316.61	1347.14	1225.73	1356.42	1360.04
R101 (100)	671.45	993.96	1246.24	1053.50	1376.74	1449.12	1301.85	1455.54	1457.21
C101 (100)	766.32	1217.22	1436.74	1215.59	1667.20	1800.42	1529.31	1803.82	1809.18
				High Variability					
R101 (25)	109.61	195.41	219.14	213.12	277.15	302.10	272.15	326.04	331.40
C101 (25)	169.01	264.90	300.51	276.92	397.80	447.47	366.01	450.30	459.99
R101 (50)	290.51	393.84	488.00	453.88	616.21	673.51	599.37	708.32	716.74
C101 (50)	247.51	444.18	529.85	486.26	673.23	780.83	623.86	829.67	851.62
R101 (75)	480.08	697.26	850.07	744.92	986.39	1065.57	944.51	1072.58	1075.62
C101 (75)	645.31	970.04	1141.26	965.10	1304.16	1356.17	1190.92	1356.27	1360.54
R101 (100)	628.55	959.50	1165.60	1015.45	1357.53	1437.42	1270.79	1445.89	1452.81
C101 (100)	775.46	1206.02	1400.92	1176.40	1628.64	1778.19	1495.75	1790.91	1804.53

Table 32: Average Demand Served by One-Step Rollout with Variable Neighborhood Search Heuristic and High Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.27	221.97	278.66	238.15	324.66	331.99	312.70	331.97	331.99
C101 (25)	224.13	299.07	320.02	317.52	455.22	459.42	391.72	459.21	459.62
				Moderate Variability					
R101 (25)	149.92	217.48	273.20	228.68	315.03	331.34	297.57	331.32	331.53
C101 (25)	217.35	293.66	319.47	307.64	434.16	458.03	385.10	456.57	458.93
				High Variability					
R101 (25)	144.75	205.71	262.16	217.58	296.90	329.50	282.51	330.81	331.54
C101 (25)	203.84	280.57	312.41	299.76	409.96	454.26	373.35	453.30	459.36

Table 33: Average Demand Served by Post-Decision Rollout with Variable Neighborhood Search Heuristic and High Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.16	221.94	278.66	237.84	325.73	332.02	311.79	331.87	331.97
C101 (25)	224.04	299.14	320.04	317.30	454.94	458.84	391.78	459.29	459.51
				Moderate Variability					
R101 (25)	149.73	216.76	273.18	228.16	314.45	331.34	295.96	330.90	331.44
C101 (25)	216.74	293.29	319.38	306.73	434.42	457.48	381.97	456.44	459.00
				High Variability					
R101 (25)	144.33	203.65	261.99	214.80	297.19	329.34	281.04	330.78	331.64
C101 (25)	203.67	280.13	311.35	297.58	406.11	453.88	369.91	453.12	459.35

Table 34: Average Demand Served by Pre-Decision Rollout with Variable Neighborhood Search Heuristic and High Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	156.27	221.93	278.66	237.32	325.40	332.01	310.68	332.02	332.04
C101 (25)	223.59	298.97	320.02	317.08	454.80	458.89	391.21	459.24	459.64
R101 (50)	359.04	501.21	632.52	523.75	716.77	721.05	663.49	721.06	721.15
C101 (50)	339.34	541.36	606.52	543.89	801.46	859.47	677.41	859.17	859.91
R101 (75)	595.01	832.99	1010.60	839.79	1077.66	1078.26	1035.17	1077.90	1078.40
C101 (75)	759.68	1098.42	1294.70	1070.33	1359.14	1359.92	1301.52	1360.03	1360.05
R101 (100)	808.28	1146.63	1376.81	1144.74	1457.27	1458.44	1387.27	1458.15	1458.60
C101 (100)	923.01	1351.92	1649.03	1306.63	1804.02	1810.66	1633.45	1809.98	1810.83
				Moderate Variability					
R101 (25)	149.45	216.95	273.20	225.92	313.48	331.28	294.11	331.34	331.61
C101 (25)	215.77	293.30	319.31	304.22	431.24	457.79	378.42	457.23	459.23
R101 (50)	332.39	487.05	610.56	499.88	694.03	720.07	628.84	720.11	720.47
C101 (50)	328.08	509.00	582.59	518.13	765.66	847.90	653.64	853.89	858.35
R101 (75)	565.08	792.75	963.92	803.36	1063.61	1076.80	989.36	1076.44	1077.49
C101 (75)	728.57	1050.84	1257.77	1027.16	1341.25	1357.94	1252.72	1359.93	1359.45
R101 (100)	770.09	1099.90	1327.86	1092.41	1439.93	1456.69	1331.08	1456.28	1457.17
C101 (100)	868.88	1301.46	1589.85	1255.32	1763.09	1807.79	1570.31	1808.32	1809.78
				High Variability					
R101 (25)	142.13	203.68	262.10	215.29	294.02	329.18	278.94	330.55	331.70
C101 (25)	198.77	278.74	310.25	291.58	402.61	454.84	368.12	452.82	459.34
R101 (50)	317.86	463.98	575.06	479.49	660.32	714.76	607.18	715.22	716.88
C101 (50)	314.59	477.66	565.33	493.29	706.10	821.04	633.11	835.45	854.82
R101 (75)	522.63	756.31	907.13	772.34	1033.27	1074.47	957.86	1074.78	1075.57
C101 (75)	691.07	1002.23	1183.76	985.39	1316.55	1358.98	1202.00	1357.80	1360.54
R101 (100)	721.00	1034.13	1249.64	1048.12	1393.00	1451.35	1289.19	1451.09	1452.82
C101 (100)	814.20	1230.49	1501.26	1205.63	1669.21	1798.50	1508.56	1802.21	1805.79

Table 35: Average Demand Served by One-Step Rollout with Variable Neighborhood Search Heuristic and Low Quality Initial Policy

Duration		short			medium			long	
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	143.29	193.08	232.77	229.56	298.50	327.35	297.95	332.03	331.99
C101 (25)	204.85	297.77	299.86	313.20	448.34	459.52	385.85	459.37	459.64
				Moderate Variability					
R101 (25)	116.84	179.62	204.13	224.10	299.32	316.42	292.81	331.01	331.49
C101 (25)	179.26	281.77	300.53	304.64	422.95	457.72	378.73	458.78	458.91
				High Variability					
R101 (25)	109.95	197.03	228.82	215.95	291.68	307.10	276.78	329.04	331.65
C101 (25)	176.03	269.49	311.49	281.23	402.27	451.04	371.09	452.06	459.96

Table 36: Average Demand Served by Post-Decision Rollout with Variable Neighborhood Search Heuristic and Low Quality Initial Policy

Duration		short			medium			long	
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	143.23	195.11	227.24	229.61	303.88	327.57	301.56	331.95	331.98
C101 (25)	204.76	297.88	300.00	314.96	437.72	459.49	384.91	459.17	459.62
				Moderate Variability					
R101 (25)	116.69	183.42	208.66	222.20	297.30	319.49	290.64	330.95	331.49
C101 (25)	177.47	284.84	300.50	303.20	420.61	457.71	376.72	458.76	458.84
				High Variability					
R101 (25)	106.23	195.12	226.53	212.95	288.94	307.56	275.31	329.37	331.60
C101 (25)	173.69	264.42	310.38	280.22	399.08	451.10	368.36	451.67	459.99

Table 37: Average Demand Served by Pre-Decision Rollout with Variable Neighborhood Search Heuristic and Low Quality Initial Policy

Duration	short			medium			long		
Capacity	small	medium	large	small	medium	large	small	medium	large
				Low Variability					
R101 (25)	142.89	192.54	218.29	216.35	296.29	316.42	297.35	331.69	332.01
C101 (25)	196.26	289.26	296.97	297.62	423.40	459.05	379.07	459.22	459.64
R101 (50)	305.27	423.39	557.71	482.94	654.65	686.62	620.51	720.64	720.93
C101 (50)	306.17	475.15	571.89	499.10	712.85	826.03	643.64	851.81	859.21
R101 (75)	534.63	738.25	848.68	785.62	1019.21	1058.30	974.35	1078.32	1078.38
C101 (75)	676.88	996.71	1192.65	990.96	1332.91	1354.16	1240.57	1359.44	1360.03
R101 (100)	652.06	1012.47	1182.86	1052.57	1381.58	1441.39	1321.52	1458.15	1458.48
C101 (100)	771.81	1226.84	1487.96	1207.08	1689.89	1765.98	1534.50	1809.44	1810.59
				Moderate Variability					
R101 (25)	117.30	178.21	199.66	219.08	290.72	310.20	287.67	330.22	331.55
C101 (25)	177.25	279.28	300.05	299.20	415.32	456.49	368.74	457.98	459.13
R101 (50)	295.56	443.09	528.59	471.11	636.67	687.83	614.07	715.75	720.40
C101 (50)	309.86	424.41	529.42	481.60	690.08	803.73	640.12	839.44	858.06
R101 (75)	475.10	718.45	845.18	778.16	1002.80	1052.10	959.90	1076.31	1077.42
C101 (75)	664.95	973.10	1179.69	974.43	1293.24	1352.53	1221.42	1358.75	1360.49
R101 (100)	666.45	985.50	1178.08	1044.39	1354.82	1424.41	1296.30	1455.31	1457.20
C101 (100)	780.64	1211.78	1459.91	1206.70	1655.12	1777.29	1514.45	1803.17	1809.24
				High Variability					
R101 (25)	104.65	189.90	224.80	211.34	284.40	300.64	273.16	327.25	331.63
C101 (25)	170.31	258.22	305.61	278.91	397.33	446.45	366.93	451.27	459.98
R101 (50)	292.25	412.37	508.01	452.08	619.24	670.41	595.57	706.27	716.63
C101 (50)	253.47	438.07	528.14	484.76	665.17	789.82	621.36	818.19	853.23
R101 (75)	479.76	704.93	839.25	747.32	977.11	1044.40	940.28	1072.28	1075.53
C101 (75)	648.97	958.43	1135.94	956.01	1283.97	1348.34	1189.68	1356.51	1360.52
R101 (100)	628.13	960.97	1143.07	1015.23	1325.16	1411.81	1258.76	1447.51	1452.73
C101 (100)	775.62	1186.93	1394.33	1166.70	1604.52	1746.24	1482.95	1787.76	1805.74