Dynamic Orienteering on a Network of Queues

Shu Zhang Jeffrey W. Ohlmann Barrett W. Thomas

January 29, 2015

Abstract

We propose a stochastic orienteering problem on a network of queues with time windows at customers. While this problem is of wide applicability, we study it in the context of routing and scheduling a textbook salesperson who visits professors on campus to gain textbook adoptions. The salesperson must determine which professors to visit and how long to wait in queues at each professor. We model the problem as a Markov decision process (MDP) with the objective of maximizing expected sales. We investigate the existence of optimal control limits and examine conditions under which certain actions cannot be optimal. To solve the problem, we propose an approximate dynamic programming approach based on rollout algorithms. The method introduces a two-stage heuristic estimation that we refer to as compound rollout. In the first stage, the algorithm decides whether to stay at the current professor or go to another professor. If departing the current professor, it chooses the professor to whom to go in the second stage. We demonstrate the value of our modeling and solution approaches by comparing the dynamic policies to a-priori-route solutions with recourse actions.

1 Introduction

We introduce a stochastic orienteering problem on a network of queues, where the traveler visits customers at several locations to collect rewards and each customer is available only during a pre-determined time window. After arrival, the traveler may need to wait in a queue to meet the customer. The queue length is unknown before the traveler's arrival and the wait time is uncertain while the traveler is in a queue. This problem is applicable to several operations, such as designing trips for tourists who may experience queueing at attractions (Vansteenwegen and Souffriau (2010)), routing taxis to taxi stands in metropolitan area where there may be queues of competitors (Cheng

and Qu (2009)), and scheduling pharmaceutical sales force to visit doctors and physicians where representatives from competing companies may be present (Zhang et al. (2014)).

In this paper, we focus on the application to textbook sales. The college textbook market consists of about 22 million college students (Hussar and Bailey, 2013), each spending an average of $655 per year on textbooks (Atkins, 2013). Because professors are the key factor driving textbook purchase, publishers employ salespeople to develop relationships with them. A textbook salesperson visits professors on campus multiple times during a semester, especially during the weeks preceding the textbook adoption decision for the following semester. By making professors aware of pedagogical product and service offerings, the salesperson hopes to influence professors' decisions and gain adoptions.

In this study, we consider a daily routing problem in which a textbook salesperson visits professors located in one or more buildings on campus. Associated with each professor is a deterministic reward representing an increase in expected sales volume, which is determined by the size of class taught by the professor and the influence of the salesperson's visit. A meeting between the salesperson and the professor may lead to an increase in the adoption chance, resulting in the increase in expected sales volume. Professors are either accessible during specific time windows (scheduled office hours) or via appointments. According to interviews with our textbook industry partners, approximately 60% to 75% of textbook salesperson's visits occur during office hours (personal communication, Kent Peterson, Vice President of National Sales Region at McGraw-Hill Higher Education, April 6, 2012). In this study, we consider how the salesperson should visit the professors during their office hours and treat appointment scheduling as an exogenous process.

Given a list of professors who have the potential to adopt a product, the textbook salesperson needs to decide which professors to visit and the order in which to visit them to maximize the total expected reward. We assume that the salesperson has lowest priority in the queue. Thus, when seeking to visit a professor, the salesperson must wait at a professor as long as there is a queue of students, regardless of whether the students arrive earlier or later than the salesperson. The wait time is uncertain due to the uncertainty in the time the professor spends with each student and in the arrivals of additional students. Upon arrival, the salesperson needs to decide whether to join the queue and wait or to depart immediately to visit another professor. Deciding to queue at a professor, the salesperson must periodically determine whether to renege and go to another

professor. When choosing who to visit next, the salesperson considers professors who she has not visited and professors who she visited but did not meet. We refer to this problem as the dynamic orienteering problem on a network of queues with time windows (DOPTW).

We model the problem as a Markov decision process (MDP). To reflect the uncertain meeting duration between a professor and students, we model the wait time faced by the salesperson as a random variable. In the MDP, a decision epoch is triggered either by the arrival of the salesperson at a professor, observing queue event(s) (a student arrival and/or departure at the professor), or reaching a specified amount of time with no queueing event. That is, if no queue event has occurred after a specified amount of time, the salesperson "checks her watch" and reconsiders her decision of whether to continue to stay at the current professor or to go to another one. To overcome significant runtime challenges, we develop an approximate dynamic programming approach based on rollout algorithms to solve the problem, in which the value of the decision of whether to stay and where to go is evaluated hierarchically by different approximations. We refer to this approach as *compound rollout*. In compound rollout, decisions are made in two stages. In the first stage, the compound rollout decides whether to stay at the current professor or go to another professor. If the first-stage decision is to go, in the second stage it decides to which professor to go.

This paper makes the following contributions to the literature. First, we introduce a new dynamic orienteering problem motivated by textbook sales, but generalizable to other settings in which a routed entity may experience queues at a series of locations. Second, we identify conditions under which certain actions can be eliminated from the action set of a given state. Third, we investigate the existence of optimal control limits and identify the limited existence of optimal control limit policies. Fourth, we propose a novel compound rollout heuristic to facilitate decision making. The compound rollout is distinct in implementing different mechanisms for reward-to-go approximation based on explicitly partitioning the action space. Finally, our computational results demonstrate the capability of compound rollout in making high quality decisions with reasonable computational efforts by comparing our dynamic policies to benchmark a priori routing solutions.

In §2, we review related literature to position our contribution. In §3, we provide a MDP model for the problem and present the derivation of state transition probabilities and action elimination conditions. We investigate the existence of optimal control limit policies and present the results in the electronic companion. We describe our rollout algorithms in §4 and present computational

results of our rollout methodology in §5. In §6, we conclude the paper and discuss future work.

2 Literature Review

The DOPTW is most related to the study by Zhang et al. (2014) on an orienteering problem with time windows and stochastic wait times at customers. The problem is motivated by the decision-making of pharmaceutical representatives who visit doctors to build brand awareness. They consider uncertain wait times at doctors that results from queues of competing sales representatives. They develop an a priori solution with recourse actions of skipping customers and balking/reneging at customers. Specifically, they propose static rules to determine whether to skip a customer and how long to wait at a customer after arriving. In the DOPTW, a salesperson faces a network of queues as in Zhang et al. (2014), but here we consider dynamic policies that enable the salesperson to decide which customer to go to and whether to stay in queue at a customer based on realized information at each decision epoch. We note that such dynamic policies allow revisiting a customer, an action not possible in an a priori policy.

In the literature, there are other studies focusing on developing a priori solutions for stochastic TSP with profits and time constraints (Teng et al. (2004), Tang and Miller-Hooks (2005), Campbell et al. (2011), Papapanagiotou et al. (2013), and Evers et al. (2014)). While in the literature time windows are not considered and the uncertain arrival time results from the stochastic travel and service times, in the DOPTW, there is a hard time window associated with each customer and the uncertainty in arrival time is induced by the stochastic waiting and service times at previous customers. Furthermore, as far as we are aware, other than the authors' previous work, this study is the only one in which there is queueing at customers.

While literature on the stochastic TSP with profits and time constraints mainly focuses on developing a priori routes, dynamic solutions have been extensively developed for vehicle routing problems (VRPs). Our work is similar to the literature in using approximate dynamic programming (ADP) to solve routing problems, especially in developing heuristic routing policies via rollout procedures (Toriello et al. (2014), Secomandi (2000, 2001), Novoa and Storer (2009), Goodson et al. (2013), and Goodson et al. (2014b)). To approximate the cost-to-go of future states, Secomandi (2000, 2001), Novoa and Storer (2009), and Goodson et al. (2013) use a priori routes as heuristic

policies, while Toriello et al. (2014) apply the approximate linear programming (ALP). Secomandi (2000, 2001) develops a one-step rollout, and Novoa and Storer (2009) develop a two-step rollout algorithm for the single-vehicle routing problem with stochastic demand. Secomandi (2000) compares the value function approximation via a heuristic policy based on a priori routes to a parametric function and concludes that the former generates higher quality solutions. For the multi-vehicle routing problem with stochastic demand and route duration limits, Goodson et al. (2013) develop a family of rollout policies, and Goodson et al. (2014b) develop restocking-based rollout policies that explicitly consider preemptive capacity replenishment. The compound rollout algorithm we propose for the DOPTW involves executing rollout policies as in the above studies. However, our study is distinct in explicitly partitioning the action space and implementing different mechanisms to approximate reward-to-go based on the partition.

Our work is also related to the queueing literature investigating the decision rules in queueing systems. Yechiali (1971) and Yechiali (1972) investigate the optimal balking/joining rules in a G1/M/1 and a G1/M/s system, respectively. They demonstrate that for both systems, a non-randomized control limit optimizing a customer's long-run average reward exists. D'Auria and Kanta (2011) study a network of two tandem queues where customers decide whether or not to join the system upon their arrivals. They investigate the threshold strategies that optimize each customer's net benefit, when the state of the system is fully observable, fully unobservable, or partially observable. Burnetas (2013) examine a system consisting of a series of $M/M/m$ queues, in which a customer receives a reward by completing service at a queue and incurs a delay cost per unit of time in queue. The customer needs to decide whether to balk or to enter a queue, with an objective to maximize the total net benefit. Once balking, the customer leaves the system permanently. Honnappa and Jain (2015) study the arrival process in a network of queues, where travelers choose when to arrive at a parallel queuing network and which queue to join upon arrival. They focus on deriving the equilibrium arrival and routing profiles. Similar to the above studies, we investigate whether or not the salesperson should join the queue when arriving at a customer. However, in the DOPTW, the traveler is routing through a network of queues, where queueing decisions of balking and reneging are made and revisiting queues is allowed. As far as we are aware, there are no studies in literature considering the queueing process in a dynamic routing problem as we do.

3 Problem Statement

We formulate the DOPTW as a Markov decision process (MDP). In §3.1, we present the dynamic program formulation. In §3.2, we present the dynamics of state transition. In §3.3, we investigate conditions under which actions that are guaranteed to not be optimal can be eliminated.

3.1 Problem Formulation

We define the problem on a complete graph $G = (V, E)$. The set V of n nodes corresponds to n potential professors that a textbook salesperson may visit. We note that while the DOPTW is generalizable to other problem settings of orienteering on a network of queues, in this discussion we use "professor" instead of "customer" as the study is motivated by routing a textbook salesperson to visit professors on campus. The set E consists of edges associated with each pair of vertices. We assume a deterministic travel time on edge (i, j), denoted as c_{ij}. We set $c_{ij} = 0$ for $i = j$. A time window $[e_i, l_i]$ is associated with each professor $i \in V$. We use r_i to represent the expected value gained by a meeting with customer i. In this study, we assume the salesperson departs as early as necessary to arrive at the first professor in her tour before the time window begins. Let Ψ_i be the random variable representing the salesperson's arrival time at professor i and ψ_i be a realization of ψ_i. Queues of students may form at each professor over time. We assume that there is a maximum possible queue length, denoted by L. The evolution of the queue length is governed by student arrivals and student departures upon completing a meeting with the professor. Let X_i be the random variable representing the students' inter-arrival time at professor i and x_i be a realization of X_i. Let Y_i be the random variable representing the duration of a meeting between a student and professor i and y_i be a realization of Y_i. We consider a discrete-time Markov model and therefore assume X_i and Y_i are geometric random variables with parameters p_{x_i} and p_{y_i}, respectively. Further, we assume the distributions of X_i and Y_i are independent.

We consider a discrete-time planning horizon $0, 1, \ldots, T$, where $T = \max_{i \in V} l_i$ is the time after which no more decisions are made. Let Ξ_k be a random variable representing the time of the kth decision epoch and ξ_k be the realization of Ξ_k. In the MDP, decision epochs are triggered by one of the following conditions:

- if the salesperson is en route towards professor i, the next decision epoch is triggered by the

arrival of the salesperson, i.e., $\xi_k = \psi_i$;

- if the salesperson is waiting at a professor, the next decision epoch is triggered by observing the first queueing event(s) or reaching a specified amount of time δ, whichever comes first, where δ is the longest time that the salesperson will wait before making a decision while at professor i. Thus, $\xi_k = \xi_{k-1} + \min\{x_i, y_i, \delta\}$.

The state of the system represents the sufficient information for the salesperson to execute a decision of whether to stay and wait at the current professor or to leave and go to another professor. The state consists of information on the salesperson's status as well as the status of each professor. The status of the salesperson is captured by the triple (t, d, q), where $q \in \{\{?\} \cup [0, L]\}$ is the queue length at the professor $d \in V$ at the current system time t. Note that $q =?$ indicates the queue length is currently unknown.

We represent the status of professors by partitioning V into three sets, H, U, and W. The set $H \subseteq V$ represents the set of professors who the salesperson has met, thereby collecting a reward. The set $U \subseteq V$ represents the set of professors whom the salesperson has not yet visited. The set $W \subseteq V$ represents the set of professors whom the salesperson has visited, but from whom the salesperson has departed before initiating a meeting. For each professor $w \in W$, the state includes information $(\breve{t}_w, \breve{q}_w)$ representing the queue length $\breve{q}_w$ observed at the time $\breve{t}_w$ that the salesperson departed professor w. The salesperson can then use this information to evaluate a decision to revisit a professor $w \in W$. Let $(\breve{t}, \breve{q}) = (\breve{t}_w, \breve{q}_w)_{w \in W}$ denote the vector of information regarding the time and queue length at previously-visited but unmet professors.

Thus, we represent the complete state of the system with the tuple $(t, d, q, H, U, W, (\breve{t}, \breve{q}))$ consisting of the information on the salesperson's status as well as the status of the set of professors. The state space S is defined on $[0, T] \times V \times \{\{?\} \cup [0, L]\} \times V \times V \times V \times \{([0, T], [0, L])\}^V$. The initial state is $s_0 = (0, 0, 0, \emptyset, V, \emptyset, (\emptyset, \emptyset))$. An absorbing state must meet one of the following conditions:

(i) the salesperson has met with all professors;

(ii) it is infeasible for the salesperson to arrive at any unmet professor within his/her time window.

Thus, the set of absorbing states is defined as $S_K = \Big\{(t, d, q, H, U, W, (\breve{t}, \breve{q})) : H = V \text{ or } t + c_{di} \geq l_i, \forall i \in \{V \setminus H\}\Big\}$.

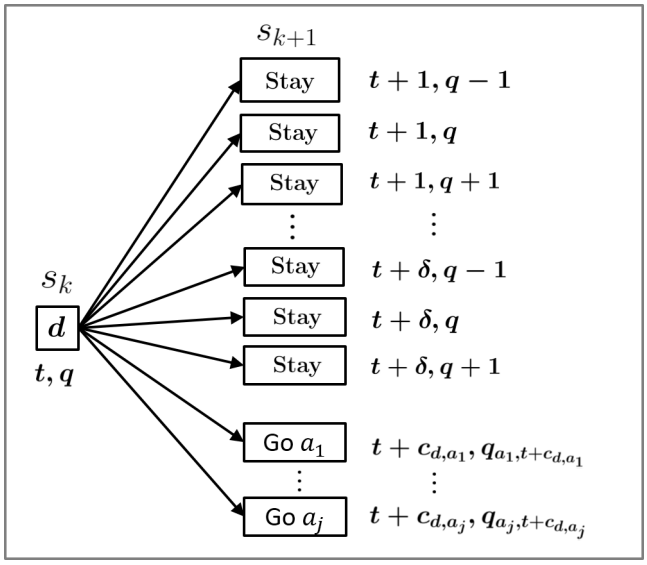

Figure 1: State Transition

At each decision epoch, the salesperson selects an action that determines the salesperson's location at the next decision epoch. At a professor d at time t, a salesperson can either stay at professor d if she has not yet met professor d or go to another unmet professor i with $l_i \geq t + c_{di}$. Thus, the action space for state s is $A(s) = \Big\{a \in V : t + c_{da} \leq l_a, a \in \{U \cup W\}\Big\}$.

3.2 System Dynamics

Figure 1 depicts the state transition that occurs upon action selection. In the remainder of this section, we describe this state transition in more detail. At decision epoch k with state $s_k = (t, d, q, H, U, W, (\breve{t}, \breve{q}))$, an action $a \in A(s_k)$ is selected, initiating a transition from state s_k to state $s_{k+1} = (t', d', q', H', U', W', (\breve{t}', \breve{q}'))$. The location at s_{k+1} is the selected action, $d' = a$. Let $P\{s_{k+1}|s_k, a\}$ be the transition probability from state s_k to s_{k+1} when selecting action a in s_k. At each epoch, the transition probability is defined by one of two action cases: (i) the salesperson decides to depart from the current location and go to another professor, or (ii) the salesperson decides to stay at the current professor and wait in line.

If the selected action in state s_k is to go to another professor a, $P\{s_{k+1}|s_k, a\}$ is specified by the queue length distribution at professor a at the arrival time, $t + c_{da}$. In §A of the electronic companion, we derive a parameterized distribution of the queue length at a professor given the arrival time. If the observed queue length is $q' > 0$ when the salesperson arrives at professor a,

then the salesperson is unable to immediately meet with the professor and no reward is collected at time $t' = t + c_{da}$. If the observed queue length is $q' = 0$, the salesperson meets with professor and collects reward r_a. We assume that the reward is collected as long as the salesperson meets with the professor, regardless of the length of the meeting. Let S_a be the random variable with a known distribution representing the duration of the meeting between the salesperson and professor a. Thus, when the selected action is to go to professor a, we update the current time as

$$t' = \begin{cases} t + c_{da}, & \text{if } q' > 0, \\ t + c_{da} + s_a, & \text{if } q' = 0, \end{cases}$$

where s_a is a realization of S_a.

In the second case, if the selected action in state s_k is to stay at the current professor d, the current time of state s_{k+1}, $t + \min\{X_d, Y_d, \delta\}$, is either the time of the first queueing event(s) or that of reaching the specified decision-making interval δ. The first queueing event may be a student arrival that increases the queue length by one, a student departure that decreases the queue length by one, or both an arrival and a departure where the queue length remains the same as in state s_k. If no queueing events occur before or at the decision-making interval $t + \delta$, the queue length remains the same. If $q' = 0$ at time $t + \min\{X_d, Y_d, \delta\}$, the salesperson meets with professor d for a duration of s_d time units and a reward of r_d is collected. Otherwise, no reward is collected. Finally, when the selected action is to stay at the current professor, the current time is updated as

$$t' = \begin{cases} t + \min\{X_d, Y_d, \delta\}, & q' > 0, \\ t + \min\{X_d, Y_d, \delta\} + s_d, & q' = 0, \end{cases}$$

where $0 < \kappa \leq \delta$. We present the state transition probability $P\{s_{k+1}|s_k, a\}$ for the action of staying at the current professor in §A of the electronic companion.

Regardless of whether the selected action is "to go" or "to stay," we update the set of met professors by

$$H' = \begin{cases} H, & \text{if } q' > 0, \\ H \cup \{a\}, & \text{if } q' = 0. \end{cases}$$

If $q' > 0$, the set of met professors remain the same as in state s_k. If $q' = 0$, professor a is added to set H. We update the set of unvisited professors by

$$U' = \begin{cases} U, & \text{if } a = d, \\ U \setminus \{a\}, & \text{if } a \neq d. \end{cases}$$

If $a = d$, the set of unvisited professors remains the same as in state s_k. If $a \neq d$ and a was previously unvisited, professor a is removed from the set U. We update the set of visited but unmet professors by

$$W' = \begin{cases} W, & \text{if } a = d, q' > 0 \text{ or } a \neq d, q' = 0, \\ W \setminus \{a\}, & \text{if } a = d, q' = 0, \\ W \cup \{a\}, & \text{if } a \neq d, q' > 0. \end{cases}$$

If the salesperson stays at the current location and does not meet with the professor ($a = d, q' > 0$), or if the salesperson goes to another professor a and meets with this professor ($a \neq d, q' = 0$), the set of visited but unmet professors remains the same as in s_k. If the salesperson stays and meets with the current professor ($a = d, q' = 0$), this professor is removed from set W. If the salesperson goes to another professor a and observes a queue ($a \neq d, q' > 0$), professor a is added to set W. Finally, we update information $(\breve{t}', \breve{q}')$ by adding $(t_a, q_a) = (t', q')$ to $(\breve{t}, \breve{q})$ if $q' > 0$ and by removing (t_a, q_a) from $(\breve{t}, \breve{q})$ if $q' = 0$.

Let Π be the set of all Markovian decision policies for the problem. A policy $\pi \in \Pi$ is a sequence of decision rules: $\pi = (\rho_0^\pi(s_0), \rho_1^\pi(s_1), ..., \rho_k^\pi(s_k))$, where $\rho_k^\pi(s_k) : s_k \mapsto A(s_k)$ is a function specifying the action to select at decision epoch k while following policy π. For notational simplicity, we denote $\rho_k^\pi(s_k)$ using ρ_k^π. Let $R_k^\pi(s_k)$ be the expected reward collected at decision epoch k if following policy π. Our objective is to seek a policy π that maximizes the total expected reward over all decision epochs: $\sum_{k=1}^{K} R_k^\pi(s_k)$, where K is the final decision epoch. Let $V(s_k)$ be the expected reward-to-go from decision epoch k through K. Then an optimal policy can be obtained by solving $V(s_k) = \max_{a \in A(s_k)} \left\{ R_k(s_k, a) + E[V(s_{k+1}) | s_k, a] \right\}$ for each decision epoch k and the corresponding state s_k.

3.3 Structural Results

As the salesperson is routing on a network of queues, it is logical to investigate the presence of optimal control limit policies. We prove that, by fixing the sequence of professors, at any given time, there exists a control limit in queue length for each professor. That is, there exists a threshold

queue length such that if it is optimal to stay at a professor when observing this queue length, it is also optimal to stay when observing a shorter queue at the same time. The proposition is stated in Theorem 1 and the proof is in the electronic companion.

Theorem 1. *For each professor i and a given time t, there exists a threshold q such that for any $q' \geq q$, it is optimal to leave and for any $q' < q$, it is optimal to stay.*

Unfortunately, there does not exist a control limit with respect to the salesperson's arrival time at a professor. Thus, even if it is optimal to stay at a professor when the salesperson arrives and observes a queue length, it is not necessarily optimal to stay if she arrives earlier and observes the same queue length. The reason for the lack of control limit structure with respect to arrival time is because the salesperson may be more likely to collect rewards from professors later in the visit sequence if she leaves the current professor at an earlier time. We demonstrate this via a counter example provided in the electronic companion.

In the remainder of this section, we investigate conditions under which actions are guaranteed to be suboptimal. By eliminating these actions, we reduce the action space and improve the computational efficiency of our approximate dynamic programming approach. In the following, we first prove that given that the salesperson has the lowest priority in the queue, there is no value for her to arrive at a professor before the time when students start arriving at the professor.

Proposition 1. *Let $o_i (\leq e_i)$ be the earliest time that students may start arriving at professor i. Given a priority queue at professor i, there is no value in the salesperson arriving at a professor earlier than o_i.*

Proof. Proof of Proposition 1 We prove this proposition by contradiction. Let s_k be the state resulting from arriving at professor i at time t ($t < o_i \leq e_i$) and s'_k be the state resulting from arriving at professor i at time $t' (\geq o_i)$. Suppose there is value in the salesperson arriving at professor i at time $t (< o_i)$, i.e., $V(s_k) > V(s'_k)$. By definition, state s_k includes no information on queue length as students have not yet began to arrive, while state s'_k includes $(t', q_i) \in (\breve{t}, \breve{q})$ as the salesperson will observe a queue length of q_i at time t'. Also, the salesperson cannot commence meeting the professor as the time window has not opened. Further, because the salesperson has a lower priority than the students, even if the salesperson arrives at time t and waits until t', any

student that arrives between o_i and t' will queue before the salesperson. Thus, the salesperson receives no information nor gains queue position by arriving at time t, i.e., $V(s_k) \leq V(s'_k)$, which contradicts the assumption. □

Based on Proposition 1, Theorem 2 states that there is also no value in departing early from a professor and going to another professor before a certain time.

Theorem 2. *Assuming $q_i > 0$ at time t, there is no value in the salesperson leaving professor i and going to a professor $j \in \{U \cup W\}$ $(j \neq i)$ until time $o_j - c_{ij}$.*

Proof. Proof of Theorem 2 We prove this theorem by contradiction. Suppose there is value in the salesperson leaving professor i for a professor $j \in \{U \cup W\}$ at time t such that $t < o_j - c_{ij}$. Then the salesperson will arrive at professor j at time $t + c_{ij} < o_j$. According to Proposition 1, there is no value in arriving at a professor j before o_j, which contradicts the assumption. □

As a consequence of Theorem 2, the actions corresponding to leaving the current professor and going to another professor before certain time can be eliminated.

Corollary 1. *If the salesperson leaves professor i at time t, the action of going to a professor $j \in \{U \cup W\}$ at time t can be eliminated if $o_j \geq \max_{\substack{k \in \{U \cup W\} \\ k \neq j}} \{t + c_{ik} + s_k + c_{kj}\}$.*

Proof. Proof of Corollary 1 If $o_j \geq \max_{\substack{k \in \{U \cup W\} \\ k \neq j}} \{t + c_{ik} + s_k + c_{kj}\}$, the salesperson can still arrive at professor j before o_j by going to another professor $k \in \{U \cup W\}$ first and then to professor j. According to Proposition 1, there is no value for the salesperson to arrive at professor j before o_j. Therefore, not going to professor j at time t would not affect the value collected by the salesperson and the action of going to professor j at time t can be eliminated. □

4 Rollout Policies

To find an optimal policy for the salesperson, we must solve the optimality equation $V(s_k) = \max_{a \in A(s_k)} \left\{R_k(s_k, a) + E[V(s_{k+1})|s_k, a]\right\}$ for state s_k at each decision epoch k. However, given the curses of dimensionality present in the model and the limits of our structural results, it is not practical to exactly determine optimal policies. Instead, we turn to rollout algorithms to develop

rollout policies. A form of approximate dynamic programming (see Powell (2007) for a general introduction to approximate dynamic programming), rollout algorithms construct rollout policies by employing a forward dynamic programming approach and iteratively using heuristic policies to approximate the reward-to-go at each decision epoch. Specifically, from a current state s_k at decision epoch k, rollout algorithms select an action a based on $\hat{V}(s_k) = \max_{a \in A(s_k)} \left\{ R_k(s_k, a) + E[\hat{V}(s_{k+1})|s_k, a] \right\}$, where $\hat{V}(s_{k+1})$ is approximated by the value of heuristic policies. For an in-depth discussion on rollout algorithms, we refer the readers to Bertsekas et al. (1997), Bertsekas (2005), and Goodson et al. (2014a).

In §4.1, we present a heuristic a-priori-route policy for estimating the reward-to-go. In §4.2, we briefly summarize existing rollout algorithm methodology from the literature. In §4.3, we propose a compound rollout algorithm that is based on a partitioned action space and hierarchal application of heuristic policies.

4.1 A-Priori-Route Policy

To estimate the reward-to-go at a decision epoch, a rollout algorithm requires a heuristic policy to apply along the possible sample paths, where a heuristic policy is a suboptimal policy for all future states. For the DOPTW, we use a class of *a-priori-route heuristic policies* to approximate the reward-to-go. Given a state s_k, the corresponding *a-priori-route policy* $\pi(\nu)$ is characterized by an a priori route $\nu = (\nu_1, \nu_2, ..., \nu_m)$, which specifies a pre-planned order of visits to m professors for the salesperson (see Campbell and Thomas (2008) for an overview on a priori routing). The a priori route starts at the current location d at time t of state s_k (i.e., $\nu_1 = d$), followed by a sequence of $m-1$ professors in set $\left\{ \{U \cup W\} \setminus \{d\} \right\}$. The realized queue information at the current location d is given by the value of q in state s_k.

To account for the random information realized during the execution of an a priori route, we implement the two types of recourse actions proposed by Zhang et al. (2014). The first recourse action, skipping the next professor in the sequence, is motivated by the notion that if the salesperson arrives late in a professor's time window, she may be unlikely to meet the professor due to the length of the queue. Zhang et al. (2014) establish a static rule stating that the salesperson will skip a professor i if the salesperson cannot arrive by a specified time $\tau_i \leq l_i$. The second recourse action corresponds to the queueing behaviors of balking and reneging. Zhang et al. (2014) utilize the

queue length observed upon arrival to establish a static rule setting the longest amount of time γ_i that the salesperson will wait at professor i. With the wait time distribution presented in the electronic companion, we implement the static decision rules established by Zhang et al. (2014) to determine the value of τ_i and γ_i for each professor i in an a priori route.

We execute a variable neighborhood descent (VND) procedure to search for an a-priori-route from the space of all possible a priori routes associated with state s. We outline the VND algorithm in §D of the electronic companion. We denote by $V^{\pi(\nu)}(s)$ the expected value of following a-priori-route policy $\pi(\nu)$ from state s. The objective is to find an a priori route ν^* that induces the optimal a-priori-route policy $\pi(\nu^*)$ such that $V^{\pi(\nu^*)}(s) > V^{\pi(\nu)}(s)$ for every $\pi(\nu)$. To reduce the computational burden of exactly evaluating the objective of $V^{\pi(\nu)}(s)$, we use Monte Carlo simulation to estimate $V^{\pi(\nu)}(s)$ collected from every neighbor policy.

4.2 Rollout Algorithms

In general, a rollout algorithm develops rollout policies by looking ahead at each decision epoch and approximating reward-to-go through heuristic policies, such as the a-priori-route policy presented in §4.1. Depending on how many steps the rollout procedure looks ahead before applying the heuristic, rollout algorithms can be categorized as one-step or multi-step rollout (Bertsekas (2005), Secomandi (2001), Novoa and Storer (2009)). More recently, Goodson et al. (2014a) characterize pre- and post-decision rollout which can be viewed as zero-step and half-step look-ahead procedures based on the pre-decision and post-decision states of stochastic dynamic programming. In the remainder of this section, we focus on describing one-step and pre-decision rollout decision rules, which we incorporate in our compound rollout algorithm.

When occupying state s_k and evaluating action a, the one-step rollout transitions to all possible states at the next decision epoch $s_{k+1} \in S(s_k, a)$, where $S(s_k, a) = \{s_{k+1} : P\{s_{k+1}|s_k, a\} > 0\}$. For each of these possible future states s_{k+1}, we obtain an a-priori-route policy $\pi(\nu, s_{k+1})$ by executing the VND heuristic presented in the electronic companion. The estimated reward-to-go for selecting action a in s_k is given by the expected value of a-priori-route policies obtained in all possible state s_{k+1}:

$$E\big[V^{\pi(\nu,s_{k+1})}(s_{k+1})|s_k, a\big] = \sum_{s_{k+1}\in S(s_k,a)} P\{s_{k+1}|s_k, a\} \times V^{\pi(\nu,s_{k+1})}(s_{k+1}), \tag{1}$$

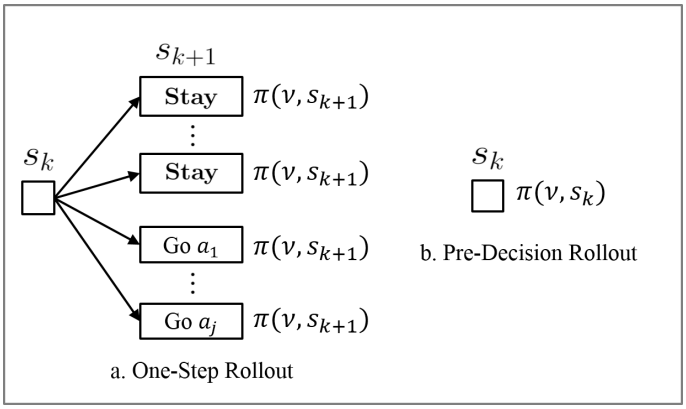

Figure 2: Rollout Decision Rule

where $V^{\pi(\nu,s_{k+1})}(s_{k+1})$ is the expected value of an a-priori-route policy $\pi(\nu)$ originating from state s_{k+1}. When the process occupies state s_k at decision epoch k, it selects an action $a \in A(s_k)$ such that the value of $R_k(s_k, a) + E\big[V^{\pi(\nu,s_{k+1})}(s_{k+1})\big)|s_k, a\big]$ is maximized. Figure 2a provides a visual representation of the one-step rollout procedure.

For a state s_k and each feasible action $a \in A(s_k)$, one-step rollout executes the search heuristic $|S(s_k, a)|$ times to find an a-priori-route policy, which results in applying the heuristic a total of $\sum_{a \in A(s_k)} |S(s_k, a)|$ times to select an action at s_k. As an example, consider the case in which the salesperson needs to decide between staying at the current professor and going to one of the five other professors. If a decision epoch occurs every minute ($\delta = 1$) while waiting at the current professor and there are four possible queue lengths at the professors, then the heuristic will be executed 23 times to select an action (3 of these correspond to the "stay" decision and 5 x 4 = 20 correspond to the "go" decision). Notably, for the action of staying, the value of $|S(s_k, a)|$ is affected by the choice of δ. For the same example as above, if $\delta = 5$, then there are 35 executions of the heuristic. While the problem size and the value of δ increase, $|A(s_k)|$ and $\sum_{a \in A(s_k)} |S(s_k, a)|$ increase, so selecting an action by evaluating $R_k(s_k, a) + E\big[V^{\pi(\nu,s_{k+1})}(s_{k+1})\big)|s_k, a\big]$ becomes computationally challenging even when determining the heuristic policy using local search and approximating $V^{\pi(\nu,s_{k+1})}(s_{k+1})$ using Monte Carlo sampling.

The formalization of the pre-decision rollout is motivated by the computational issues associated with one-step rollout (Goodson et al. (2014a)). As shown in Figure 2b, the pre-decision state decision rule does not look ahead at all, but instead selects the action to take in state s_k via an a-priori-route policy $\pi(\nu, s_k)$ starting at the current location d_k and time t_k of state s_k. Specifically,

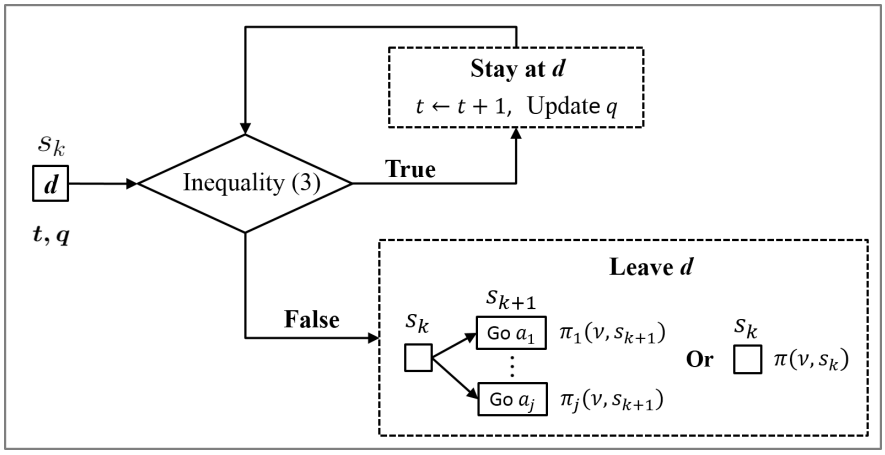

Figure 3: Compound Rollout

the threshold value γ_d calculated for the current location d in the a priori route indicates how long to stay at this professor. If $\gamma_d \neq 0$, the action selected in s_k is to stay at professor d for a duration of γ_d. If $\gamma_d = 0$, then the action is to go to the first professor specified in the a priori route $\pi(\nu, s_k)$ after leaving professor d at time t. In this case, the VND heuristic is executed only once in each state s_k.

4.3 Compound Rollout

Because of the large state and action spaces induced by the waiting process, using one-step rollout to solve realistically sized instances is not computationally tractable, even when limiting per action evaluation run time to one minute. Thus, we propose a compound rollout algorithm to reduce the computational burden of one-step rollout and improve the policy quality of pre-decision rollout. As discussed in §3.1, if the salesperson has not met a professor after arrival, she can either stay at the professor or go to another unmet professor. Compound rollout considers the stay-or-go decision in two stages. In the first stage, the salesperson decides whether or not to stay at the current professor. If the decision made in the first stage is to no longer stay at the current professor, compound rollout then enters the second stage to determine which professor to visit next. That is, we partition the action space $A(s_k)$ into two sets. One set is the singleton $\{d\}$ (if $d \in A(s_k)$), corresponding to staying and waiting at the current professor. The other set is $\big\{A(s_k) \setminus \{d\}\big\}$, composed of actions of leaving and going to another professor.

In the first stage of compound rollout, the salesperson will stay at the current professor if the

expected total reward collected by staying at time t is greater than the expected total reward collected if the salesperson departs at time t. Specifically, the salesperson will stay at the current professor at time t if

$$P(\Omega_{dt} < l_d - t|q,t) \times r_d + \sum_{\omega_{dt} \leq l_d - t} P(\Omega_{dt} = \omega_{dt}) V_{go}(t + \omega_{dt}) > V_{go}(t), \tag{2}$$

where Ω_{dt} is the random variable representing the wait time at professor d given the queue length q observed at time t and $V_{go}(t)$ denotes the expected reward-to-go if departing professor d at time t and going to another professor. We derive the distribution of Ω_{dt} in §B of the electronic companion. The comparison in Inequality (2) requires the computing estimates of $V_{go}(\cdot)$ over the entire support of Ω_{dt}, which may be too computationally expensive for a real-time algorithm. Therefore, we replace Inequality (2) with

$$P(\Omega_{dt} < l_d - t|q,t) \times r_d + V_{go}(t + E[\Omega_{dt}|q,t]) > V_{go}(t). \tag{3}$$

We denote by $E[\Omega_{dt}|q,t]$ the expected wait time for observing queue length q at time t. We use $E[\Omega_{dt}|q,t]$ to estimate the actual wait time for the salesperson. To approximate the value of $V_{go}(t)$ and $V_{go}(t + E[W_{dt}|q,t])$, we execute the VND heuristic onward from state s_k to find a-priori-route policies $\pi(\nu, s_k)$ and $\pi(\bar{\nu}, s_k)$ with expected value $V^{\pi(\nu,s_k)}(s_{k+1})$ and $V^{\pi(\bar{\nu},s_k)}(s_{k+1})$, respectively. Notably, though both policies start at professor d, the time to leave professor d in policy $\pi(\nu, s_k)$ is t, and in policy $\pi(\bar{\nu}, s_k)$, it is $t + E[W_{dt}|q,t]$.

As Figure 3 shows, the salesperson will stay and wait at professor d while Inequality (3) is true. If deciding to stay, she will wait for one time unit and re-evaluate the criteria. To do so, we advance the current time t by one, generate a realization of the queue length at time $t+1$ (from the queue length distribution derived in §A of the electronic companion), and re-evaluate Inequality (3) with the updated information. The salesperson will leave professor d once Inequality (3) does not hold. Then compound rollout enters the second stage to decide which professor to go to next. Note that, if $q = 0$ at time t, the salesperson will meet with the professor and then leave. In this case, the next professor to visit is determined similarly to the second stage of our compound rollout.

As the lower-right hand corner of Figure 3 shows, in the second stage, compound rollout employs

one-step rollout or pre-decision rollout to approximate the reward-to-go associated with candidates in the set $\{A(s_k) \setminus \{d\}\}$. In Algorithm 1, we formalize the action-selection logic used by the compound rollout algorithm. Line 1 and Line 2 indicate the criteria to stay at the current professor d or to go to another professor. If Inequality (3) indicates "go" rather than "stay", compound rollout implements one-step rollout (Line 5) or pre-decision rollout (Line 6) to select the next professor to visit after leaving professor d. We note that while applying pre-decision rollout from state s_k to make "go" decision, the salesperson will visit the first professor specified in the a priori route $\pi(\nu, s_k)$ after d, i.e., ν_2 from $\nu = (\nu_1, \nu_2, ..., \nu_m)$ with $\nu_1 = d$.

Algorithm 1: Compound Rollout

1: **if** $d \in A(s_k)$ and Inequality (3) is true **then**
2: $\quad a^* \leftarrow d$
3: **else**
4: $\quad$ Implement (i) one-step rollout or (ii) pre-decision rollout:
5: $\quad\quad$ (i) $a^* \leftarrow \arg\max_{a \in \{A(s_k) \setminus \{d\}\}} \left\{ R_k(s_k, a) + E\left[V^{\pi(\nu, s_{k+1})}(s_{k+1}) | s_k, a \right] \right\}$
6: $\quad\quad$ (ii) $a^* \leftarrow \nu_2$, where $\nu_2 \in \pi(\nu, s_k)$

5 Computational Experiments

In this section, we present computational results from applying the compound rollout procedure of §4.3. In §5.1, we detail the generation of problem instances for the DOPTW from existing benchmark problems. In §5.2, we provide implementation details, and in §5.3 and §5.4, we present and discuss our computational results.

5.1 Problem Instances

We derive our datasets from Solomon's VRPTW instances (Solomon, 1987). We modify benchmark instances corresponding to randomly-located professors (R sets), clustered professors (C sets), and a mix of random and clustered professors (RC sets). A textbook salesperson typically visits between 10 and 20 professors depending on the professors' locations and office hours, i.e., fewer visits correspond to more widely scattered professors and/or more overlapping time windows (personal communication, Tracy Ward, Senior Learning Technology Representative at McGraw-Hill Education, September 18, 2014). Accordingly, we select the first 20 professors from Solomon's instances.

For each instance, we maintain each professor's location and demand information as in Solomon (1987). However, we consider an eight-hour working day for professors and assume the length of a professor's office hours to be either 60 minutes, 90 minutes or 120 minutes. To create a 480-minute working day, we first scale time from the original time horizon in Solomon's instances to a 480-minute day, [0, 480]. If the resulting time window is either 60 minutes, 90 minutes, or 120 minutes wide, then it is complete. If the resulting time window for professor i is less than 60 minutes, we modify it to 60 minutes by setting $l_i = e_i + 60$. If the resulting time window for professor i is between 60 minutes and 90 minutes wide, we modify it to 90 minutes by setting $l_i = e_i + 90$. If the resulting time window width for professor i exceeds 90 minutes, we set it to 120 minutes by setting $l_i = e_i + 120$. These DOPTW data sets are available at `http://ir.uiowa.edu/tippie_pubs/63`.

5.2 Implementation and Details

We code our algorithms in C++ and execute the experiments on 2.6-GHz Intel Xeon processors with 64-512 GB of RAM. As mentioned in §4.1, we use a VND procedure to find the a-priori-route policies whose expected rewards are used to approximate the reward-to-go. In the VND, we estimate the expected value of heuristic policies using Monte Carlo sampling with 1000 samples. In this study, we consider one minute as the minimum time increment in the discrete-time Markov process. Thus, we use the best a-priori-route policy returned by the VND within a minute to approximate the reward-to-go. We determine the first professor to visit by employing the VNS from Zhang et al. (2014) as this first decision does not need to be solved within one minute. We implement the action elimination procedures presented in §3.3. For each problem instance, we randomly generate 1000 sample paths based on the transient queue length distribution presented in §A of the electronic companion. For the geometric distributions of student arrivals and departures, we respectively set $p_{x_i} = 0.125$ and $p_{y_i} = 0.1$ for all professors $i \in V$.

5.3 Computational Results

To benchmark the performance of our rollout policies, we generate lower and upper bounds on the reward collected for each instance. To obtain a lower bound, we execute the a priori routes produced by the approach of Zhang et al. (2014). We establish an upper bound on each instance by solving it with the best-case assumption that the salesperson experiences zero wait time at each professor.

Dataset	A Priori	Dynamic	CI	Gap	UB
R101	114.85	115.27	*[114.32, 116.23]*	0.37%	175
R102	131.11	133.65	[132.90, 134.39]	1.94%	203
R103	112.81	116.01	[114.69, 117.34]	2.84%	195
R104	104.71	110.10	[108.71, 111.48]	5.14%	189
R105	113.60	116.96	[115.80, 118.13]	2.96%	200
R106	125.82	130.59	[129.70, 131.48]	3.79%	221
R107	110.35	115.89	[114.89, 116.88]	5.02%	197
R108	101.65	106.76	[105.69, 107.83]	5.03%	189
R109	122.68	125.57	[124.73, 126.42]	2.36%	219
R110	110.06	117.65	[116.75, 118.55]	6.90%	189
R111	122.62	128.92	[127.99, 129.85]	5.14%	208
R112	92.23	95.60	[94.56, 96.63]	3.65%	177
R201	126.40	128.46	[127.63, 129.29]	1.63%	203
R202	124.46	132.32	[131.37, 133.26]	6.31%	215
R203	110.56	117.72	[116.58, 118.85]	6.47%	196
R204	98.95	107.66	[106.44, 108.88]	8.80%	189
R205	130.70	132.29	[131.36, 133.22]	1.22%	229
R206	126.64	135.16	[134.10, 136.22]	6.73%	230
R207	110.99	116.67	[115.67, 117.66]	5.11%	199
R208	99.57	106.50	[105.38, 107.61]	6.95%	189
R209	118.06	122.02	[121.06, 122.97]	3.35%	202
R210	127.64	132.19	[131.20, 133.18]	3.56%	228
R211	106.77	111.87	[110.78, 112.95]	4.77%	204

Table 1: DOPTW Results for R Instances with 20 Professors

As we already assume deterministic travel times between professors, the problem then becomes a deterministic orienteering problem. We employ the dynamic programming solution approach modified by Zhang et al. (2014) from Feillet et al. (2004) to solve this deterministic problem as an elementary longest path problem with resource constraints.

In our computational experiments, we solve the DOPTW with compound rollout using one-step rollout in the second stage. Tables 1 through 3 compare the compound rollout policies to the bounds. In each of the three tables, the second and sixth columns report the lower and upper bounds obtained in the manner discussed in the previous paragraph. The third column reports the average objectives over 1000 sample paths from the compound rollout policies obtained by using one-step rollout to make the "go" decision. The fourth column present the 95% confidence intervals on the dynamic objective values. The fifth column shows the gap between the a priori and the dynamic solutions ($Gap = \frac{Dynamic - A\ Priori}{A\ Priori} \times 100\%$).

Dataset	A Priori	Dynamic	CI	Gap	UB
C101	249.56	253.20	[252.22, 254.18]	1.46%	340
C102	212.69	216.94	[215.64, 218.24]	2.00%	330
C103	198.02	204.81	[203.46, 206.16]	3.43%	310
C104	168.00	176.84	[175.32, 178.36]	5.26%	280
C105	251.06	255.20	[254.30, 256.10]	1.65%	340
C106	248.74	255.11	[254.10, 256.12]	2.56%	340
C107	237.06	241.63	[240.70, 242.56]	1.93%	350
C108	235.37	238.86	[237.57, 240.15]	1.48%	360
C109	219.41	224.89	[223.71, 226.07]	2.50%	360
C201	288.25	291.08	[289.85, 292.31]	0.98%	360
C202	262.76	264.69	[263.76, 265.62]	0.74%	330
C203	208.85	212.62	[211.38, 213.86]	1.81%	300
C204	192.48	201.27	[199.91, 202.63]	4.57%	280
C205	290.80	295.58	[294.30, 296.86]	1.64%	350
C206	272.27	278.48	[277.02, 279.94]	2.28%	360
C207	233.21	247.59	[246.01, 249.17]	6.17%	360
C208	271.12	279.52	[278.52, 280.52]	3.10%	360

Table 2: DOPTW Results for C Instances with 20 Professors

Dataset	A Priori	Dynamic	CI	Gap	UB
RC101	170.46	177.14	[175.56, 178.72]	3.92%	260
RC102	207.8	214.13	[212.25, 216.01]	3.05%	330
RC103	191.07	199.24	[197.42, 201.06]	4.28%	320
RC104	174.4	183.64	[182.07, 185.21]	5.30%	300
RC105	203.71	207.42	[206.20, 208.64]	1.82%	300
RC106	180.07	188.60	[186.74, 190.46]	4.74%	320
RC107	165.37	173.61	[171.89, 175.33]	4.98%	290
RC108	175.6	177.05	[176.13, 177.97]	0.83%	300
RC201	215.48	219.63	[218.37, 220.89]	1.93%	300
RC202	218.13	223.24	[221.71, 224.77]	2.34%	340
RC203	205.2	207.44	[205.73, 209.15]	1.09%	330
RC204	168.84	176.52	[174.97, 178.07]	4.55%	300
RC205	210.95	220.65	[219.05, 222.25]	4.60%	350
RC206	213.18	219.65	[217.79, 221.51]	3.03%	370
RC207	196.41	195.49	*[193.92, 197.06]*	-0.47%	350
RC208	189.07	197.56	[195.86, 199.26]	4.49%	330

Table 3: DOPTW Results for RC Instances with 20 Professors

Overall, the average objectives of compound rollout policies are 3.47% better than the a priori solutions. The average gap between the dynamic and a priori solutions for C instances is 2.56%, 4.35% for the R instances, and 3.15% for the RC instances. As shown in the fourth columns of the tables, for 54 out of 56 instances, the compound rollout policies are statistically better than the a priori solutions based on a 95% confidence level. For the remaining 2 instances (R101 and RC207), the compound rollout policies are not significantly different from the a priori solutions. In part, the lack of difference is due to the constraints imposed by these instances. In instance R101, every professor has 60-minute time window, and because of this limited availability, no revisits occur in the compound rollout policies, which reduces the possible advantage of the dynamic solution over an a priori solution.

As mentioned in §5.2, we employ VNS to search for the a priori solutions. On average, the search time taken by VNS is around 23 minutes. In the dynamic compound rollout approach, we only allow a VND heuristic one minute of search time per epoch according to the minimum time increment considered in our model. Running the VND within this restricted time may lead to inferior a-priori-route policies and thereby imprecise estimates of reward-to-go. In general, the rollout procedure helps overcome the minimal runtime afforded to the search heuristic as noted in Chang et al. (2013, p. 197) "... what we are really interested in is the ranking of actions, not the degree of approximation. Therefore, as long as the rollout policy preserves the true ranking of actions well, the resulting policy will perform fairly well." However, for some instances, the values of the a priori route policies affect the choice of the next professor to visit, thereby the quality of dynamic policies. For C instances, with clustered professors and overlapping time windows, the values of the a priori route policies used by the rollout approach to select the next professor to visit are close together, and it is difficult to distinguish one from the other within the one-minute search time. We test instances C102 and C104 by approximating the reward-to-go with a-priori-route policies obtained via executing VND for up to five minutes. The average objective of C102 improves from 216.94 to 219.4 or by 1.13% and that of C104 improves from 176.84 to 178.57 or by 0.98%. In the case of instance RC207, in which there is a mixture of random and clustered customers, substantial runtime was required to distinguish one choice from another due to significant overlap in the time windows within a cluster. We test instance RC207 by obtaining a-priori-route policies via running VNS for up to 20 minutes and the average objective improves

Dataset	C.I. on Improv.	Dataset	C.I. on Improv.	Dataset	C.I. on Improv.
R101	[2.46%, 4.77%]	R109	[0.13%, 2.30%]	R205	[3.06%, 5.21%]
R102	[2.44%, 4.08%]	R110	[2.69%, 4.94%]	R206	[4.25%, 6.37%]
R103	[-1.95%, 1.28%]	R111	[2.40%, 4.60%]	R207	[2.60%, 4.94%]
R104	[-1.20%, 2.41%]	R112	[0.31%, 3.50%]	R208	[4.80%, 7.78%]
R105	[2.63%, 5.24%]	R201	[0.48%, 2.37%]	R209	[2.70%, 4.92%]
R106	[3.07%, 5.04%]	R202	[0.87%, 2.87%]	R210	[1.58%, 3.68%]
R107	[0.36%, 2.83%]	R203	[0.07%, 2.74%]	R211	[4.05%, 6.65%]
R108	[-0.56%, 2.30%]	R204	[4.01%, 7.30%]		
C101	[-0.54%, 0.57%]	C107	[0.39%, 1.52%]	C204	[1.07%, 3.10%]
C102	[-1.06%, 0.63%]	C108	[-1.04%, 0.51%]	C205	[0.12%, 1.39%]
C103	[-1.24%, 0.65%]	C109	[-0.56%, 0.88%]	C206	[2.70%, 4.13%]
C104	[-1.65%, 0.86%]	C201	[0.50%, 1.63%]	C207	[-0.34%, 1.48%]
C105	[-0.52%, 0.54%]	C202	[-0.41%, 0.60%]	C208	[-0.35%, 0.69%]
C106	[-0.25%, 0.87%]	C203	[-0.22%, 1.44%]		
RC101	[-0.28%, 2.24%]	RC107	[5.02%, 8.02%]	RC205	[0.01%, 2.06%]
RC102	[3.81%, 6.35%]	RC108	[2.04%, 3.64%]	RC206	[-0.75%, 1.74%]
RC103	[4.22%, 6.92%]	RC201	[-0.14%, 1.51%]	RC207	[1.49%, 3.86%]
RC104	[-1.27%, 1.21%]	RC202	[0.28%, 2.35%]	RC208	[-1.08%, 1.36%]
RC105	[0.59%, 2.19%]	RC203	[-1.35%, 0.88%]		
RC106	[-0.12%, 2.64%]	RC204	[1.29%, 3.81%]		

Table 4: Comparison beteween Compound and Pre-Decision Rollout

from 195.49 to 200.62, with a confidence interval of [198.96, 202.29], in which case the dynamic solution is statistically better than the a priori solution.

The upper bound obtained by solving the DOPTW with the assumption of zero wait time at professors is weak in all problem instances. The average gap between dynamic solutions and upper bounds is 56.58%, suggesting that the queueing effects at the professors is the complicating feature in this problem. However, even for small size problems, it is computationally intractable to solve the DOPTW optimally to obtain an accurate upper bound.

To demonstrate the value of looking ahead and observing the queue length information at the next decision epoch while making the "go" decision, we compare our compound rollout policies using one-step rollout in making the "go" decision with those using pre-decision rollout. Specifically, line 5 in Algorithm 1 is implemented when using one-step rollout in the second stage, and line 6 is implemented when using pre-decision rollout. In the following discussion, we name the compound rollout procedure using one-step rollout in making "go" decision the *compound-one-step rollout* and

that using pre-decision rollout the *compound-pre-decision rollout*. In Table 4, we provide detailed comparisons between the two types of policies. The second, fourth, and sixth columns present the 95% confidence intervals on the percentage of improvement that compound-one-step rollout policies have over compound-pre-decision rollout policies. While, on average, the compound-pre-decision rollout policies are 1.48% better than the a priori solutions, they are 1.98% worse than compound-one-step rollout policies. As shown in the table, out of 56 instances, there are 34 instances for which the compound-one-step rollout policies are statistically better than the compound-pre-decision rollout policies. Out of these 34 instances, compound-one-step rollout has more than 2% of average improvement over compound-pre-decision rollout.

5.4 Policy Analysis

To illustrate the differences between a priori solutions and our compound rollout policies, we provide a detailed comparison of select instances. In the comparison, the compound rollout policies are obtained by using one-step rollout in making the "go" decision. Overall, the advantage of dynamic solutions over a priori solutions is that dynamic solutions select actions based on the realized random information, which may enable the salesperson to adapt to the observed queues, e.g., visiting more or different professors by taking advantage of favorable realizations or revisiting a professor. In contrast, a priori solutions specify a fixed sequence of professors that maximizes the collected reward averaged over all possible scenarios. Further, it is difficult to construct a fixed sequence to facilitate revisiting professors as this action comes as a reaction to a specific queue length observations.

We compare a priori solutions and compound rollout policies for instances R107 and C204 in Figures 4 and 5. In each figure, the professor indices are ordered according to the a priori route. For each professor, the left bar corresponds to the a priori solution and the right bar represents data from the compound rollout policy. However, we note that the sequence of professors in a dynamic solution does not necessarily correspond to the a priori sequence as the dynamic approach adjusts the sequence based on realized observations. The overall height of each bar shows the probability of visiting a professor. For a priori solutions, we compute the probability of the salesperson visiting but not meeting a professor and the probability of meeting a professor via the analytical formulation of Zhang et al. (2014). For dynamic policies, the probabilities of visiting but not meeting a professor,

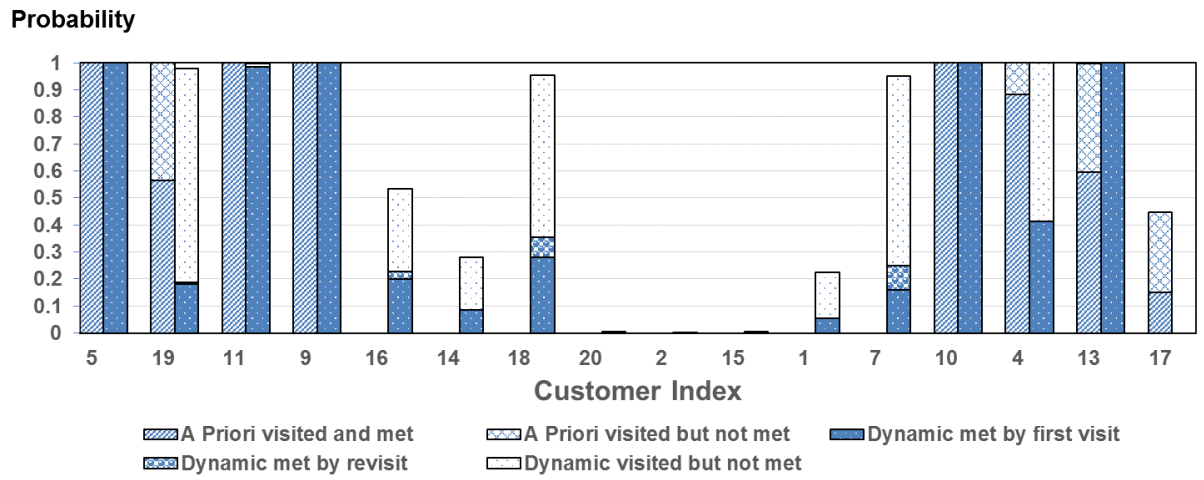

Figure 4: Compound Rollout vs. A Priori for Dataset R107

meeting a professor on the first visit, and meeting a professor by revisiting are computed via simulation with 1000 sample paths.

For instance R107 with 20 professors, the a priori route is (5, 19, 11, 9, 16, 14, 18, 20, 2, 15, 1, 7, 10, 4, 13, 17). Note that professors not visited by the salesperson in the a priori or dynamic solutions are not listed here. From Figure 4, we can see that when professors are randomly located, the a priori solution is only able to visit 8 out of 20 professors, while dynamic solution is likely to visit 15 professors. Both the a priori and dynamic solutions are able to meet professors 5, 11, 9, and 10 with certainty or near certainty. The a priori solution is more likely to visit and meet professors 19 and 4, and has around 45% of likelihood of visiting and 15% of likelihood of meeting professor 17, who is skipped in the dynamic solution. However, the dynamic solution is more likely to meet with professor 13, who has the second largest reward of all professors, and has a positive probability of visiting professors 16, 14, 18, 20, 2, 15, 1, 7, who are always skipped by the a priori solution. By allowing revisiting, the dynamic solution increases the likelihood of collecting rewards at several professors. For instance, revisiting leads to around a 10% of increase in the likelihood of meeting professor 7, a more than 5% for professor 18, and around a 3% of increase for professor 16.

Similarly, for instance C204, the a priori route is (16, 14, 12, 19, 18, 17, 10, 5, 2, 1, 7, 3, 4, 15, 13, 9, 11, 8), where professors skipped in both the a priori and dynamic tours are not listed. We note that the salesperson is able to visit and meet more professors when professors are clustered than randomly located as in the R instances. As Figure 5 shows, professors 16, 4, 15, 13, 9, and 8

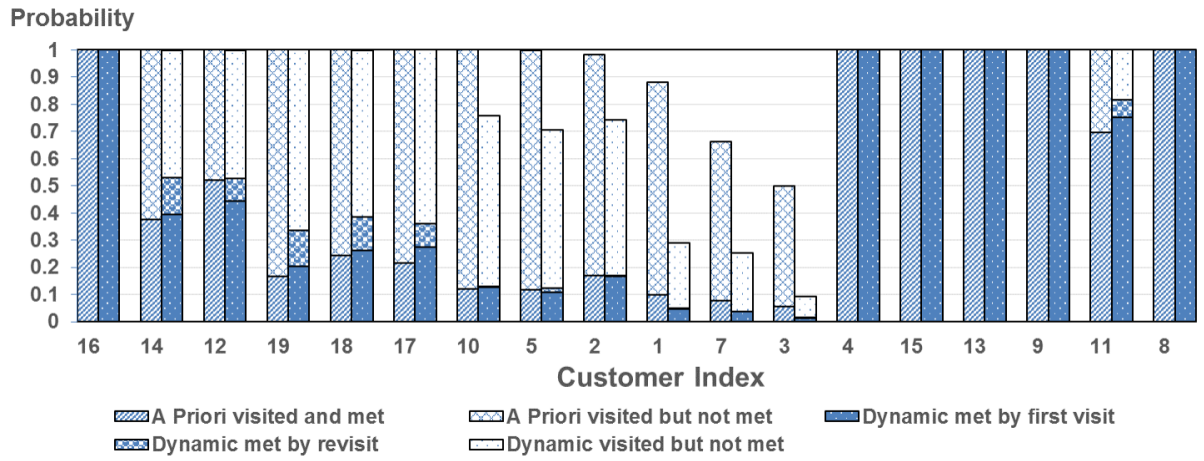

Figure 5: Compound Rollout vs. A Priori for Dataset C204

are always visited and met by the salesperson in both the a priori and dynamic solutions. Both a priori and dynamic solutions always visit professor 12 and a priori solution is more likely to visit professors 10, 5, and 2, but both solutions have similar likelihoods of meeting professors 12, 10, 5, and 2. The a priori solution is more likely to visit and meet professors 1, 7, and 3. However, the overall performance of the dynamic solution is better because it has a higher probability of meeting with professors 14, 19, 18, 17, and 11, which together provide more reward compared to professors 10, 5, and 2. Revisits enable the dynamic solution to increase the likelihood of collecting rewards from professors 14, 12, 19, 18, 17, and 11. Specifically, revisits to professors 14, 19, and 18 in the dynamic solution lead to over 10% of increase in the likelihood of meeting these professors, while increasing the likelihood of meeting professors 12, 17, and 11 by up to 8%. The likelihood of meeting professor 12 by the first visit in the dynamic solution is lower than that of the a priori solution. However, revisits make the overall probability of meeting professor 12 from the dynamic solution greater than that of the a priori solution.

6 Summary and Future Work

Motivated by the daily routing problem faced by a textbook salesperson, we introduce a dynamic orienteering problem with time windows (DOPTW) characterized by a queueing process and a time window at each professor. Specifically, the queueing process induces uncertain wait times at professors. To generate dynamic policies that govern routing and queueing decisions, we propose a

novel compound rollout algorithm that explicitly partitions the action space and executes rollout policies based on the partitioning. Our dynamic policies perform favorably in comparison to the a priori routing solutions with recourse actions. We demonstrate the value of incorporating queueing information in making "where to go" decision by comparing the compound rollout using one-step rollout to make the "go" decision to that using pre-decision rollout which does not look ahead when selecting a professor to go next.

On average, our compound-one-step rollout policies perform about 3.47% better than the a priori solutions, which suggests the merit in solving the DOPTW dynamically. By looking ahead and involving future queue information in approximating reward-to-go when making "where to go" decision, our compound-one-step rollout policies are preferable to the compound-pre-decision rollout policies by having an average improvement of 1.98% on objective values. However, compound-pre-decision rollout is a viable alternative if computation becomes a concern.

An important direction for future work is to extend from the one-day routing problem to a multi-period problem. As mentioned in §1, the textbook salesperson visits professors multiple times during the weeks preceding the textbook adoption decision. To maximize the overall likelihood of gaining adoptions, a dynamic rolling plan needs to be developed so that the salesperson can incorporate a priori daily routing results in planning campus visits for the following periods.

A State Transition Probabilities

In this section, we derive the state transition probabilities, $P\{s_{k+1}|s_k, a\}$, for the various state-action pairs. For both the decision to stay at the current professor and the decision to go to another professor, the transition probabilities depend on the evolution of the queue at the respective professor. As mentioned in §3.1, the evolution of the queue length is governed by student arrivals and student departures upon completing a meeting with the professor. We define X_i as the random variable representing the students' inter-arrival time at professor i and x_i is a realization of X_i. We define Y_i as the random variable representing the duration of a meeting between a student and professor i and y_i is a realization of Y_i. We consider a discrete-time Markov model and assume X_i and Y_i are geometric random variables with parameters p_{x_i} and p_{y_i}, respectively. We assume the distributions of X_i and Y_i are independent. Let Q_{it} be the random variable representing the queue

length observed at professor i at time t and q_{it} be the realization of Q_{it}. We model the evolution of queue at professor i as a discrete-time Markov Chain (DTMC) $\{Q_{it}, t = 0, 1, 2, ..., T\}$. We assume that $\{Q_{it}, t = 0, 1, 2 ..., T\}$ and $\{Q_{jt}, t = 0, 1, 2, ..., T\}$ are independent Markov chains for $i \neq j$.

Given $0 < q_{it} < L$, the possible realizations of $q_{i,t+1}$ are:

- $q_{i,t+1} = q_{it} + 1$ with probability $p_{x_i}(1 - p_{y_i})$ if there is only a student arrival and no student departure occurring at time $t + 1$;
- $q_{i,t+1} = q_{it} - 1$ with probability $p_{y_i}(1 - p_{x_i})$ if there is only a student departure and no student arrival at time $t + 1$;
- $q_{i,t+1} = q_{it}$ with probability $(1 - p_{x_i})(1 - p_{y_i}) + p_{x_i}p_{y_i}$ if there are no queueing events or both a student arrival and a departure at time $t + 1$.

If $q_{it} = 0$, then $q_{i,t+1} = 1$ or $q_{i,t+1} = 0$ with probabilities p_{x_i} and $1 - p_{x_i}$, respectively. If $q_{it} = L$, then $q_{i,t+1} = L - 1$ with probability p_{y_i} and $q_{i,t+1} = L$ with probability $1 - p_{y_i}$. Consequently, the DTMC on state space $\{0, 1, 2, ...L\}$ is described by the following time-homogeneous one-step transition matrix:

$$R_i = \begin{bmatrix} 1 - p_{x_i} & p_{x_i} & 0 & \cdots & 0 & 0 & 0 \\ (1 - p_{x_i})p_{y_i} & \overline{p}^i & (1 - p_{y_i})p_{x_i} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & (1 - p_{x_i})p_{y_i} & \overline{p}^i & (1 - p_{y_i})p_{x_i} \\ 0 & 0 & 0 & \cdots & 0 & p_{y_i} & 1 - p_{y_i} \end{bmatrix},$$

where $\overline{p}^i = (1 - p_{y_i})(1 - p_{x_i}) + p_{x_i}p_{y_i}$.

From a state $s_k = (t, d, q, H, U, W, (\breve{t}, \breve{q}))$, consider the transition to a state s_{k+1} when deciding to leave the current professor d and go to another professor z. As mentioned in §3.1, the only uncertain element of this transition is the queue length observed at professor z at time $t + c_{dz}$. Therefore, the stochastic transition from s_k to s_{k+1} is governed by the queue length distribution.

If $z \in U$, the distribution of queue lengths observed at time $t + c_{dz}$ is defined over $[0, L]$, by $\phi_0 R_z^{(\max\{t+c_{dz}, e_z\} - o_z)}$, where ϕ_k is a vector with a one in the position corresponding to queue length k (≥ 0), $R_z^{(n)}$ is the n-step transition matrix, and o_z ($\leq e_z$) is the earliest time that a queue (of students) begins to form at professor z. If $z \in W$, the distribution of queue lengths observed at

time $t + c_{dz}$ is defined over $[0, L]$, by $\phi_{\breve{q}_z} R_z^{(\max\{t+c_{dz}, e_z\} - \max\{\breve{t}_z, o_z\})}$. Note that $\breve{t}_z$ is the time of the most recent visit to professor z and $\breve{q}_z$ is the queue length observed at time $\breve{t}_z$.

Now consider the transition to a state s_{k+1} from a state $s_k = (t, d, q, H, U, W, (\breve{t}, \breve{q}))$ when deciding to stay at the current professor d. As mentioned in §3.1, the next decision epoch that is triggered by a queueing event or δ, the maximum amount of time between epochs, occurs at time $t + \min\{X_d, Y_d, \delta\}$. Thus, both the queue length and the system time of the next epoch is uncertain. The state transition probability $P\{s_{k+1}|s_k, d\}$ is defined over 3δ possible states characterized by $q' \in \{q+1, q, q-1\}$ and $t' \in \{t+1, t+2, ..., t+\delta\}$. Let κ be the realization of $\min\{X_d, Y_d, \delta\}$. When deciding to stay at the current professor, the time of next decision epoch is $t + \kappa$ $(0 < \kappa \leq \delta)$. For $\kappa < \delta$, the event triggering the epoch at $t + \kappa$ is:

(i) a student arrival increasing queue length by one with probability $p_{x_d}(1 - p_{x_d})^{\kappa-1}(1 - p_{y_d})^{\kappa}$;

(ii) a student departure decreasing queue length by one with probability $p_{y_d}(1-p_{y_d})^{\kappa-1}(1-p_{x_d})^{\kappa}$;

(iii) the concurrent arrival and departure so that queue length remains the same as in state s_k^a with probability $p_{x_d}p_{y_d}[(1 - p_{x_d})(1 - p_{y_d})]^{\kappa-1}$;

For $\kappa = \delta$, events (i), (ii), and (iii) can occur with the associated probabilities. Additionally, with probability $(1 - p_{x_d})^{\delta}(1 - p_{y_d})^{\delta}$, no queueing events may occur before or at time $t + \delta$ so that the queue length remains the same as in state s_k.

B Distribution of Wait Time

In this section, we derive the distribution of wait time Ω_{it} at professor i if observing queue length q_{it} at time t. As the salesperson has the lowest priority, Ω_{it} is the time it takes the observed queue q_{it} to diminish to zero. Thus, the wait time for q_{it} at professor i is the first passage time from state q_{it} to state 0 for the DTMC described by the one-step transition matrix R_i in §A. Let $f_{q0}^i(n)$ be the probability that the first passage from state q $(0 \leq q \leq L)$ to state 0 occurs after n periods at professor i. Let $P_{qj}^i(n)$ be the probability that starting from state q, the DTMC is in state j after n periods at professor i. As presented in Scholtes (2001), based on the Bayes' formula, we have

$$P_{q0}^i(n) = P_{00}^i(n-1)f_{q0}^i(1) + \ldots + P_{00}^i(1)f_{q0}^i(n-1) + f_{q0}^i(n). \tag{4}$$

Then $f^i_{q0}(n)$ can be computed recursively via:

$$\begin{aligned} f^i_{q0}(1) =& P^i_{q0}(1) \\ & \vdots \\ f^i_{q0}(n) =& P^i_{q0}(n) - f^i_{q0}(1)P^i_{00}(n-1) - \ldots - f^i_{q0}(n-1)P^i_{00}(1). \end{aligned} \tag{5}$$

For queue length q_{it} observed at time t at professor i, the distribution of Ω_{it} is defined over $\big[\max\{e_i - t, 0\}, l_i - t\big]$, by the q_ith row of the following matrix

$$\begin{bmatrix} f^i_{00}(1) & f^i_{00}(2) & \cdots & f^i_{00}(l_i - t - 1) & \overline{f}^i_{00} \\ f^i_{10}(1) & f^i_{10}(2) & \cdots & f^i_{10}(l_i - t - 1) & \overline{f}^i_{10} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f^i_{L-1,0}(1) & f^i_{L-1,0}(2) & \cdots & f^i_{L-1,0}(l_i - t - 1) & \overline{f}^i_{L-1,0} \\ f^i_{L0}(1) & f^i_{L0}(2) & \cdots & f^i_{L0}(l_i - t - 1) & \overline{f}^i_{L0} \end{bmatrix},$$

where $\overline{f}^i_{q0} = 1 - \sum_{k=1}^{l_i - t - 1} f_{q0}(k)$ for $q = 0, \ldots, L$.

C Analytical Results on Queueing Control Limits

In this section, we first prove the results on the optimal control limit policies presented in §3.1. We prove that fixing the sequence of professors, there exist a control limit in queue length at any given time for each professor. That is, if it is optimal to stay at a professor at time t when observing a queue length q, it is also optimal to stay when observing a shorter queue length q' ($q' < q$) at the same time t. Then we provide counter example showing that there does not exist a control limit with respect to the salesperson's arrival time at a professor. Specifically, even if it is optimal to stay at a professor while the salesperson arrives at time t and observes a queue length q, it is not necessarily optimal to stay if she arrives earlier at time t' ($t' < t$) and observes the same queue length.

We denote by $g_i(q'|q,t)$ the distribution of the current queue length at professor i given the queue length q observed t time units ago. According to the transient queue length distribution presented in §A, the distribution $g_i(q'|q,t)$ is given by the qth row of the t-step transition matrix $R_i^{(t)}$. Let $Stay(i,t,q)$ be the expected reward of staying and waiting at professor i with queue

length q at time t. Let $Leave(i,t,q)$ be the onward expected reward of leaving and going to the next available professor $i+1$ in the sequence when there is a queue length q at professor i at time t. The formulations of $Stay(i,t,q)$ and $Leave(i,t,q)$ are:

$$\begin{aligned} Stay(i,t,q) &= \sum_{q'} g_i(q'|q,1)v(i,t+1,q'), \\ Leave(i,t,q) &= \sum_{q'} P(Q_{i+1,t+c_{i,i+1}} = q')v(i+1,t+c_{i,i+1},q'). \end{aligned} \tag{6}$$

Note that the queue length at professor $i+1$ at time $t+c_{i,i+1}$ is independent with that at professor i at time t. Let $v(i,t,q)$ be the expected reward from professor i and the onward reward from other professors on the route if the salesperson observes queue q at professor i at time t. By definition, $v(i,t,q)$ can be represented as:

$$v(i,t,q) = \max\big\{Stay(i,t,q), Leave(i,t,q)\big\}. \tag{7}$$

Before proving Theorem 1, we state and prove a series of Lemmas.

Lemma 1. *The distribution of the queue length has the Increasing Failure Rate (IFR) property, i.e.,* $\sum_{q=k}^{L} g_i(q|\tilde{q},t) \le \sum_{q=k}^{L} g_i(q|q',t)$ *for all* i, k, t, *and* $q' > \tilde{q}$.

Proof. Proof of Lemma 1 We prove this lemma by induction. First, by the definition of R_i in §A, there is $\sum_{q=k}^{L} g_i(q|\tilde{q},1) \le \sum_{q=k}^{L} g_i(q|q',1)$ for all k and $q' > \tilde{q}$ ($\tilde{q}, q' \in [0,L]$). Assuming $\sum_{q=k}^{L} g_i(q|\tilde{q},n) \le \sum_{q=k}^{L} g_i(q|q',n)$ for all k and $q' > \tilde{q}$, now we show $\sum_{q=k}^{L} g_i(q|\tilde{q},n+1) \le \sum_{q=k}^{L} g_i(q|q',n+1)$ for all k and $q' > \tilde{q}$. Then,

$$\sum_{q=k}^{L} g_i(q|\tilde{q},n+1) = \sum_{q=k}^{L}\sum_{r=0}^{L} g_i(q|r,n)g_i(r|\tilde{q},1) \tag{8}$$

$$\le \sum_{q=k}^{L}\sum_{r=0}^{L} g_i(q|r,n)g_i(r|q',1) \tag{9}$$

$$= \sum_{q=k}^{L} g_i(q|q',n+1), \tag{10}$$

Equalities (8) and (10) follow the definition of n-step transition probability. Inequality (9) results from the induction hypothesis. □

Lemma 2. *For each professor i, $v(i,t,\tilde{q}) \geq v(i,t,q')$, for all i, t and $q' > \tilde{q}$.*

Proof. Proof of Lemma 2 We prove the lemma by induction. First, we show that, with $T = \max_{i \in V} l_i$, $v(i,T,\tilde{q}) \geq v(i,T,q')$ for all i and $q' > \tilde{q}$. By definition, when there is a queue at time T, $Stay(i,T,q) = 0$ and $Leave(i,T,q) = 0$. Therefore, $Stay(i,T,q)$ and $Leave(i,T,q)$ are non-increasing in q. As the the maximum of non-increasing functions is non-increasing, $v(i,T,\tilde{q}) \geq v(i,T,q')$ for $\tilde{q} < q'$.

Assuming $v(i,t+1,\tilde{q}) \geq v(i,t+1,q')$ for all i and $q' > \tilde{q}$, we next show that $v(i,t,\tilde{q}) \geq v(i,t,q')$ for all i and $q' > \tilde{q}$. We have:

$$
\begin{aligned}
& Stay(i,t,\tilde{q}) - Stay(i,t,q') \\
=& \sum_q g_i(q|\tilde{q},1) v(i,t+1,q) - \sum_q g_i(q|q',1) v(i,t+1,q) && (11) \\
\geq& 0. && (12)
\end{aligned}
$$

Equality (11) follows by definition. By the induction hypothesis, $v(i,t+1,q)$ is non-increasing in q. By Lemma 1, $\sum_{q=k}^{L} g_i(q|q',1)$ is a non-decreasing function of q' for all k. Thus, $\sum_{q=k}^{L} g_i(q|q',1) v(i,t+1,q)$ is non-increasing in q', thereby implying Inequality(12). By definition, $Leave(i,t,\tilde{q}) = Leave(i,t,q')$ because the queue length at professor i has no bearing on the queue length at professor $i+1$. So both $Stay(i,t,q)$ and $Leave(i,t,q)$ are non-increasing on q for all i and t. Thus $v(i,t,q)$ is non-increasing on q for all i and t, i.e., $v(i,t,\tilde{q}) \geq v(i,t,q')$ for all i, t and $q' > \tilde{q}$. □

Theorem 3. *For each professor i and a given time t, there exists a threshold $\tilde{q}$ such that for all $q' \geq \tilde{q}$, it is optimal to leave and for all $q' < \tilde{q}$, it is optimal to stay.*

Proof. Proof of Theorem 3 We prove the result by showing that if it is optimal to leave at time t while observing queue length $\tilde{q}$, it is also optimal to leave at time t while observing queue length

$q' > \tilde{q}$. Suppose it is optimal to leave at time t with queue length $\tilde{q}$, then

$$\begin{aligned}
&Leave(i,t,\tilde{q}) - Stay(i,t,\tilde{q}) \\
=&\sum_q P(Q_{i+1,t+c_{i,i+1}} = q) v(i+1, t+c_{i,i+1}, q) - \sum_q g_i(q|\tilde{q},1) v(i,t+1,q) \qquad (13) \\
\geq& 0.
\end{aligned}$$

Now we show that it is also optimal to leave at t with $q' > \tilde{q}$. We have:

$$\begin{aligned}
&Leave(i,t,q') - Stay(i,t,q') \\
=&\sum_q P(Q_{i+1,t+c_{i,i+1}} = q) v(i+1, t+c_{i,i+1}, q) - \sum_q g_i(q|q',1) v(i,t+1,q) \qquad (14) \\
\geq&\sum_q P(Q_{i+1,t+c_{i,i+1}} = q) v(i+1, t+c_{i,i+1}, q) - \sum_q g_i(q|\tilde{q},1) v(i,t+1,q) \qquad (15) \\
\geq& 0. \qquad (16)
\end{aligned}$$

Equality (14) follows by definition. Inequality (15) results from Lemma 2, and Inequality (16) follows from the assumption. Therefore, it is also optimal to leave professor i at t when $q' > \tilde{q}$. □

We next provide an example showing that, given a queue, there does not exist a control limit with respect to the salesperson's arrival time at a professor. We consider a case with two professors. The first professor is available during time window [0,40] and associated with a reward of 30, while the second professor is available during time window [0,60] and with a reward of 50. Assuming the queue length observed at professor 1 upon arrival is 2, we show that although it is optimal for the salesperson to join the queue at professor 1 when she arrives at time 25, it is not optimal to join when she arrives at an earlier time 10.

Table 5 and 6 detail the counter-example. Let A_i be the random variable representing the arrival time at professor i and S_i represent the meeting duration between professor i and the salesperson. Let Ω_{it} be the random variables representing the wait time at professor i. Let R_i be the reward collected at professor i. In Table 5, the first column presents the two arrival times considered at professor 1. The second column reports the wait time distribution with respect to each arrival time and the given queue length of 2. We use ">" to denote the situations in which

the salesperson will not be able to meet with the professor even if she waits till the end of the time window. We assume the travel time between the two professors to be 20 and the meeting duration with the professor 1 is 10. The third and fourth columns report whether the salesperson can meet with professor 1 and the reward collected from professor 1. The fifth through the seventh columns correspond to the time the salesperson spends in meeting with professor 1, the arrival time at professor 2, and the expected reward from professor 2 if the salesperson stays in queue at professor 1. The ninth and tenth columns present the arrival time at professor 2 and the expected rewards from professor 2 if balking at professor 1. Note that the expected rewards from professor 2 in the seventh and tenth columns are computed in Table 6. The eighth and eleventh columns present the overall expected rewards from the two professors for the action of staying in queue and balking at professor 1, respectively. Similarly, in Table 6, the first column includes the arrival times reported in the sixth and ninth columns of Table 5. The second and third columns present the queue length and wait time distributions, respectively. The fourth column indicates whether the salesperson can meet with professor 2 and the fifth column report the reward collected at professor 2. The sixth column presents the expected reward from professor 2 according to the queue length and wait time distributions.

When the salesperson arrives at professor 1 at the time of 10, with professor 2 having greater value, by leaving professor 1 earlier she may be more likely to collect reward from professor 2 and thereby maximize the total expected reward from the tour. If arriving at professor 1 at the later time of 25, however, the salesperson may be less likely to collect reward from professor 2, even if leaving professor 1 immediately upon arrival. Thus, it is better for her to stay in line at professor 1. As shown in Table 5, for an arrival time of 10, it is optimal to balk and go to professor 2 because $E[\text{Stay at 1}] = 30 < E[\text{Leave 1}] = 34$. For an arrival time of 25, with $E[\text{Stay at 1}] = 6 > E[\text{Leave 1}] = 2$, it is optimal to stay in queue at professor 1.

D Variable Neighborhood Descent Algorithm

We present the variable neighborhood descent (VND) heuristic used to search the a-priori-route policy for approximating the reward-to-go in Algorithm 2. Line 3 of Algorithm 2 initializes the search with an a-priori-route policy from state s with randomly ordered professors. Line 5 states

Arrive Time	Wait Time Dist.	Stay at Professor 1						Leave Professor 1		
A_1	ω_{1t}, $P(\omega_{1t})$	Meet	R_1	S_1	A_2	$E[R_2]$	E[Stay at 1]	A_2	$E[R_2]$	E[Balk at 1]
10	(10, 0.4)	Yes	30	10	50	0	30	30	34	34
	(20, 0.6)	Yes	30	10	60	0				
25	(10, 0.2)	Yes	30	10	65	0	6	45	2	2
	(>15, 0.8)	No	0	0	60	0				

Table 5: Professor 1 with time window [0, 40]

Arrive Time	Queue Dist.	Wait Time Dist.	Meet	R_2	$E[R_2]$
A_2	$q_{2t}, P(q_{2t})$	$\omega_{2t}, P(\omega_{2t})$			
30	(1, 0.6)	(10, 0.3)	Yes	50	34
		(15,0.7)	Yes	50	
	(2, 0.4)	(20, 0.2)	Yes	50	
		(>30, 0.8)	No	0	
45	(2, 0.4)	(10, 0.1)	Yes	50	2
		(>15, 0.9)	No	0	
	(3, 0.6)	(>15, 1)	No	0	
50	(3, 0.2)	(>10, 1)	No	0	0
	(4, 0.8)	(>10, 1)	No	0	

Table 6: Professor 2 with time window [0, 60]

Algorithm 2: Variable Neighborhood Descent (VND)

Input: A pre- or post-decision state s, an ordered set of neighborhoods N
Output: An a priori routing policy $\pi(\nu)$
$\pi(\nu) \leftarrow initialize()$
$iter \leftarrow 0, k \leftarrow 1, level \leftarrow 1, count \leftarrow 1, improving \leftarrow true$
while runtime < 59 seconds **and** $iter < iterationMax$ **or** $level < levelMax$ **do**
  $\pi'(\nu) \leftarrow Shake\big(\pi(\nu)\big)$
  while *improving* **and** runtime < 59 seconds **do**
    $\pi''(\nu) \leftarrow BestNeighbor\big(\pi'(\nu), k\big)$
    if $V^{\pi''(\nu)}(s) > V^{\pi'(\nu)}(s)$ **then**
      $\pi'(\nu) \leftarrow \pi''(\nu), count \leftarrow 1$
    else if $k = |N|$ and $count < |N|$ **then**
      $k \leftarrow 1, count \leftarrow count + 1$
    else if $k < |N|$ and $count < |N|$ **then**
      $k \leftarrow k + 1, count \leftarrow count + 1$
    else
      $improving \leftarrow false$
  if $V^{\pi''(\nu)}(s) > V^{\pi(\nu)}(s)$ **then**
    $\pi(\nu) \leftarrow \pi''(\nu)$
    $level \leftarrow 1$
  else
    $level \leftarrow level + 1$
  $iter \leftarrow iter + 1$

the termination criteria for the VND. As mentioned in §5.2, the minimum time increment in our discrete Markov process is one minute, so we restrict the heuristic search time for an a-priori-route policy to 59 seconds in Line 5. Within the time limit, we use the variable *iter* and *level* to ensure that the algorithm performs at least $iterationMax$ iterations in total and $levelMax$ iterations after finding an improved solution. We determine that setting $iterationMax = 35$ and $levelMax = 15$ balances the tradeoff between computational efficiency and heuristic performance. The *Shake* procedure in Line 6 randomly re-sequences professors in policy $\pi(\nu)$. Line 7 through Line 16 find a locally-optimal policy relative to a set of specified neighborhoods N, which is composed of 1-shift and 2-opt neighborhoods in our study. The $BestNeighbor$ function in Line 8 returns the best policy in the neighborhood k of the current policy $\pi'(\nu)$. Line 9 through Line 16 manage the updating of the locally optimal policy, the active neighborhood, and the variable *count* that ensures all neighborhoods are explored before terminating. Line 17 through Line 21 update the current solution and the value of *level*. Line 22 advances the interaction counters.

References

Atkins, A. 2013. The high cost of textbooks. Retrieved October 2, 2014, `http://atkinsbookshelf.wordpress.com/tag/how-much-do-students-spend-on-textbooks/`.

Bertsekas, D. P. 2005. *Dynamic Programming and Optimal Control*, vol. I. 3rd ed. Athena Scientific, Belmont, MA.

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

Burnetas, A. N. 2013. Customer equilibrium and optimal strategies in Markovian queues in series. *Annals of Operations Research* **208** 515–529.

Campbell, A. M., M. Gendreau, B. W. Thomas. 2011. The orienteering problem with stochastic travel and service times. *Annals of Operations Research* **186** 61–81.

Campbell, A. M., B. W. Thomas. 2008. The probabilistic traveling salesman problem with deadlines. *Transportation Science* **42** 1–21.

Chang, H. S., M. C. Fu, J. Hu, S. I. Marcus. 2013. Simulation-based algorithms for markov decision processes. *Communications and Control Engineering*, 2nd ed., chap. 5. Springer, London, 179–226.

Cheng, S., X. Qu. 2009. A service choice model for optimizing taxi service delivery. *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, Piscataway, NJ, 66–71.

D'Auria, B., S. Kanta. 2011. Equilibrium strategies in a tandem queue under various levels of information. `http://e-archivo.uc3m.es/bitstream/10016/12262/1/ws113325.pdf`. Working paper.

Evers, L., K. Glorie, S. van der Ster, A. I. Barros, H. Monsuur. 2014. A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research* **43** 248–260.

Feillet, D., P. Dejax, M. Gendreau, C. Gueguen. 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* **44**(3) 216–229.

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

Goodson, J. C., B. W. Thomas, J. W. Ohlmann. 2014a. A generalized rollout policy framework for stochastic dynamic programming. `http://www.slu.edu/~goodson/papers/GoodsonRolloutFramework.pdf`. Working paper.

Goodson, J. C., B. W. Thomas, J. W. Ohlmann. 2014b. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. *Transportation Science* to appear.

Honnappa, H., R. Jain. 2015. Strategic arrivals into queueing networks: The network concert queueing game. *Operations Research* Articles in Advance.

Hussar, W. J., T. M. Bailey. 2013. Projections of education statistics to 2021. Tech. rep., National Center for Education Statistics. Http://nces.ed.gov/pubs2013/2013008.pdf.

Novoa, C., 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.

Papapanagiotou, V., D. Weyland, R. Montemanni, L. M. Gambardella. 2013. A sampling-based approximation of the objective function of the orienteering problem with stochastic travel and service times. *Lecture Notes in Management Science* **5** 143–152.

Powell, W.B. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, *Wiley Series in Probability and Statistics*, vol. 703. Wiley-Interscience, Hoboken, New Jersey.

Scholtes, S. 2001. Markov chain. `http://www.eng.cam.ac.uk/~ss248/G12-M01/Week3/Lecture.ppt`.

Secomandi, N. 2000. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research* **27** 1201–1224.

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

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

Tang, H., E. Miller-Hooks. 2005. Algorithms for a stochastic selective travelling salesperson problem. *Journal of the Operational Research Society* **56**(4) 439–452.

Teng, S. Y., H. L. Ong, H. C. Huang. 2004. An integer L-shaped algorithm for time-constrained traveling salesman problem. *Asia-Pacific Journal of Operational Research* **21**(2) 241–257.

Toriello, A., W. B. Haskell, M. Poremba. 2014. A dynamic traveling salesman problem with stochastic arc costs. *Operations Research* **62**(5) 1107–1125.

Vansteenwegen, P., W. Souffriau. 2010. Tourist trip planning functionalities: State-of-the-art and future. *Lecture Notes in Computer Science* **6385** 474–485.

Yechiali, U. 1971. On optimal balking rules and toll charges in the GI/M/1 queuing process. *Operations Research* **19**(2) 349–371.

Yechiali, U. 1972. Customers' optimal joining rules for the G1/M/s queue. *Management Science* **18**(7) 434–443.

Zhang, S., J. W. Ohlmann, B. W. Thomas. 2014. A priori orienteering with time windows and stochastic wait times at customers. *European Journal of Operational Research* **239** 70–79.