Summer 2015

Stochastic orienteering on a network of queues with time windows

Shu Zhang
University of Iowa

STOCHASTIC ORIENTEERING ON A NETWORK OF QUEUES WITH TIME WINDOWS

by

Shu Zhang

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Business Administration
in the Graduate College of
The University of Iowa

August 2015

Thesis Supervisors: Associate Professor Jeffrey Ohlmann
Associate Professor Barrett Thomas

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Shu Zhang

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Business Administration at the August 2015 graduation.

Thesis Committee: _____
Jeffrey Ohlmann, Thesis Supervisor

_____
Barrett Thomas, Thesis Supervisor

_____
Ann Campbell

_____
Renato de Matta

_____
Yong Chen

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my co-advisors Jeffrey Ohlmann and Barrett Thomas. Without their guidance and support, I could not imagine finishing this thesis. I have learned a great deal from them in the past few years, in and out of our weekly meetings. They are not just my advisors, but my role models.

I would like to thank my committee members, Ann Campbell, Renato de Matta, and Yong Chen, for their contributions. I am very fortunate to be in a department with great faculty members. I would like to express my sincere appreciation to professors Nick Street, Sam Burer, Ann Campbell, and Ray de Matta, for their support and encouragement during my time at the University of Iowa. In particular, I thank Ann Campbell for her invaluable suggestions for the thesis and advices about my career. I do appreciate all the administrative assistance that our department secretary, Barb Carr, and the Ph.D. program coordinator, Renea Jay have provided. Without their efforts, I could have a rougher experience in earning this degree.

I gratefully acknowledge the Information Technology Services at the University of Iowa for providing the high-performance computing resources, without which I would not be able to carry out my computational experiments. I would like to thank Ben Rogers and Glenn Johnson for their help when I encountered tech difficulties.

Throughout my studies I have enjoyed the camaraderie of present and past fellow graduate students: Stacy Voccia, Xi Chen, Huan Jin, Yuanyang Liu, Bhupesh Shetty, Amin Khezerlou, Fahrettin Cakir, Guanglin Xu, Wenjun Wang, Chuanjie

Liao, Lian Duan, Michael Rechenthin, Amit Verma and Kamal Lamsal. A special thanks to Justin Goodson, who has given me advice and support on many occasions. To Jingyang Zhang, Hongbo Dong, Boshi Yang, Fan Liu, Zhenhao Zhou, Lixi Yu, Lin Tong, Peiwen He, and Yunye Shi, thanks for the companionship and fun you brought to my life. I will always value the wonderful time we spent together.

Last but not the least, I would like to thank my supportive and loving family. I could not go this far without their encouragement and unconditional love. I dedicate this thesis to them.

ABSTRACT

Motivated by the management of sales representatives who visit customers to develop customer relationships, we present a stochastic orienteering problem on a network of queues, in which a hard time window is associated with each customer and the representative may experience uncertain wait time resulting from a queueing process at the customer.

In general, given a list of potential customers and a time horizon consisting of several periods, the sales representative needs to decide which customers to visit in each period and how to visit customers within the period, with an objective to maximize the total reward collected by the end of the horizon. We start our study with a daily orienteering problem, which is a subproblem of the general problem. We focus on developing a priori and dynamic routing strategies for the salesperson to implement during a day.

In the a priori routing case, the salesperson visits customers in a pre-planned order, and we seek to construct a static sequence of customers that maximizes the expected value collected. We consider two types of recourse actions. One is to skip a customer specified by an a priori route if the representative will arrive late in the customer's time window. The other type is to leave a customer immediately after arriving if observing a sufficiently long queue (balking) or to leave after waiting in queue for a period of time without meeting with the customer (reneging). We propose customer-specific decision rules to facilitate the execution of recourse actions and

derive an analytical formula to compute the expected sales from the a priori route. We tailor a variable neighborhood search (VNS) heuristic to find a priori routes.

In the dynamic routing case, the salesperson decides which customer to visit and how long to wait at each customer based on realized events. To seek dynamic routing policies, 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 customer or go to another customer. If departing the current customer, it chooses the customer 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 solutions with recourse actions.

Finally, we address the multi-period orienteering problem. We consider that each customer's likelihood of adopting the representative's product stochastically evolves over time and is not fully observed by the representative. The representative can only estimate the adoption likelihood by meeting with the customer and the estimation may not be accurate. We model the problem as a partially observed Markov decision process with an objective to maximize the expected sales at the end of the horizon. We propose a heuristic that decomposes the problem into an assignment problem to schedule customers for a period and a routing problem to decide how to visit the scheduled customers within the period.

PUBLIC ABSTRACT

We study a problem motivated by routing sales representatives to visit customers for developing customer relationships. We consider that each customer can only be accessed during a time window and the representative may experience uncertain wait time resulting from a queue of others at the customer. We first address a daily routing problem where the representative needs to decide which customers to visit and how to visit the customers within a day. We develop two type of solution approaches for the daily problem. One is an a priori routing strategy, in which the salesperson visits customers in a pre-planned order, and we seek to construct a sequence of customers that maximizes the expected value collected. The other is a dynamic routing strategy, in which the salesperson decides which customer to visit and whether to wait at a customer based on realized information, such as the arrival time and the observed queue length at the customer. We then present a multi-period routing problem, where each customer's likelihood of adopting the representative's product may change over time and is not fully observed by the representative. The representative can only estimate the adoption likelihood by meeting with the customer but the estimation may not be accurate.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

1.1 Research Overview

From manufacturers' perspective, the challenges of higher costs, shrinking margins, and increased customer expectations require them to adopt service-based competitive strategies that could provide a differentiated source of value to customers (Neely, 2014). As a result, service operations are incorporated as a larger component in product offerings than in the past, with businesses shifting from providing just products to product-service bundles (Falk et al., 2012). One way that manufacturers provide service is through a knowledgable sales force who helps match customers with the right products and train customers in the products' use. With this model, manufacturers rely heavily on sales representatives in developing customer relationships, and sales force management becomes very important.

One key aspect in managing sales force is the routing of sales representatives. The market has recognized the value of efficient and effective sales representative routing, as multiple companies supply solutions (e.g., Portatour, Route4Me, TourSolver, eRouteLogistics) to the market. The route planners are able to design daily or weekly schedules for representatives with the consideration of time windows at customers. However, the existing software does not account for the uncertainty that is inherent in the execution of a route plan. For instance, the representative may not be able to meet a customer after arriving within the time window due to waiting. In designing a

tour, however, the software assumes no uncertainty. The software handles unexpected events, such as a customer becoming unavailable due to the representative needing to wait to see the customer, by redesigning the route.

In contrast to the existing software, we explicitly model the uncertainty in developing tours. In doing so, we can anticipate that there will be cases in which representatives will not be able to meet with customers as planned and resultantly build better routes. Specifically, we consider two situations. One is that the representative may have to wait before meeting the customer, even if the representative arrives within the customer's specified time window. Such waiting occurs when the customer is conducting other business (meetings, paperwork, etc.) in addition to talking with the representative. The other is that each customer's likelihood of adopting the product is uncertain and may evolve stochastically over time. Without the representative's visits, each customer's adoption likelihood may be influenced by peers and/or marketing efforts from the representative's competitors. The representative's visits to a customer may increase the customer's adoption likelihood by increasing the customer's awareness and knowledge of the product.

In this thesis, 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. We study the problem in the context of routing pharmaceutical and

textbook sales representatives in this thesis. However, this problem is applicable to several other operations, such as designing trips for tourists who may experience queueing at attractions (Vansteenwegen and Souffriau (2010)) and routing taxis to taxi stands in metropolitan area where there may be queues of competitors (Cheng and Qu (2009)).

In this study, we consider three variants of the problem described above. In §2, we study a daily orienteering problem in the context of routing pharmaceutical sales representatives and develop a priori solutions. The pharmaceutical sales force has experienced a sizable reduction in recent years that is likely to continue. By the end of 2011, the number of pharmaceutical sales jobs in the U.S. dropped to 72,000 from its peak of 105,000 in 2006 (Rockoff, 2012). In 2012, Britain's second largest drugmaker, AstraZeneca, is cutting 1,150 sales representatives in its U.S. organizations, which is 24% of its sales force in the U.S. (Arnold, 2012). Pharmaceutical companies have tried to compensate for the reduced sales force through the use of digital tools like websites and smartphone apps that allow doctors to directly query the company. However, sales representatives are not totally replaceable as they provide the means to develop relationships with customers that may otherwise go unreached. Dealing with a reduced number of sales representatives, pharmaceutical companies face the challenge of better managing their reduced sales forces to maximize their benefit to the company.

On a daily basis, pharmaceutical sales representatives needs to sequence a set of doctors to visit. In general, a doctor will meet representatives in between other

work obligations (seeing patients, paperwork, etc.) during a specified time window. In the problem, we assume a first-come-first-served protocol for the doctor meeting with all sales representatives. The waiting experienced by the representative after arriving at a doctor's office results from a queue of competing sales representatives. We focus on designing an a priori route for the representative that maximizes the expected value collected on a given day. We develop an analytical formula that computes the expected reward from an a priori tour and propose customer-specific recourse action corresponding to the queueing notions of balking and reneging. We solve the problem with a variable neighborhood search heuristic and demonstrate the value of modeling uncertainty by comparing the solutions to our model to that of a deterministic version using expected values of the associated random variables. We also compute an empirical upper bound on our solutions by solving deterministic instances corresponding to perfect information.

In §3, we study the daily orienteering problem in the context of routing textbook sales representatives who plan on visiting professors on campus to promote textbooks. The U.S. 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.

Similar to the pharmaceutical sales routing, professors' availability to textbook sales representatives is in general restricted to time windows (office hours). However, unlike the first-come-first-served queues composed of competing representatives in pharmaceutical sales routing, there is a priority queue at a professor's office consisting of students who have priority in meeting the professor during office hours. That is, the representative must wait as long as there are students waiting at the professor's office, which may mean that the representative's best option is to leave a professor immediately after arriving if observing a long queue and revisit the professor later.

The ability of the salesperson to skip and revisit a professor suggests the development of dynamic solutions that accounts for the realized random events. We model the problem as a Markov decision process (MDP). The state consists of information on the representative's status as well as the status of each professor. To overcome the challenges posed by the curse of dimensionality, we first seek to characterize the structure of the optimal policy. We prove the existence of a control limit on the queue length for a given arrival time under the assumption that the queue length distribution has the increasing failure rate property. We present counter-examples to demonstrate that a control limit policy does not exist on the arrival time for a given queue length. For a given state, we identify conditions under which the representative would not choose to visit certain professors, thus reducing the computational burden in action selection.

Given the restrictive assumptions required to characterize the optimal policy,

we focus on developing an approximate dynamic programming scheme to obtain dynamic routing policies to maximize the total expected sales. We propose a two-stage heuristic that we refer to as compound rollout for this problem. 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 solutions with recourse actions.

In §4, we introduce the multi-period orienteering problem where each customer's likelihood of adopting the product is uncertain. The representative visits customers over a multi-period horizon to observe and potentially increase the chance of customer adoption. However, when meeting with a customer, the representative may not be able to fully observe the customer's adoption likelihood. Instead, she estimates the underlying likelihood based on the information obtained via communicating with the customer, but the estimation may not be accurate. We consider a time window and a queueing process associated with each customer. The decisions faced by the representative are which customers to schedule in each period and in which order to visit them, knowing that due to time windows and uncertain wait times the representative may not be able to meet customers even if they are on the schedule. We propose two models for the problem. The first is a Markov decision process (MDP) in which we assume that the representative can obtain perfect information of the customers' adoption likelihood by meeting with the customers. The other is a partially observed Markov decision process (POMDP) which accounts for

the partial observable nature of customers' adoption likelihood. The objective of both the MDP and POMDP is to maximize the expected sales by the end of the horizon. We propose a heuristic approach to solve the problem. The heuristic iteratively solves an assignment problem and a routing problem to determine which customers to visit in a period and how to visit the scheduled customers within the period, respectively.

We conclude the thesis by summarizing the contributions in §5.

1.2 Literature Review

In this section, we review related literature to position our contributions. In §1.2.1, we investigate the literature of traveling salesman problem (TSP) with profits, specifically the stochastic TSP with profits and the team orienteering problem. In §1.2.2, we relate our work to the queueing literature in which queueing decision rules are investigated. In §1.2.3, we examine the literature that use approximate dynamic programming to solve routing problems, which is similar to our work in §3.

1.2.1 Orienteering Problem

Our study is rooted in the TSP with profits. The notion of profit or value can be incorporated in a TSP in several ways and the problem titling typically varies depending on the treatment of profit. One way to model profit is to minimize the overall travel costs in the objective function while using a constraint to ensure that a certain amount of profit is collected (prize collecting TSP). Another way is to maximize the overall profit collected, while limiting the total travel cost to a specific upper bound (orienteering problem or selective TSP). The third way minimizes the

travel costs minus collected profit in the objective function (profitable tour problem). Two extensions of the orienteering problem, usually referred to as the team orienteering problem and the selective vehicle routing problem, plan more than one tour and each tour collects rewards during the same period of time. Feillet et al. (2005) provides a survey of the deterministic TSPs with profits and Vansteenwegen et al. (2011) conducts a survey of the orienteering problem.

Stochastic elements have been incorporated in TSPs with profits in several ways. Teng et al. (2004) study a time-constrained TSP with uncertain travel and service time whose objective is to maximize the total profit collected on a tour. A limit is set to the total travel and service time of a route. They formulate the problem as a two-stage stochastic problem with recourse, where in the first stage a subset of customers are sequenced before the travel and service time are realized and in the second stage an expected penalty is imposed on the objective function for the violation of the limit time. Tang and Miller-Hooks (2005) propose a selective TSP with stochastic service times, travel times and travel costs. They formulate the problem as a chance-constrained stochastic integer program with an objective to maximize the total profit collected from an a priori tour while restricting the probability that the tour length exceeds a certain threshold to a given value. Campbell et al. (2011) address an orienteering problem with stochastic travel and service times where a customer specific penalty is incurred for each scheduled customer not reached before a given known deadline. They investigate special versions of the problem that can be solved optimally and present variable neighborhood search heuristics to solve general

versions of the problem. Similar to Campbell et al. (2011), Papapanagiotou et al. (2013) investigate an orienteering problem with stochastic travel and service times and a given deadline. They focus on developing a Monte Carlo sampling procedure to approximate the objective function. They demonstrate the effectiveness of the objective approximation by comparing it with the exact objective evaluation. Evers et al. (2014) study an orienteering problem with a stochastic weight on each arc and a hard constraint on the total weight of a tour. They introduce a two-stage recourse model where a recourse action of aborting the tour (returning to the depot) is taken in the second stage. They propose a sample average approximation (SAA) procedure to solve the problem, in which the objective function is approximated via Monte Carlo sampling. Verbeeck et al. (2015) study a stochastic time-dependent orienteering problem with time windows where the travel time between vertices is stochastic and depends on the departure time at the first vertex. They propose an ant colony system to develop a priori solutions.

The stochastic elements in the studies we describe in the preceding paragraph are similar to our stochastic orienteering problem in that the arrival time at a customer is uncertain (in our problem the arrival time at a customer is uncertain due to the uncertain waiting time at previous customers). However, only Verbeeck et al. (2015) addresses the stochastic orienteering problem with hard time windows. Voccia et al. (2013) conduct a study of a probabilistic traveling salesman problem with time windows (PTSPTW). Though inducing uncertain arrival time via uncertain presence of customers, this PTSPTW is not comparative to our problem as the former's ob-

jective is to minimize the expected traveling costs and does not consider any measure of profit. As far as we are aware, no previous studies on orienteering has considered the uncertainty in adoption likelihood as we do in §4.

The multi-period orienteering problem we consider is related to the team orienteering problem with time windows (TOPTW). In the TOPTW, time windows are associated with each point and a score can only be collected if the competitor arrives before the closure of the time window. The deterministic team orienteering problem (TOP) is first introduced by Chao et al. (1996), where a team of competitors are routed to visit several points to collect scores within a prescribed time limit. The score associated with each point can only be collected once. The objective is to design multiple routes, each for a team member, such that the total score collected can be maximized. Recent studies in the TOP consider time windows at customers (Souffriau et al. (2013), Hu and Lim (2014), Labadie et al. (2011), Labadie et al. (2012), and Montemanni and Gambardella (2009)).

Similar to the TOPTW, we design several routes in the problem presented in §4. However, rather than developing one route for each competitor in the same period, we design one route for each period over the multi-period time horizon. Our problem can be categorized as a multi-period orienteering problem with time windows (MuPOPTW). Tricoire et al. (2010) study a deterministic multi-period orienteering problem with multiple time windows, where each customer may have up to two time windows on each day and can only be visited once during the horizon. Different from the above TOPTWs and the MuPOPTW that are deterministic, the problem

we consider is a stochastic MuPOPTW, where the stochasticity is induced by the uncertain adoption likelihood associated with customers and the uncertain wait time at customers. In addition, existing literature on the TOPTW and MuPOPTW does not consider multiple visits to a customer in different periods.

1.2.2 Queueing Theory

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 our problem, 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 queueing mechanism in a routing problem as we do.

1.2.3 Approximate Dynamic Programming

In §3 of the thesis, we propose a approximate dynamic programming approach to solve the stochastic orienteering problem on a network of queues. 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. (2015)). 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. (2015) develop restocking-based rollout policies that explicitly consider preemptive capacity replenishment. The compound rollout algorithm we propose for the problem in §3 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.

CHAPTER 2
A PRIORI ORIENTEERING WITH TIME WINDOWS AND STOCHASTIC WAIT TIMES AT CUSTOMERS

2.1 Introduction

In this chapter, we study a variant of the problem motivated by the routing of pharmaceutical sales representatives. On a daily basis, pharmaceutical sales representatives must determine a daily schedule specifying which doctors to visit and the order to visit them in order to maximize potential sales. In general, a doctor's accessibility to a sales representative is limited. Typically, the doctor specifies a time window during which she will meet representatives in between other work obligations (seeing patients, paperwork, etc.). We assume that the doctor meets with the representatives following a first-come-first-served protocol within the time window. To talk with a doctor, a representative must arrive at the doctor's office before the closure of the doctor's time window. However, after arriving, the representative may have to wait while the doctor attends to work obligations or meets with another representative. The time that doctors spend in with other work obligations is uncertain, while the duration of a meeting between a doctor and a representative is typically limited. In fact, representatives develop well-rehearsed sales pitches that fit within their brief time allocation. Thus, we assume that the meeting duration is known and constant, while the time that a representative spends waiting to meet with a doctor is uncertain. This daily decision-making problem faced by pharmaceutical sales representatives is shared by other industries such as the textbook industry. Therefore, we more gen-

erally refer to this problem as a stochastic orienteering problem with time windows (SOPTW), in which a time window is associated with each customer and a uncertain wait time occurs at each customer.

We model the wait time faced by the representative as a random variable dependent on the arrival time and the queue length (of other representatives) upon arrival. In turn, we model queue length as a random variable dependent on the arrival time at the doctor's office. We note that the queue at each doctor is a non-priority queue where each representative has the same priority in the queue and a first-come-first-served rule applies. We associate a deterministic economic value with each doctor representing the expected sales to the doctor if the representative meets with the doctor that day. In this study, our objective is to construct an a priori route for the representative that maximizes the expected value collected on a given day. An a priori route is a pre-planned order of visits to customers (see Campbell and Thomas (2008a) for an overview). That is, we seek to construct a static sequence of customers (doctors) that maximizes the expected value collected. A priori tours are usually used as subproblems in dynamic solution approaches (see Goodson et al. (2013)). In a dynamic solution, decisions are made based on realized random information. We discuss dynamic solution strategies in §3.

During the execution of an a priori route, after a random event is realized, a corrective action, often called a recourse action, may be required. In our treatment of the SOPTW, we consider two types of recourse actions, both of which preserve the route's ordering of the visited customers. The first recourse action is motivated by

the observation that a representative should skip a customer specified by an a priori route if she will not arrive within the customer's time window. With the assumption of deterministic travel times, the representative can determine whether or not she can arrive at the next customer on the route within the respective time window given the departure time from the current customer. That is, the next customer visited by the representative will be the first customer in the remaining sequence of customers that can be visited within the time window. In §2.2, we generalize this recourse action to allow skipping if a representative will not arrive by any specified time.

The second recourse action operationalizes the notions of balking and reneging from queueing theory. The basic premise is that the representative may leave the queue immediately after arriving at a customer if observing a sufficiently long queue (balking) or she may leave after waiting in queue for a period of time without meeting with the customer (reneging). We propose a decision rule that takes as input the observed queue length upon the representative's arrival to a customer to determine the latest time she will wait there. We elaborate on this second recourse action in §2.2.

This study makes four primary contributions to the literature. First is the formulation of a routing model that explicitly accounts for the stochastic wait times resulting from queues of competing salespeople. Second, the customer-specific recourse action corresponding to the queueing notions of balking and reneging is a novel addition to routing models. We also derive an analytical formula to compute the expected reward from an a priori tour. Finally, our computational results demon-

strate the value of incorporating stochastic information into the routing model for sales representatives.

In §2.2 and §2.3, we provide a formal model of the problem and derive a closed-form computation of the objective. We discuss our solution approach to the SOPTW in §2.4. In §2.5, we discuss the generation of input data, notably the data for the transient queueing, for our SOPTW instances. §2.6 presents a deterministic version of the SOPTW in which waiting time distributions are replaced with expected wait times, resulting in an elementary longest path problem solvable via an exact solution method. §2.7 provides a computational comparison and §2.8 concludes with a summary and discussion of future work.

2.2 Problem Description

We define the SOPTW on a complete graph $G = (V, E)$. The set V consists of $n + 1$ nodes corresponding to a depot and n potential customers that a pharmaceutical representative may visit. The set E denotes the set of edges that is associated with each pair of vertices. A deterministic travel time, denoted c_{ij}, is associated with each edge (i, j). A profit r_i and a time window $[e_i, l_i]$ are associated with each customer $i \in V$. Let A_i be a random variable representing the arrival time at customer i. Let Q_i be a random variable representing the queue length at customer i observed upon the representative's arrival and q_i be a realization of Q_i. We assume the distribution of Q_i to be a function of a_i, denoted with the probability mass function $g(q_i|a_i)$, where a_i is a realization of A_i. Let W_i be a random variable representing the wait

time at customer i and w_i be a realization of W_i. We assume the distribution of W_i to be a function of q_i and a_i, denoted with the probability mass function $f(w_i|q_i, a_i)$. We note that the possible values of W_i range from $\max\{e_i - a_i, 0\}$ to $l_i - a_i$. Let S_i be a random variable with a known distribution representing the duration of the meeting between the sales representative and the customer. While our model can easily incorporate an uncertain amount of meeting time (as our analytical derivation in §2.3 shows), in our computational results of §2.7.2 we assume $S_i = s$ where s is a known and constant value. This assumption is based on the observation that meeting durations for pharmaceutical sales representatives are often short and pre-determined by the doctor.

As stated in the introduction, we consider two types of recourse actions in the model for the SOPTW, both of which incorporate the value of customers. The first recourse action is to skip a planned customer j on the a priori tour if the representative cannot arrive before a specified time τ_j. That is, if $a_j > \tau_j$ then the representative will skip customer j and consider the next customer on the a priori tour. Let D_i be a random variable representing the departure time from customer i. Suppose customer i is followed by customer j on the a priori tour. Once we know the realization of D_i, d_i, the arrival time at j can be computed as $a_j = d_i + c_{ij}$. The representative will skip customer j if $a_j = d_i + c_{ij} > \tau_j$, where $\tau_j \in [0, l_j]$.

The value of τ_i for customer i is determined by the value of the parameter $\tilde{r}$, the minimum expected reward for which the representative is willing to travel to

customer i, where $\tilde{r} \in \left[0, \min_i\{r_i\}\right]$. We compute τ_i as follows:

$$\tau_i = \max\left\{t \geq 0 : P\left(W_i < l_i - t | a_i = t\right) \times r_i \geq \tilde{r}, t \in \mathit{support}(W)\right\}, \tag{2.1}$$

where we obtain the probability measure from $g(q_i|a_i)$ and $f(w_i|q_i, a_i)$ via the law of total probability. Intuitively, τ_i corresponds to a threshold value after which time the likelihood that the representative will meet with the customer i is deemed too low.

The second recourse action enables the representative either to leave the queue immediately after arriving and observing a sufficiently long queue (balking) or to leave after waiting in the queue for a period of time without having met the doctor (reneging). In the following, we establish a static decision rule to determine how long the representative should wait at a customer upon arriving.

Let the random variable Δ_i represent the longest amount of time that the representative will wait at customer i. Let $\check{r}$ be a parameter indicating the minimum expected reward for which the representative is willing to wait at customer i, where $\check{r} \in \left[0, \min_i\{r_i\}\right]$. We then specify Δ_i such that the realized value of Δ_i, denoted by δ_i, is known with certainty given $\check{r}$, the arrival time a_i at customer i, and the observed queue length q_i at customer i. Specifically,

$$\delta_i = \max\left\{t \geq 0 : P\left(W_i < l_i - a_i | q_i, w_i \geq t\right) \times r_i \geq \check{r}, t \in \mathit{support}(W)\right\}, \tag{2.2}$$

where the probability measure is based on $f(w_i|q_i, a_i)$ and $\mathit{support}(W)$ denotes the support of W. Intuitively, δ_i establishes a threshold value after which the probability of meeting with the customer is considered too low to be valuable. If no value of t satisfies Equation (2.2), we define $\delta_i = 0$; this case corresponds to the representative

balking and immediately leaving customer i upon arrival.

Let $\Gamma_i = A_i + \Delta_i$ be the latest time that the representative will stay at customer i without having started a conversation with the customer. Let γ_i be the realization of Γ_i. Given the arrival time and observed queue length at customer i, the value of Γ_i is fully specified, $\gamma_i = a_i + \delta_i$. If there is no possibility of visiting a future customer for the representative who is supposed to leave customer i at γ_i, then the representative will stay at customer i until the end of customer i's time window, i.e., $\gamma_i = l_i$ if customer i is the last unskipped customer in the sequence.

As an example, consider a case in which the representative arrives at customer i at time 10 and observes a queue length of four. Let customer i's time window be $[e_i = 8,\ l_i = 28]$ and economic value be 30. Further, suppose the distribution of $f(w_i|q_i = 4, a_i = 10)$ is defined by the values 8, 12, 16, 18 with probabilities 0.2, 0.3, 0.4, 0.1, respectively. Then, define the distribution $f(w_i|q_i = 4, a_i = 10, w_i \geq 12)$ by the values of 12, 16 and 18 with probability 0.375, 0.5, 0.125, respectively. Define the distribution $f(w_i|q_i = 4, a_i = 10, w_i \geq 16)$ by the values of 16 and 18 with probabilities 0.8 and 0.2, respectively. The distribution of $f(w_i|q_i = 4, a_i = 10, w_i \geq 18)$ is defined by the value of 18 with probability 1. For a minimum expected reward $\check{r} = 15$, $\delta_i = 16$ from Equation (2.2), and consequently $\gamma_i = 26$.

Our objective is to find an a priori tour that, armed with the two recourse actions described above, maximizes the expected revenue. To express this objective, let B_i be a random variable denoting the time that the representative starts a meeting with customer i. Due to the uncertainties in this problem, it is unlikely to be able

to visit all customers; we let $B_i = \infty$ denote that customer i was not visited. Let Z_i be a random variable representing the reward collected from customer i. We assume that a deterministic reward of $r_i > 0$ is collected from customer i if and only if a representative starts a meeting with customer i no later than γ_i. In other words, $Z_i = r_i$ if $b_i \leq \gamma_i$ and 0 otherwise, where b_i is a realization of B_i. The expected revenue for a priori tour v is $\Omega(v) = \sum_{i \in v} r_i P[B_i \leq \Gamma_i]$. The objective of this problem is to find an a priori tour v^* such that $\Omega(v^*) \geq \Omega(v)$ for every v.

2.3 Objective Computation

To compute the expected revenue of an a priori tour, we must compute $P[B_i \leq \Gamma_i]$ for each customer i. In the following derivation, the index i refers to the ith customer in tour v. To develop a recursion for computing these values of $P[B_i \leq \Gamma_i]$, we begin with the observation that

$$\begin{aligned} P[B_i \leq \Gamma_i] &= P[A_i + W_i \leq \Gamma_i] \\ &= \sum_{a_i} \sum_{q_i} \sum_{w_i} P[W_i \leq \Gamma_i - a_i | A_i = a_i, Q_i = q_i, W_i = w_i] \\ &\times P[W_i = w_i | Q_i = q_i, A_i = a_i] P[Q_i = q_i | A_i = a_i] P[A_i = a_i] \\ &= \sum_{a_i} \sum_{q_i} \sum_{w_i \leq \gamma_i - a_i} \Lambda(i, \gamma_i) f(w_i | q_i, a_i) g(q_i | a_i) P[A_i = a_i]. \end{aligned} \tag{2.3}$$

where the first equality follows from the definition of B_i and the second equality results from conditioning on the arrival time, queue length and wait time. The third equality results from several observations. First, the values for $P[Q_i = q_i | A_i = a_i]$ and $P[W_i = w_i | Q_i = q_i, A_i = a_i]$ are provided by the known probability mass distributions $g(q_i|a_i)$ and $f(w_i|q_i, a_i)$, respectively. Second, once the realizations of A_i and Q_i are

known, the realization of Γ_i is determined, so $P[W_i \leq \Gamma_i - a_i | A_i = a_i, Q_i = q_i, W_i = w_i] = \Lambda(i, \gamma_i)$, where $\Lambda(i,t)$ is an indicator function such that

$$\Lambda(i,t) = \begin{cases} 1, & \text{if } a_i + w_i \leq t \\ 0, & \text{otherwise.} \end{cases}$$

Third, because for a particular arrival time a_i and arrival queue length q_i, the representative will leave customer i at time γ_i if she cannot start a meeting with the customer before γ_i, we know that the wait time w_i for the representative at the customer cannot exceed $\gamma_i - a_i$.

In Equation (2.3), we need to determine $P[A_i = a_i]$. Define the random variable

$$X_i = \begin{cases} 1, & \text{if customer i is not skipped} \\ 0, & \text{if customer i is skipped.} \end{cases}$$

Because customer i will be skipped if the arrival time at the customer will be τ_i or later, we have $P[X_i = 1] = P[A_i \leq \tau_i]$ and $P[X_i = 0] = P[A_i > \tau_i]$. We express the

arrival time at customer i as:

$$\begin{aligned}
P[A_i = a_i] &= \sum_{j=0}^{i-1} P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}] P[X_j = 1, D_j = a_i - c_{ji}] \\
&= \sum_{j=0}^{i-1} P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}] P[D_j = a_i - c_{ji} | X_j = 1] P[X_j = 1] \\
&= \sum_{j=0}^{i-2} P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0, \ldots, X_{i-1} = 0] \\
&\quad \times P[X_{j+1} = 0, \ldots, X_{i-1} = 0 | X_j = 1, D_j = a_i - c_{ji}] \\
&\quad \times P[D_j = a_i - c_{ji} | X_j = 1] P[X_j = 1] \\
&\quad + (P[A_i = a_i | X_{i-1} = 1, D_{i-1} = a_i - c_{i-1,i}] \\
&\quad \times P[D_{i-1} = a_i - c_{i-1,i} | X_{i-1} = 1] P[X_{i-1} = 1]) . \qquad (2.4)
\end{aligned}$$

The first equality holds because the arrival time at customer i depends on the departure time from the customer that is visited before customer i. The second equality follows from the definition of joint probability. The third equality follows from the observation that arc (j, i) appears on the tour only if customers $j+1, j+2, \ldots, i-1$ are skipped.

Notably, $P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0, \ldots, X_{i-1} = 0] = 1$ if $a_i \leq \tau_i$ as we would not skip customer i and $P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0, \ldots, X_{i-1} = 0] = 0$ if $a_i > \tau_i$ as we would skip customer i. By defining the indicator function

$$\Psi(i, t) = \begin{cases} 0, & \text{if } t > \tau_i \\ 1, & \text{otherwise,} \end{cases}$$

we express $P[A_i = a_i | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0, \ldots, X_{i-1} = 0] = \Psi(i, a_i)$. Hence, to determine $P[A_i = a_i]$ via Equation (2.4) it remains to compute $P[X_{j+1} =$

$0, \ldots, X_{i-1} = 0 | X_j = 1, D_j = a_i - c_{ji}]$, $P[D_j = a_i - c_{ji} | X_j = 1]$, and $P[X_j = 1]$ for $j = 0, \ldots i-1$.

For notational convenience, let $R(j, i-1, a_i - c_{ji}) = P[X_{j+1} = 0, \ldots, X_{i-1} = 0 | X_j = 1, D_j = a_i - c_{ji}]$ for $j = 0, \ldots i-2$. By consecutively conditioning on $X_{j+1}, X_{j+2}, \ldots, X_{i-1}$, we obtain

$$\begin{aligned} R(j, i-1, a_i - c_{ji}) &= P[X_{j+2} = 0, \ldots, X_{i-1} = 0 | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0] \\ &\quad \times P[X_{j+1} = 0 | X_j = 1, D_j = a_i - c_{ji}] \\ &= P[X_{j+2} = 0, \ldots, X_{i-1} = 0 | X_j = 1, D_j = a_i - c_{ji}, X_{j+1} = 0] \\ &\quad \times [1 - \Psi(j+1, a_i - c_{ji} + c_{j,j+1})] \\ &= \ldots \\ &= \prod_{k=j+1}^{i-1} (1 - \Psi(k, a_i - c_{ji} + c_{jk})). \end{aligned} \tag{2.5}$$

From Equation (2.5), we see that given possible realizations of A_i, we can compute $R(j, i-1, a_i - c_{ji})$ iteratively.

Next, we calculate $P[D_j = a_i - c_{ji} | X_j = 1]$ via conditioning:

$$\begin{aligned} &P[D_j = a_i - c_{ji} | X_j = 1] \\ &= \sum_{a_j} \sum_{q_j} (P[D_j = a_i - c_{ji} | X_j = 1, A_j = a_j, Q_j = q_j] \times P[Q_j = q_j | A_j = a_j, X_j = 1] \\ &\times P[A_j = a_j | X_j = 1]) \\ &= \sum_{a_j} \sum_{q_j} \sum_{w_j \le \gamma_j - a_j} \sum_{s_j} P[A_j + W_j + S_j = a_i - c_{ji} | X_j = 1, A_j = a_j, Q_j = q_j, W_j = w_j, S_j = s_j] \\ &\times P[S_j = s_j | X_j = 1, A_j = a_j, Q_j = q_j, W_j = w_j] P[W_j = w_j | X_j = 1, Q_j = q_j, A_j = a_j] \\ &\times P[Q_j = q_j | X_j = 1, A_j = a_j] P[A_j = a_j | X_j = 1] \\ &+ \sum_{a_j} \sum_{q_j} \sum_{w_j > \gamma_j - a_j} P[\Gamma_j = a_i - c_{ji} | X_j = 1, A_j = a_j, Q_j = q_j, W_j = w_j] \\ &\times P[W_j = w_j | X_j = 1, Q_j = q_j, A_j = a_j] P[Q_j = q_j | X_j = 1, A_j = a_j] P[A_j = a_j | X_j = 1]. \end{aligned} \tag{2.6}$$

The first inequality follows from conditioning on the arrival time and queue length. To achieve the second equality results, we condition on the wait time W_j and service time S_j for $w_j \leq \gamma_j - a_j$ (representative meets with the customer), and we condition on just the wait time W_j for $w_j > \gamma_j - a_j$ (representative departs the customer without meeting).

In Equation (2.6), $P[A_j + W_j + S_j = a_i - c_{ji} | X_j = 1, A_j = a_j, Q_j = q_j, W_j = w_j, S_j = s_j]$ is either 0 or 1. We assume random variable S_i is independent of random variables A_i, Q_i, W_i, and X_i, and thus $P[S_j = s_j | X_j = 1, A_j = a_j, W_j = w_j, Q_j = q_j] = P[S_j = s_j]$; this quantity is specified by the known distribution of S_j. The known distributions $f(w_i|q_i, a_i)$ and $g(q_j|a_j)$ specify the probabilities $P[W_j = w_j | X_j = 1, Q_j = q_j, A_j = a_j]$ and $P[Q_j = q_j | X_j = 1, A_j = a_j]$, respectively. Because the value of Γ_j is known given realizations of arrival time and queue length, the probabilities $P[\Gamma_j = a_i - c_{ji} | X_j = 1, A_j = a_j, Q_j = q_j, W_j = w_j]$ are 0 or 1. Finally,

$$P[A_j = a_j | X_j = 1] = \frac{P[X_j = 1 | A_j = a_j] P[A_j = a_j]}{P[X_j = 1]} = \frac{\Psi(j, a_j) P[A_j = a_j]}{P[X_j = 1]}. \quad (2.7)$$

Let $\Theta(j, a_i - c_{ji}) = P[D_j = a_i - c_{ji} | X_j = 1] P[X_j = 1]$. Equations (2.6) and (2.7) imply

$$\begin{aligned}
&\Theta(j, a_i - c_{ji}) \\
&= \sum_{a_j} \sum_{q_j} \sum_{w_j \leq \gamma_j - a_j} \sum_{s_j} I_{\{a_j + w_j + s_j = a_i - c_{ji}\}} P[S_j = s_j] f(w_j | q_j, a_j) g(q_j | a_j) \Psi(j, a_j) \\
&\times P[A_j = a_j] \\
&+ \sum_{a_j} \sum_{q_j} \sum_{w_j > \gamma_j - a_j} I_{\{\gamma_j = a_i - c_{ji}\}} f(w_j | q_j, a_j) g(q_j | a_j) \Psi(j, a_j) P[A_j = a_j], \quad (2.8)
\end{aligned}$$

where $I_{\{x\}} = 1$ if condition x is true and 0 otherwise.

Using Equations (2.5), (2.6) and (2.8), we restate Equation (2.4) as

$$P[A_i = a_i] = \Psi(i, a_i) \left(\sum_{j=0}^{i-2} R(j, i-1, a_i - c_{ji}) \Theta(j, a_i - c_{ji}) + \Theta(i-1, a_i - c_{i-1,i}) \right). \tag{2.9}$$

Using Equations (2.3) and (2.9), we express the expected revenue of an priori tour with n customers as

$$\begin{aligned} \sum_{i=1}^{n} r_i P[B_i \leq \Gamma_i] = \sum_{i=1}^{n} \sum_{a_i} \sum_{q_i} \sum_{w_i}^{\gamma_i - a_i} & r_i \Lambda(i, \gamma_i) f(w_i | q_i, a_i) g(q_i | a_i) \Psi(i, a_i) \\ & \times \left(\sum_{j=0}^{i-2} R(j, i-1, a_i - c_{ji}) \Theta(j, a_i - c_{ji}) + \Theta(i-1, a_i - c_{i-1,i}) \right). \end{aligned} \tag{2.10}$$

To determine the computational complexity of Equation (2.10), we first define the cardinality of the sets involved. Because customer i is skipped if $a_i > \tau_i$, there are at most τ_i possible arrival time realizations and we define $n_a = \max_i\{\tau_i\}$. Let n_{s_i} be the number of possible service time realizations at customer i based on s_i's distribution, and define $n_s = \max_i\{n_{s_i}\}$. Let $n_{q_i|a_i}$ be the number of possible queue length realizations given the arrival time a_i at customer i based on $g(q_i|a_i)$ and define $n_q = \max_i\{\max_{a_i}\{n_{q_i|a_i}\}\}$. As the value γ_i is known given a_i and q_i, we let $n_{w_i|a_i,q_i}$ be the number of possible wait time realizations satisfying $w_i \leq \gamma_i - a_i$, given the arrival time a_i and queue length q_i. Then, we define $n_w = \max_i\{\max_{q_i,a_i}\{n_{w_i|a_i,q_i}\}\}$. Similarly, we let $n'_{w_i|a_i,q_i}$ be the number of possible wait time realizations satisfying $w_i > \gamma_i - a_i$, given the arrival time a_i and queue length q_i. Then, we define $n'_w = \max_i\{\max_{q_i,a_i}\{n'_{w_i|a_i,q_i}\}\}$ and $n_{ws} = \max\{n_w n_s, n'_w\}$.

The complexity of computing $\Theta(j, a_i - c_{ji})$ with Equation (2.8) is $O(n_a n_q n_{ws})$. The complexity of computing $R(j, i-1, a_i - c_{ji})$ with Equation (2.5) is $O(n)$. Therefore, the complexity of computing the expected revenue of an a priori tour by Equation (2.10) is $O(n^2 n_a^2 n_q^2 n_w n_{ws})$, assuming $n < n_a n_q n_{ws}$ (which would be typical in a stochastic problem). In our computational study, we demonstrate the computational cost of exact evaluation of a an priori tour via Equation (2.10). Further, we show that we can reduce the computational burden of repeatedly evaluating the objective function in our search algorithm by utilizing sampled wait time and queue lengths to estimated the expected revenue of an a priori tour (with little loss of accuracy versus the exact evaluation).

2.4 Solution Approach for the SOPTW

We apply a variable neighborhood search (VNS) heuristic to solve the SOPTW. For an overview of the VNS, see Hansen and Mladenovic (2003). Algorithm 2.1 outlines our VNS implementation, which is a variant of the algorithm used by Campbell et al. (2011) to solve an orienteering problem with stochastic travel and service times. Though having a different problem in terms of constraints and recourse actions, we are seeking a permutation of customers as is done in Campbell et al. (2011). The feasibility of tours in our problem is maintained by not allowing the salesperson to visit a customer outside the customer's time window. VNS has been proven to be effective in finding near-optimal a priori tours in other routing problems (Campbell et al. (2011) and Voccia et al. (2013)).

Algorithm 2.1 Variable Neighborhood Search (VNS) for the SOPTW

1: **Input:** Data for an SOPTW instance, an ordered set of neighborhoods N
2: **Output:** A SOPTW solution v
3: $v \leftarrow initialize()$
4: $i \leftarrow 0, k \leftarrow 1, level \leftarrow 0$
5: **while** $i < iterationMax$ or $level < levelMax$ **do**
6: $\quad v' \leftarrow Shake(v, k)$
7: $\quad v'' \leftarrow VND(v')$
8: $\quad$ **if** $f(v'') > f(v)$ **then**
9: $\quad\quad v \leftarrow v'', level \leftarrow 0$
10: $\quad$ **else if** $k = |N|$ **then**
11: $\quad\quad k \leftarrow 1, level \leftarrow level + 1$
12: $\quad$ **else**
13: $\quad\quad k \leftarrow k + 1, level \leftarrow level + 1$
14: $\quad$ **end if**
15: $\quad i \leftarrow i + 1$
16: **end while**

Line 3 of Algorithm 2.1 initializes the search with a randomly-generated SOPTW solution. Line 5 states the termination criteria for the VNS. In addition to an iteration limit, the variable *level* is used to ensure that the algorithm performs at least $levelMax$ iterations after finding an improved solution. We determine that setting $iterationMax = 200$ and $levelMax = 20$ results in robust performance. The *Shake* procedure in Line 6 returns v', a randomly selected neighbor of solution v from a neighborhood $k \in N$. Line 7 then applies a variable neighborhood descent (VND) procedure to this "shaken" solution v' to obtain a locally-optimal solution v''. Line 8 compares the newly-found local optimal solution to the incumbent solution using the function f, which evaluates a solution using Equation (2.10) in §2.3 to determine the

expected reward of a SOPTW solution. Lines 9 - 15 manage the updating of the current solution, the active neighborhood, and the iteration counters.

For the *Shake* procedure, we consider a set N that consists of three neighborhoods. The first and the second are 1-shift and 2-opt neighborhoods, respectively. The third neighborhood is based on the ruin-and-recreate principle (Schrimpf et al., 2000). As Campbell et al. (2011), we implement ruin and recreate by removing $\min\{n, \lfloor ni/10 \rfloor\}$ customers from a tour and randomly inserting them back into the tour, where n is the number of customers in the tour and i is the current iteration in the VNS algorithm.

Algorithm 2.2 describes the VND procedure we apply in Line 7 of Algorithm 2.1. The VND finds a locally-optimal solution relative to a set of specified neighborhoods N'. The 1-shift and 2-opt neighborhoods compose N' in our study. The *BestNeighbor* procedure in Line 8 returns the best solution in the neighborhood j of the current solution v. For computational efficiency, instead of evaluating solutions using Equation (2.10) in §2.3 (as function f does in the Algorithm 2.1), Line 6 of Algorithm 2.2 evaluates solutions by sampling queue lengths and wait times to estimate the economic value collected from a tour (Papapanagiotou et al. (2013), Evers et al. (2014)). Lines 9 - 13 manage the updating of the locally optimal solution, the active neighborhood, and the variable *count* which ensures that all neighborhoods are explored from the current local optimum before terminating.

Algorithm 2.2 Variable Neighborhood Descent (VND) for the SOPTW

Input: A SOPTW solution v, an ordered set of neighborhoods N'
Output: A locally optimal solution SOPTW v
$j \leftarrow 1, count \leftarrow 1, improving \leftarrow true$
while *improving* **do**
  $v' \leftarrow BestNeighbor(v, j)$
  if $g(v') > g(v)$ **then**
    $v \leftarrow v', count \leftarrow 1$
  else if $j = |N'|$ and $count < |N'|$ **then**
    $j \leftarrow 1, count \leftarrow count + 1$
  else if $j < |N'|$ and $count < |N'|$ **then**
    $j \leftarrow j + 1, count \leftarrow count + 1$
  else
    $improving \leftarrow false$
  end if
end while

2.5 Data Generation

In this section, we discuss the generation of our data sets. In the absence of detailed industry data, we use a discrete-event simulation to generate queue length and wait time distributions. We also describe how we modify existing benchmark problems to suit our SOPTW.

2.5.1 Distribution Generation through Simulation

We develop a discrete-event simulation using the Arena 14.00.00 software package to generate data to construct the transient queue length and wait time distributions. Because of the relatively short interval of time in which representatives work, steady-state distributions are not applicable. While we were unable to identify an industry

partner who collected the detailed data required by our model, interviews with an anonymous industry expert guided our choice of parameters.

In the simulation, salespeople arrive at a doctor with exponentially distributed inter-arrival times with a mean of 15 minutes. Sales representatives are aware that the effectiveness of their visit may degrade if it occurs at the end of a long sequence of sales calls as the doctor experiences "representative fatigue". Consequently, in our data generation we assume that a competing salesperson will balk after arriving if there are five or more other salespeople already waiting in the queue. Otherwise, the competing salesperson will join the queue. As is mentioned by our industry partner, a representative usually considers reneging after she has been waiting for 30 minutes but meet the doctor. To capture this behavior, we assume that a competing salesperson j will renege if she has waited u_j minutes and is not first in the queue, where u_j is a uniformly distributed random variable between 30 and 40. Our sales representative, the decision-maker in our model, does not follow these industry-based rules-of-thumb, but rather implements the recourse actions discussed in §2.2 to improve performance. We set the service time to a constant of 10 minutes as salespeople tend to have brief, rehearsed sales pitches. Based on our interviews with an industry expert, we set the amount of time between a doctor finishing a meeting with one representative and starting with another salesperson as uniformly distributed between 0 and 60 minutes. This period accounts for the time that the doctor spends with other job-related tasks in between meetings with sales representatives.

We execute the discrete-event simulation over five hours of simulated time.

Salespeople arrive throughout the five-hour run, but the doctor does not begin processing arrivals until after the first hour. This structure mimics the situation in which salespeople arrive early (often waiting in the clinic parking lot or lobby). The four-hour block during which the doctor meets with salespeople corresponds to the widest time window we consider in our data sets and is based on the observation that doctors often allow sales visits in either a four-hour block in the morning or afternoon.

For each arrival, we record the arrival time and the queue length observed upon arrival. For each salesperson meets with the doctor, we record the wait time that preceded the meeting. To generate the distribution of queue length, we compute the relative frequencies of queue lengths corresponding to each five-minute arrival time interval over the simulation run. The queue length distribution is used together with each doctor's time window information to construct empirical queue length distributions given arrival time, $g(q_i|a_i)$.

Similarly, we compute the relative frequency of wait times (expressed in five-minute intervals) corresponding to each possible queue length to generate the distribution of wait times. We use the wait time distribution with customers' time window information to construct the empirical wait time distributions given queue length and arrival time, $f(w_i|q_i, a_i)$ for each doctor.

The data for these queue length and wait time distributions are available via the University of Iowa's digital repository at http://ir.uiowa.edu/tippie_pubs/61/.

2.5.2 Dataset Generation

We derive our SOPTW data sets from Solomon's VRPTW instances (Solomon, 1987). We modify Solomon's R, C and RC benchmark instances, which correspond to randomly located customers, clustered customers, and a mix of random and clustered customers, respectively. According to the interviews with pharmaceutical sales representatives, in an urban area, a representative could spend a whole morning meeting five to six physicians at a hospital, while in rural area, the representative on average visits five doctors/physicians during a day. We select the first 20 customers from Solomon's instances as this corresponds to the largest number of doctors that a sales representative would consider when devising her daily schedule. We maintain each customer's location and demand information while modifying each customer's time window. Our first step in modifying the time windows was to scale time from the 1000 time-unit day to a 540-minute day, [0, 540], which corresponds to an 8 A.M. to 5 P.M. work day. If the resulting time window is less than 240 minutes wide, then it is complete. If the resulting time window exceeded 240 minutes, it was truncated to 240 minutes and re-positioned to either [0, 240] or [300, 540] (choice made randomly). Thus, the maximum time window width is 240 minutes, and 240-minute time windows correspond to cases in which doctors have set aside a morning or afternoon during which they may meet with representatives in between handling other work obligations (seeing patients, paperwork, and other tasks). Our SOPTW data sets are as well available at http://ir.uiowa.edu/tippie_pubs/61/.

2.6 Lower and Upper Bounds

In this section we describe a deterministic orienteering problem with time windows which we use to help determine the value of explicitly modeling stochasticity in the SOPTW. We describe a dynamic programming approach to solve this deterministic problem. By solving the deterministic problem with appropriate sets of input data, we obtain lower and upper bounds for the SOPTW.

In a deterministic instance of an orienteering problem with time windows, we assume that given the arrival time at customer i, a queue length q_i, wait time w_i, and service time s_i are known with certainty. Let N_v denote the set of customers visited in a tour. The objective of the problem can be represented as $\sum_{i \in N_v} r_i I(B_i)$, where $I(B_i)$ equals 1 if $B_i \leq l_i$ and 0 otherwise.

We modify the dynamic programming solution approach proposed by Feillet et al. (2004) to solve this deterministic problem as an elementary longest path problem with resource constraints. The dynamic program is formulated as follows. Let N represent the set of vertices visited prior to node i and including node i. Let R be the sum of the rewards collected from all nodes $k \in N$. Use L to denote the departure time from node i. We represent a state as (i, N, R, L). For a state (i, N, R, L), we can either travel from node i to a node j ($j \in \{V \setminus N\}$) or end the tour. Traveling from node i to node j results in a transition from a state (i, N, R, L) to a state $(j, N \cup j, R + r_j, L + c_{ij} + w_j + s_j)$. The reward r_j collected from customer j equals 0 if the representative cannot start a meeting with customer j before his/her time window ends (i.e. $a_j + w_j > l_j$). Therefore, in our problem, we do not update a state

(i, N, R, L) to a state $(j, N \cup j, R + r_j, L + c_{ij} + w_j + s_j)$ if $r_j = 0$.

Let Ξ_i denote the set of states associated with node i, where each state corresponds to a path from the depot to node i. For each state $(i, N, R, L) \in \Xi_i$, consider each of its successors $(j, N \cup j, R + r_j, L + c_{ij} + w_j + s_j)$. If $j \in \{V \setminus N\}$ and $r_j > 0$, the path from the depot to node i can be extended to node j, thus the state $(j, N \cup j, R + r_j, L + c_{ij} + w_j + s_j)$ is added to the states set Ξ_j. The dynamic program begins with the state $(0, 0, 0, 0)$, where node 0 is the depot, and ends when all states in each set Ξ_i have been examined.

A dominance relation can be defined for two states if they differ in only departure time and the sum of rewards. Consider two states (i, N, R, L) and (i, N', R', L') associated with node i. The state (i, N, R, L) dominates the state (i, N', R', L') if $L \leq L'$ while $N = N'$ and $R = R'$. The state (i, N, R, L) dominates the state (i, N', R', L') if $R > R'$ when $N = N'$ and $L = L'$. For each state set Ξ_i, we consider only non-dominated states. The dominance relation improves the computational efficiency of the dynamic program by discarding dominated states. Also, as discussed earlier, our dynamic program never extends a path to a node if no reward can be collected from that node.

To establish a lower bound on an instance of the SOPTW with random variables Q_i and W_i, we consider a corresponding deterministic instance in which, given the arrival time at customer i, the queue length and wait time are set equal to the the expectations, $E[Q_i|a_i]$ and $E[W_i|q_i, a_i]$, respectively. As was discussed in the distribution generation, the service time is set to be 10 minutes. The deterministic dynamic

program generates a sequence of customers which, based on the expected wait times, can be visited within the respective time windows. We refer to this sequence as the original tour. We append the original tour with the set of unvisited customers in order of ascending time window closures and refer to the resulting tour as the deterministic tour. Evaluating the deterministic tour using Equation (2.10) provides a lower bound for the SOPTW instance under consideration.

We establish an empirical upper bound on an instance of the SOPTW with random variables Q_i and W_i by treating each of the 1000 samples used by the VND (Algorithm 2.2) as deterministic input to the dynamic programming algorithm of §2.6. The reported upper bound is the average of the perfect information solutions over 1000 samples.

2.7 Computational Experiments

We discuss the implementation of computational experiments in §2.7.1 and present corresponding results in §2.7.2. We describe setting values of $\tilde{r}$ and $\check{r}$ to execute the recourse actions. As discussed in §2.2, $\tilde{r}$ and $\check{r}$ indicate the minimum expected reward for which the representative is willing to travel to a customer and to wait at a customer, respectively. We then test the quality of VNS by comparing heuristic solutions to optimal solutions of instances with 10 customers, which are the largest instances for which we could obtain optimal solutions (via enumeration). We also use these 10-customer instances to evaluate the quality of the lower bounds and empirical upper bounds described in §2.6. We examine the effectiveness of using

sampling in approximating the objective evaluation versus analytical evaluation via Equation (2.10). Finally we compare the SOPTW solutions to the lower and upper bounds described in §2.6.

2.7.1 Implementation

The VNS and the deterministic dynamic programming approach from §2.6 are programmed in C++ and implemented on a Intel Core i7 processor with 3.4 GHz and 16 GB of RAM. We execute the VNS 10 times for each instance and evaluate the expected objective using Equation (2.10) derived in §2.3. For each SOPTW solution, we report the average expected objective of the 10 runs. In the execution of an a priori tour, we assume that the sales representative can arrive at the first customer as early as she wants. That is, the tour effectively starts at the first customer rather than at a depot. In our data sets, this implies the representative arrives at the first customer one hour before the customer's time window opens.

To investigate the setting of $\tilde{r}$ and $\check{r}$, we first run the VNS with different pairs of ($\tilde{r}$,$\check{r}$) on the R, C and RC instances described in §2.5.2. We test with both $\tilde{r}$ and $\check{r}$ values ranging from 0 to $\min_i\{r_i\}$ and consider an increment in each parameter as small as 0.25 (i.e., we consider the values of 0, 0.25, 0.5, ..., $\min_i\{r_i\}$ for $\tilde{r}$ and $\check{r}$, respectively). For various values of ($\tilde{r}$,$\check{r}$), we compare the expected reward collected by the best-found tour (averaged over 10 VNS runs per instance). The results in Table 2.1 through 2.3 demonstrate that it is challenging to find a universal value of the ($\tilde{r}$,$\check{r}$) pair. There is no single pair of $(\tilde{r}, \check{r})$ that dominates all others for each

Table 2.1: Results on $(\tilde{r}, \check{r})$ for R Instances with 20 Customers

Dataset	**(0,0)**	**(0,1)**	**(0,2)**	**(0.1,0.1)**	**(0.25,0.75)**	**(0.25,1)**	**(0.5,0.5)**
R101	55.9	55.9	55.6	55.8	55.7	55.7	55.7
R102	84.2	85.8	85.0	85.9	85.9	85.9	86.0
R103	97.9	97.9	97.8	97.9	97.9	97.8	97.8
R104	101.5	101.7	102.2	101.7	101.7	101.9	101.9
R105	69.5	69.4	69.1	69.5	69.3	69.2	69.1
R106	91.6	91.4	91.4	91.5	91.4	91.5	91.4
R107	94.0	94.1	94.1	94.1	94.1	94.1	94.1
R108	104.8	105.1	105.1	105.1	105.2	105.2	105.1
R109	89.4	89.6	90.0	89.7	89.6	89.6	89.5
R110	93.8	94.4	94.4	94.2	94.4	94.5	94.4
R111	101.6	101.5	101.5	101.6	101.5	101.5	101.5
R112	114.2	114.3	114.7	114.4	114.5	114.6	114.6
R201	73.6	74.2	73.8	73.9	74.1	74.1	74.3
R202	94.4	94.3	94.0	94.4	94.4	94.3	94.3
R203	102.8	103.6	103.7	103.5	103.6	103.7	103.3
R204	107.6	107.6	108.2	107.6	107.8	107.9	107.6
R205	91.4	91.6	92.6	91.6	92.1	92.2	91.9
R206	102.6	103.0	103.0	102.8	103.0	103.0	102.9
R207	111.2	111.3	111.2	111.2	111.3	111.3	111.2
R208	107.8	107.9	107.6	107.7	107.9	107.9	107.9
R209	102.4	102.0	101.9	102.2	102.1	102.0	102.0
R210	106.0	106.1	106.1	106.1	105.9	106.0	106.1
R211	114.3	114.6	115.0	114.8	115.0	115.1	115.1
Average	96.2	96.4	96.4	96.4	96.4	96.5	96.4

instance. However, (0.25, 1), (2,6) and (0.5,0.5) give the best average over all R, C and RC instances, respectively. Therefore, we use them in the following experiments for SOPTW solutions. For the lower bound, the deterministic solutions are evaluated using the same value of $(\tilde{r}, \check{r})$ as are used in obtaining the corresponding SOPTW solutions.

Table 2.2: Results on $(\tilde{r}, \check{r})$ for C Instances with 20 Customers

Dataset	**(0,0)**	**(0,2)**	**(0.1,0.1)**	**(0.25,0.25)**	**(0.25,10)**	**(2,2)**	**(2,6)**
C101	88.4	90.9	89.0	97.0	100.2	98.5	98.8
C102	89.7	89.8	89.2	89.3	89.9	91.1	92.2
C103	134.4	135.7	136.1	136.1	130.4	135.3	135.9
C104	145.7	146.4	146.3	146.3	145.4	147.0	149.6
C105	100.5	99.7	100.5	100.4	101.4	102.5	100.0
C106	81.7	91.2	87.0	87.0	96.4	94.6	97.8
C107	125.7	123.4	123.9	123.9	124.6	126.6	128.1
C108	130.4	130.1	130.1	130.1	130.0	129.4	129.3
C109	156.8	157.6	157.1	157.1	154.7	157.1	155.8
C201	89.6	92.3	89.1	91.0	94.9	92.1	94.1
C202	107.9	109.6	108.5	109.1	110.4	110.7	114.1
C203	136.1	136.8	136.6	137.1	131.8	137.1	137.4
C204	130.7	131.1	131.0	130.9	124.2	129.9	131.6
C205	108.3	107.7	108.2	108.3	108.1	106.9	107.5
C206	124.5	125.7	125.4	125.6	120.5	123.5	122.1
C207	142.1	143.1	142.4	142.5	137.1	143.1	142.8
C208	142.2	142.3	142.1	142.2	140.1	142.2	141.9
Average	119.7	120.8	120.1	120.8	120.0	121.6	122.3

Table 2.3: Results on $(\tilde{r}, \check{r})$ for RC Instances with 20 Customers

Dataset	**(0,0)**	**(0,2)**	**(0.25,0.25)**	**(0.5,0.5)**	**(0,4)**	**(1,1)**	**(4,4)**
RC101	90.6	91.8	90.7	91.8	91.0	91.8	90.5
RC102	130.8	132.0	130.8	130.7	132.6	130.7	132.4
RC103	161.4	161.2	161.5	161.4	160.4	161.1	159.4
RC104	166.4	165.8	166.5	165.9	166.5	165.0	165.6
RC105	134.2	134.1	135.0	135.1	133.8	135.1	134.6
RC106	116.5	116.4	116.5	116.5	116.3	116.4	116.0
RC107	180.4	180.3	180.5	180.5	179.4	180.2	178.8
RC108	179.5	180.1	180.2	180.3	179.9	179.8	178.9
RC201	107.5	106.5	107.1	108.0	104.6	107.9	108.0
RC202	149.0	148.9	149.1	148.9	147.7	148.7	146.8
RC203	153.3	154.3	154.0	154.2	153.2	154.2	152.9
RC204	159.6	159.4	159.7	159.7	158.7	159.3	157.4
RC205	132.1	133.0	132.1	133.0	133.9	132.8	133.6
RC206	131.9	132.0	131.9	131.8	131.8	131.8	132.2
RC207	160.9	162.1	161.4	161.6	162.0	162.2	162.3
RC208	173.3	173.9	173.7	173.4	173.7	173.5	173.1
Average	145.5	145.7	145.7	145.8	145.3	145.7	145.2

As a preliminary test of the quality of the VNS heuristic, we execute it on 10-customer R, C and RC instances, which are the largest instances we could solve optimally via enumeration. The VNS heuristic (using sampling) obtains the optimal solution in 42 of the 56 instances, with average optimality gaps for R, C and RC instances of 0.2%, 0.01% and 0.02%, respectively. We note that these small gaps are typically due to the sampling and executing the VNS heuristic with the analytical objective evaluation in Equation 2.10 eliminates the gap. We also use these 10-customer instances to compare the lower bounds and empirical upper bounds to the optimal solutions. We observe that the lower bounds are on average 5.6%, 8.0%, and 7.2% less than the optimal solutions and the upper bounds are on average 11.8%, 13.2%, and 11.6% greater than the optimal solutions for R, C and RC instances, respectively. We provide the detailed results in Tables 2.4 through 2.6.

To demonstrate the effectiveness of approximating objective function via sampling, we compare the performance of sample-based VNS to VNS using Equation (2.10) in evaluating objective. We first run the VNS using sampling to evaluate the objective function. Then, for the same instance, we run the VNS evaluating the objective function exactly. The corresponding results are presented in Tables 2.7 through 2.9. On average, the objectives obtained using exact objective function evaluation are only 0.05%, 0.06%, and 0.14% better than those obtained with sampling for the 20-customer R, C and RC instances, respectively. However, the computational time increases dramatically from an average of around 12 minutes for using sampling to an average of around 6 hours for using exact objective function. Thus, we focus the

Table 2.4: Results for R Instances with 10 Customers

Dataset	$\overline{R}$	$\overline{R}_{best}$	$\overline{\delta}_R$	$Opt.$	LB	UB	Gap_{opt}	Gap_{lb}	Gap_{ub}
R101	49.37	49.37	0.00	49.37	49.37	61	0.0%	0.0%	19.1%
R102	60.81	60.81	0.00	61.43	60.39	71	1.0%	1.7%	13.5%
R103	73.66	73.67	0.01	73.67	69.73	84	0.0%	5.3%	12.3%
R104	71.30	71.30	0.00	71.30	69.96	84	0.0%	1.9%	15.1%
R105	58.74	58.74	0.00	58.74	54.35	68	0.0%	7.5%	13.6%
R106	71.66	71.66	0.00	71.79	67.41	78	0.2%	6.1%	8.0%
R107	66.37	66.37	0.00	66.37	62.28	75	0.0%	6.2%	11.5%
R108	76.92	76.92	0.00	76.92	74.61	88	0.0%	3.0%	12.6%
R109	68.63	68.63	0.00	68.63	65.03	77	0.0%	5.3%	10.9%
R110	73.34	73.34	0.00	73.34	72.75	87	0.0%	0.8%	15.7%
R111	71.73	71.73	0.00	71.73	66.87	88	0.0%	6.8%	18.5%
R112	81.32	81.32	0.00	81.45	77.41	90	0.2%	5.0%	9.5%
R201	59.82	59.82	0.00	59.91	57.95	68	0.2%	3.3%	11.9%
R202	75.23	75.23	0.00	75.23	66.07	83	0.0%	12.2%	9.4%
R203	74.91	74.91	0.00	74.91	65.30	86	0.0%	12.8%	12.9%
R204	82.72	82.72	0.00	82.72	82.71	90	0.0%	0.0%	8.1%
R205	73.02	73.02	0.00	73.02	71.48	81	0.0%	2.1%	9.8%
R206	74.74	74.74	0.00	75.06	71.11	84	0.4%	5.3%	10.6%
R207	83.51	83.51	0.00	83.51	72.98	86	0.0%	12.6%	2.9%
R208	81.18	81.18	0.00	81.18	68.82	91	0.0%	15.2%	10.8%
R209	80.94	80.94	0.00	80.94	77.65	90	0.0%	4.1%	10.1%
R210	62.26	62.26	0.00	62.40	60.19	73	0.2%	3.5%	14.5%
R211	83.31	83.49	0.00	83.49	77.28	94	0.2%	7.4%	11.2%
						Average	0.1%	5.6%	11.8%

$\overline{R}$, δ_R: the average objective and standard deviation of the best solution over 10 VNS runs;
$\overline{R}_{best}$: the best objective found over 10 VNS runs;
$Opt.$: the optimal solution;
LB: the lower bound;
UB: the empirical upper bound;
$Gap_{opt.}$: the gap between $\overline{R}$ and the optimal solution, $Gap_{opt.} = \frac{Opt.-\overline{R}}{Opt.}$.
Gap_{lb}: the difference between the optimal solution and the lower bound, $Gap_{lb} = \frac{Opt.-LB}{Opt.}$
Gap_{ub}: the difference between the optimal solution and the empirical upper bound, $Gap_{ub} = \frac{UB-Opt.}{UB}$

Table 2.5: Results for C Instances with 10 Customers

Dataset	$\overline{R}$	$\overline{R}_{best}$	$\overline{\delta}_R$	*Opt.*	*LB*	*UB*	Gap_{opt}	Gap_{lb}	Gap_{ub}
C101	68.06	68.06	0.00	68.06	60.34	80	0.0%	11.3%	17.5%
C102	85.03	85.03	0.00	85.03	79.92	100	0.0%	6.0%	17.6%
C103	97.78	97.78	0.00	97.85	95.49	120	0.1%	2.4%	22.6%
C104	102.07	102.07	0.00	102.08	92.64	110	0.0%	9.2%	7.8%
C105	72.30	72.30	0.00	72.30	69.90	80	0.0%	3.3%	10.7%
C106	73.73	73.73	0.00	73.73	67.96	80	0.0%	7.8%	8.5%
C107	86.04	86.04	0.00	86.04	84.60	100	0.0%	1.7%	16.2%
C108	88.70	88.70	0.00	88.71	86.76	100	0.0%	2.2%	12.7%
C109	99.87	99.87	0.00	99.91	96.41	120	0.0%	3.5%	20.1%
C201	54.80	54.80	0.00	54.80	52.26	70	0.0%	4.6%	27.7%
C202	84.12	84.12	0.00	84.12	79.68	90	0.0%	5.3%	7.0%
C203	87.78	87.78	0.00	87.78	80.39	90	0.0%	8.4%	2.5%
C204	94.38	94.38	0.00	94.38	78.12	100	0.0%	17.2%	6.0%
C205	63.64	63.64	0.00	63.64	63.34	70	0.0%	0.5%	10.0%
C206	73.65	73.65	0.00	73.65	62.21	80	0.0%	15.5%	8.6%
C207	74.78	74.78	0.00	74.81	67.31	90	0.0%	10.0%	20.3%
C208	83.08	83.08	0.00	83.08	61.43	90	0.0%	26.1%	8.3%
						Average	0.01%	7.95%	13.2%

Table 2.6: Results for RC Instances with 10 Customers

Dataset	$\overline{R}$	$\overline{R}_{best}$	$\overline{\delta}_R$	*Opt.*	*LB*	*UB*	Gap_{opt}	Gap_{lb}	Gap_{ub}
RC101	91.01	91.01	0.00	91.01	83.6013	100	0.0%	8.1%	9.0%
RC102	125.90	125.90	0.00	125.90	125.766	150	0.0%	0.1%	16.1%
RC103	133.80	133.80	0.00	133.80	133.555	150	0.0%	0.2%	10.8%
RC104	156.58	156.58	0.00	156.58	144.02	160	0.0%	8.0%	2.1%
RC105	120.00	120.00	0.00	120.11	106.124	130	0.1%	11.6%	7.6%
RC106	109.55	109.55	0.00	109.55	98.8467	120	0.0%	9.8%	8.7%
RC107	144.58	144.58	0.00	144.58	137.141	160	0.0%	5.1%	9.6%
RC108	138.93	138.93	0.00	138.93	127.221	160	0.0%	8.4%	13.2%
RC201	102.43	102.43	0.00	102.43	88.9562	120	0.0%	13.2%	14.6%
RC202	119.81	119.81	0.00	119.81	115.422	140	0.0%	3.7%	14.4%
RC203	138.30	138.30	0.00	138.30	122.392	160	0.0%	11.5%	13.6%
RC204	137.96	137.96	0.00	138.24	124.32	160	0.2%	10.1%	13.6%
RC205	119.25	119.25	0.00	119.25	118.777	140	0.0%	0.4%	14.8%
RC206	123.64	123.64	0.00	123.65	121.028	150	0.0%	2.1%	17.6%
RC207	134.25	134.25	0.00	134.25	128.405	160	0.0%	4.4%	16.1%
RC208	153.67	153.67	0.00	153.67	126.276	160	0.0%	17.8%	4.0%
						Average	0.02%	7.16%	11.6%

rest of our presentation on the results from runs using sampling.

2.7.2 Results

Tables 2.10 through 2.12 compare the expected reward collected by the a priori tours obtained by the VNS heuristic to the bounds. The upper and lower bounds are obtained in the manner discussed in §2.6. In each of the three tables, the second, third and fourth columns report the lower bounds, the averaged expected objectives of SOPTW over parameter sets discussed above, and the upper bounds, respectively. The fifth column shows the gap between the lower bound and the SOPTW ($Gap\ L = \frac{SOPTW-LB}{LB} \times 100\%$) and the sixth column shows the gap between the upper bound and the SOPTW ($Gap\ U = \frac{UB-SOPTW}{UB} \times 100\%$). The runtime for solving a deterministic version of the problem is trivial, less than one second in most cases. The average runtime over the 10 VNS runs for each instance is reported in the last column of the tables, where parameters in the VNS are set as discussed in §2.4.

On average, the stochastic solutions are around 9.2% better than the lower bound for all datasets. Specifically, the average gap between the stochastic solutions and the lower bounds of Solomon's C instances is 10.2%, which is slightly larger than that of the RC (8.9%) and R instances (8.4%). In some cases, the improvement is significant, with the largest improvement in R-instances being 22.7%, 20.8% for the C-instances, and 21.5% for the RC-instances.

For instances R105, R209, C105, C108 and C205 the gaps between SOPTW solutions and lower bounds are quite small. There does not seem to be a clear factor

Table 2.7: Approx. v.s. Exact Eval. for R Instances

Dataset	$\overline{R}_s$	$\bar{t}_s$	$\overline{R}_e$	$\bar{t}_e$	*Gap*
R101	55.74	194	55.75	298	0.02%
R102	85.91	292	85.91	5931	0.00%
R103	97.79	360	97.90	20517	0.11%
R104	101.88	421	102.14	22115	0.25%
R105	69.23	264	69.23	3290	0.00%
R106	91.52	379	91.53	14666	0.01%
R107	94.07	349	94.15	30763	0.08%
R108	105.18	458	105.18	25909	0.00%
R109	89.57	310	89.70	11972	0.14%
R110	94.47	478	94.47	39375	0.00%
R111	101.47	439	101.51	32517	0.04%
R112	114.64	506	114.69	54025	0.04%
R201	74.05	218	74.15	2732	0.13%
R202	94.31	329	94.31	7911	0.00%
R203	103.72	394	103.76	13489	0.04%
R204	107.92	350	107.93	15052	0.01%
R205	92.19	243	92.25	17140	0.07%
R206	102.99	251	102.99	19012	0.00%
R207	111.28	229	111.32	40315	0.04%
R208	107.89	277	107.89	31202	0.00%
R209	102.01	281	102.07	34747	0.06%
R210	106.04	250	106.13	17517	0.08%
R211	115.05	250	115.14	44082	0.08%
				Average	0.05%

$\overline{R}_s$: the average objective over 10 VNS runs obtained via using sampling in objective evaluation;
$\overline{R}_e$: the best objective over 10 VNS runs obtained via using exact objective function;
$\bar{t}_s$: the average CPU in seconds for using sampling;
$\bar{t}_e$: the average CPU in seconds for using exact objective function;
Gap: the gap between $\overline{R}_s$ and $\overline{R}_e$, $Gap = \frac{\overline{R}_e - \overline{R}_s}{\overline{R}_e}$.

Table 2.8: Approx. v.s. Exact Eval. for C Instances

Dataset	$\overline{R}_s$	$\bar{t}_s$	$\overline{R}_e$	$\bar{t}_e$	*Gap*
C101	98.81	347	98.83	788	0.02%
C102	92.16	373	92.26	6052	0.11%
C103	135.87	638	135.88	37627	0.01%
C104	149.60	553	149.60	24918	0.00%
C105	100.04	195	100.05	1353	0.01%
C106	97.77	247	97.77	1888	0.00%
C107	128.13	272	128.13	4357	0.00%
C108	129.26	289	129.54	9231	0.22%
C109	155.85	563	156.28	46365	0.28%
C201	94.07	295	94.07	1282	0.00%
C202	114.09	529	114.09	21872	0.00%
C203	137.44	534	137.44	15256	0.00%
C204	131.61	675	132.03	53336	0.32%
C205	107.53	320	107.55	2274	0.02%
C206	122.12	335	122.14	6622	0.02%
C207	142.81	499	142.85	16681	0.03%
C208	141.90	338	141.90	13017	0.00%
				Average	0.06%

Table 2.9: Approx. v.s. Exact Eval. for RC Instances

Dataset	$\overline{R}_s$	$\bar{t}_s$	$\overline{R}_e$	$\bar{t}_e$	Gap
RC101	91.76	209	91.76	3754	0.00%
RC102	130.67	286	130.76	3618	0.07%
RC103	161.42	429	161.42	9833	0.00%
RC104	165.87	470	166.68	13560	0.49%
RC105	135.11	404	135.11	10167	0.00%
RC106	116.53	297	116.71	13525	0.15%
RC107	180.47	675	180.47	76762	0.00%
RC108	180.25	503	180.25	37345	0.00%
RC201	107.96	258	108.66	4246	0.64%
RC202	148.89	365	148.89	9524	0.00%
RC203	154.19	453	154.32	14594	0.08%
RC204	159.65	482	159.65	22749	0.00%
RC205	133.02	437	133.02	16487	0.00%
RC206	131.85	424	132.31	8965	0.35%
RC207	161.57	557	162.00	58098	0.27%
RC208	173.35	436	173.80	78743	0.26%
				Average	0.14%

underlying the differences. For example, we find that instances R105, C105, C108 and C205 have narrow time windows, while this is not the case for R209. Admittedly, factors such as a customer's location, expected economic value, and traveling distances between customers also has impact on the structure of a solution.

The advantage of a SOPTW solution over the deterministic solution using expected values is that a SOPTW solution inserts customers early in the sequence that, while unlikely to be visited, can be considered with a favorable realization of queue lengths and waiting times at preceding customers. In contrast, the deterministic solution using expected values will only sequence the customers guaranteed to be visited using the expected wait times (the remaining customers are appended in increasing order of l_i).

To see this, we examine the structure of both the deterministic (lower bound) and the stochastic solutions for instances R202 and R209 in Tables 2.13 and 2.14. The first column in both tables presents the deterministic solution, the second column is the time window for each customer and the third column shows the probability of collecting a reward from a customer, which is computed via Equation (2.10) in §2.3. Similarly, the fourth, fifth and sixth columns in both tables present the SOPTW solution, the time window and the probability of collecting a reward from a customer, respectively. We note that $0^{\dagger}$ denotes a nonzero number less than 10^{-3}).

For instance R202, the original deterministic tour is $(0, 5, 14, 18, 10, 13, 12)$. The deterministic solution presented in the table consists of both the original tour and the appended customers. We can see that, in the deterministic solution, a reward

is more likely to be collected from customers 5, 14, 18, 10, 13, and 12 than from others. Using the average queue lengths and wait times, these customers will be visited with certainty. In the stochastic solution, customers 5 and 14 are still visited early in the sequence (although in reversed order), but customer 10 is moved to the end of the sequence. Furthermore, customers 13 and 12 have been moved later in the sequence as well, but actually have a larger probability of being visited than they do in the sequence from the deterministic solution. This occurs because customers 3, 9, 8, 1, and 6 have been inserted earlier in the tour (and each of these customers has a relatively low probability of being visited implying that they will often be skipped or balked at, but having the option to visit them is beneficial if there is a favorable realization of queue length and wait time).

For instance R209, the situation is similar. The original deterministic tour is $(0, 5, 13, 14, 10, 1, 4)$. As demonstrated in Table 2.14, for both the deterministic and stochastic solutions, customers 5, 13, 14, 10, 1 and 4 have greater chances to provide a reward than other customers. In the stochastic tour, however, customers 16, 19, 12, 17 and 11, which are less likely to provide a reward, are inserted earlier in the tour to take advantage of favorable realizations in which they can be visited.

The average gaps between the SOPTW solutions and the upper bounds for the R, C and RC instances are 25.6%, 25.5% and 27.8%, respectively. The upper bound gaps are large in general, which suggests the looseness of the perfect information bounds. There does not exist a discernible cause for the magnitude of any particular gap. Among the R instances, the largest upper bound gap is 29.6% from instance

R107 and the smallest is 17.4% from instance R101. R101 has the smallest average time window width over all R instances while R107's is neither the smallest or the largest. Among the C instances, the largest upper bound gap is 31% from instance C102 and the smallest is 14.8% from instance C101. Instance C101 has the second smallest average time window width over all C instances while C102 has neither the smallest or the largest. For the RC instances, the largest upper bound gap is 32.6% from RC208, the instance with the largest average time window width. The smallest upper bound gap is 23.2% from RC103, whose average time window width is neither the largest nor the smallest.

Recall that we motivate consideration of 20-customer data sets as this size of problem corresponds to an upper bound on the number of doctors that a sales representative would be able to visit in a day. That is, we assume that the sales representative identifies a set of 20 customers on which to base her a priori itinerary. For sake of comparison, we also consider 40-customer R, C, and RC instances and present the detailed results in Table 2.15. To construct the instances, we select the first 40 customers from Solomon's instances and modify the original time windows following the procedure discussed in §2.5.2. We use the same settings of $(\tilde{r}, \check{r})$ as in the 20-customer instances. The objectives presented in the second and sixth column of Table 2.15 are the expected reward averaged over the 10 VNS runs. In the third and seventh columns, we provide the average runtime in seconds for each instance. The fourth and eighth columns present the average number of customers (over the 10 VNS runs) in an a priori route whose chance of being visited is no less than 0.01

Table 2.10: Results for R Instances with 20 Customers

Dataset	LB	SOPTW	UB	Gap L	Gap U	Runtime
R101	50.86	55.74	67.45	9.6%	17.4%	194
R102	78.01	85.91	109.60	10.1%	21.6%	292
R103	89.19	97.79	128.42	9.6%	23.9%	360
R104	97.90	101.88	137.82	4.1%	26.1%	421
R105	68.15	69.23	90.29	1.6%	23.3%	264
R106	79.74	91.52	121.34	14.8%	24.6%	379
R107	91.31	94.07	133.58	3.0%	29.6%	349
R108	96.49	105.18	147.01	9.0%	28.5%	458
R109	86.73	89.57	118.32	3.3%	24.3%	310
R110	86.21	94.47	129.50	9.6%	27.0%	478
R111	87.30	101.47	143.46	16.2%	29.3%	439
R112	107.49	114.64	159.25	6.6%	28.0%	506
R201	60.36	74.05	95.22	22.7%	22.2%	218
R202	83.85	94.31	123.85	12.5%	23.8%	329
R203	97.00	103.72	147.03	6.9%	29.5%	394
R204	104.08	107.92	143.35	3.7%	24.7%	350
R205	85.76	92.19	125.24	7.5%	26.4%	243
R206	95.64	102.99	141.30	7.7%	27.1%	251
R207	95.64	111.28	149.01	16.3%	25.3%	229
R208	100.12	107.89	153.01	7.8%	29.5%	277
R209	100.21	102.01	138.46	1.8%	26.3%	281
R210	99.64	106.04	140.52	6.4%	24.5%	250
R211	111.25	115.05	157.51	3.4%	27.0%	250

Table 2.11: Results for C Instances with 20 Customers

Dataset	LB	SOPTW	UB	Gap L	Gap U	Runtime
C101	86.81	98.81	116.0	13.8%	14.8%	347
C102	82.04	92.16	133.6	12.3%	31.0%	373
C103	130.21	135.87	194.0	4.4%	29.9%	638
C104	128.16	149.60	210.0	16.7%	28.8%	553
C105	100.01	100.04	134.9	0.02%	25.8%	195
C106	89.19	97.77	119.5	9.6%	18.1%	247
C107	124.60	128.13	165.3	2.8%	22.5%	272
C108	126.89	129.26	182.4	1.9%	29.1%	289
C109	152.12	155.85	219.9	2.4%	29.1%	563
C201	80.20	94.07	115.3	17.3%	18.4%	295
C202	96.42	114.09	151.9	18.3%	24.9%	529
C203	119.71	137.44	183.8	14.8%	25.2%	534
C204	108.99	131.61	188.1	20.8%	30.0%	675
C205	107.39	107.53	141.7	0.1%	24.1%	320
C206	114.45	122.12	165.8	6.7%	26.3%	335
C207	127.62	142.81	200.7	11.9%	28.8%	499
C208	119.14	141.90	193.5	19.1%	26.7%	338

Table 2.12: Results for RC Instances with 20 Customers

Dataset	LB	SOPTW	UB	Gap L	Gap U	Runtime
RC101	88.98	91.76	124.70	3.1%	26.4%	209
RC102	122.36	130.67	180.67	6.8%	27.7%	286
RC103	144.26	161.42	210.20	11.9%	23.2%	429
RC104	136.55	165.87	244.27	21.5%	32.1%	470
RC105	116.38	135.11	179.04	16.1%	24.5%	404
RC106	107.54	116.53	169.59	8.4%	31.3%	297
RC107	173.35	180.47	236.53	4.1%	23.7%	675
RC108	169.29	180.25	255.24	6.5%	29.4%	503
RC201	102.28	107.96	142.42	5.6%	24.2%	258
RC202	141.10	148.89	200.90	5.5%	25.9%	365
RC203	137.16	154.19	209.01	12.4%	26.2%	453
RC204	144.13	159.65	224.50	10.8%	28.9%	482
RC205	127.62	133.02	191.97	4.2%	30.7%	437
RC206	126.04	131.85	192.20	4.6%	31.4%	424
RC207	152.24	161.57	220.12	6.1%	26.6%	557
RC208	150.97	173.35	257.12	14.8%	32.6%	436

Table 2.13: Deterministic v.s. SOPTW for R202

Det. Sol.	Time Window	Prob.	SOPTW Sol.	Time Window	Prob.
0	[0,540]	1	0	[0,540]	1
5	[0,240]	1	14	[17,131]	1
14	[17,131]	0.542	5	[0,240]	0.928
18	[203,234]	0.586	3	[0,240]	0.541
10	[311,341]	0.750	9	[216,268]	0.127
13	[372,446]	0.516	18	[203,234]	1.684E-03
12	[300,540]	0.762	8	[218,259]	2.115E-03
15	[94,162]	0	1	[0,240]	1.885E-03
11	[111,175]	0	6	[224,277]	1.687E-03
16	[146,201]	0	13	[372,446]	0.983
19	[145,204]	0	12	[300,540]	0.868
1	[0,240]	0	15	[94,162]	0
3	[0,240]	0	16	[146,201]	0
8	[218,259]	0	17	[395,469]	0
9	[216,268]	0	2	[300,540]	0.272
6	[224,277]	0	7	[300,540]	8.924E-03
20	[313,359]	0	11	[111,175]	0
4	[366,432]	$0^{\dagger}$	4	[366,432]	0
17	[395,469]	0	20	[313,359]	0
2	[300,540]	0.221	19	[145,204]	0
7	[300,540]	0.015	10	[311,341]	0

Table 2.14: Deterministic v.s. SOPTW for R209

Det. Sol.	Time Window	Prob.	SOPTW Sol.	Time Window	Prob.
0	[0,540]	1	0	[0,540]	1
5	[10,199]	1	5	[10,199]	1
13	[0,240]	0.912	13	[0,240]	0.912
14	[17,245]	0.813	14	[17,245]	0.813
10	[267,385]	0.799	16	[119,228]	0.174
1	[343,496]	0.843	19	[116,233]	0.016
4	[300,540]	0.749	12	[91,217]	$0^{\dagger}$
2	[39,189]	0	17	[0,240]	$0^{\dagger}$
12	[91,217]	0	11	[0,240]	2.996E-03
16	[119,228]	0	10	[267,385]	0.742
19	[116,233]	0	1	[343,496]	0.821
11	[0,240]	0	9	[137,347]	$0^{\dagger}$
17	[0,240]	0	6	[143,358]	0
7	[157,242]	0	4	[300,540]	0.718
18	[188,249]	0	20	[290,382]	0
8	[155,322]	0	8	[155,322]	0
3	[268,331]	0	18	[188,249]	0
9	[137,347]	0	3	[268,331]	0
6	[143,358]	0	15	[300,540]	0.181
20	[290,382]	0	7	[157,242]	0
15	[300,540]	0.192	2	[39,189]	0

(via evaluating the a priori route using the exact objective function). We find that although they consider twice as many customers, the a priori routes with 40 customers do not visit more customers than in the 20-customer solutions. On average, the numbers of customers having at least 0.01 probability of being visited are 10, 12 and 9 for the 40-customer R, C and RC sets and are 9, 11 and 10 for the 20-customer R, C and RC instances, respectively. Note that we cannot provide lower and upper bounds for these solutions (the deterministic dynamic program is unable to solve instances with 40 customers). This demonstrates that considering larger data sets only potentially impacts which customers are visited and not the number of customers visited.

2.8 Summary and Future Work

Motivated by the daily routing problem faced by a pharmaceutical representative, we introduce a stochastic orienteering problem with time windows (SOPTW) characterized by stochastic wait times as well as a time window for each customer. Specifically, the wait time a representative encounters depends on the queue lengths observed when the representative arrives at a doctor's office. In the development of a priori tours, we consider two types of recourse actions. The first recourse action determines whether or not to skip a customer on an a priori tour based on the arrival time at that customer. The second recourse action determines how long a representative should wait at a customer after arriving and observing a queue. These recourse actions are addressed in the analytical formula that we derive to compute the expected reward for

Table 2.15: Results on Modified Solomon Instances with 40 Customers

Dataset	**Obj.**	**Runtime**	**Visited**	**Dataset**	**Obj.**	**Runtime**	**Visited**
R101	58.5	2044	5	R201	93.9	2811	7
R102	105.7	5912	11	R202	110.4	4034	9
R103	122.1	6700	11	R203	130.2	4763	11
R104	138.2	5465	11	R204	130.7	5724	12
R105	81.4	3019	8	R205	111.6	4785	10
R106	117.2	4020	9	R206	135.6	4797	10
R107	120.8	4858	10	R207	127.4	6035	10
R108	134.2	5043	11	R208	139.6	6163	11
R109	103.4	4266	9	R209	136.0	6058	10
R110	137.8	5830	10	R210	136.8	5173	9
R111	135.7	5534	12	R211	148.0	6587	12
R112	139.8	5965	10				
C101	104.1	2629	14	C201	131.9	4642	13
C102	130.6	6999	16	C202	146.3	5552	14
C103	159.9	6596	12	C203	155.1	6079	13
C104	172.3	6742	14	C204	168.3	7390	13
C105	107.2	2532	8	C205	121.0	4552	9
C106	114.7	4099	11	C206	144.1	3645	9
C107	133.2	2730	8	C207	180.9	5272	10
C108	136.8	3763	9	C208	161.6	4071	11
C109	173.9	6518	12				
RC101	106.5	2473	7	RC201	126.2	2549	9
RC102	152.2	2920	8	RC202	164.2	4262	9
RC103	161.5	4063	9	RC203	156.7	3774	10
RC104	181.9	5161	11	RC204	183.1	4802	12
RC105	138.4	3965	9	RC205	154.0	4101	8
RC106	122.8	3096	8	RC206	150.1	3942	9
RC107	194.8	4940	11	RC207	175.1	4591	10
RC108	203.7	5509	12	RC208	182.0	5051	9

an a priori tour.

We solve the SOPTW via a VNS heuristic and compute lower and upper bounds on the solution. We obtain the lower bound by solving a deterministic version in which wait times and queue lengths are set to their expectations. We also obtain an upper bound for an a priori tour expected rewards by solving the deterministic problem with perfect information. From our computational results, we find that the SOPTW solutions perform about 9.2% better than the lower bound and about 26.3% worse than the upper bound. The SOPTW solution's 9.2% average improvement over the deterministic approach using expected queue lengths and wait times as well as the difference in the tour sequences suggest the merit in considering uncertainty in this problem.

One future research direction is to investigate dynamic policies for the SOPTW. For the first recourse action proposed in the study, we allow skipping of a customer, but never allow resequencing of customers. A dynamic routing policy would allow resequencing based on realized events. For the second recourse action, we employ an action that determines the longest amount of time a representative will wait based on the queue length upon arrival. A dynamic policy would allow the sales representative to consider the "stay in queue or depart for next customer" decision at each time epoch based on the current queue length. Our results motivate the investigation of dynamic policies in two ways. First, the gaps between our a priori results and the upper bounds suggest that there may be potential to improve the rewards by taking explicit advantage of realizations. Second, there does not seem to be universal set-

tings of $\tilde{r}$ and $\check{r}$ for the recourse actions that result in effective static rules. Thus, dynamically determining when to leave a customer and adapting the tour based on the observed realizations may result in improved performance.

Secondly, this study is limited to a one-day planning horizon. A routing strategy of a longer planning horizon is applicable in real-world situations where the pharmaceutical sales representative needs to develop a weekly or monthly schedule of doctor visits. Finally, in this study, we focus on application to the pharmaceutical industry. We leave as future work the consideration of other domains in which the logistics of managing customer relationships is important. For example, in the textbook industry, representatives can leave and return to a customer depending on the observed line length, adding further complexity to the routing problem.

CHAPTER 3
DYNAMIC ORIENTEERING ON A NETWORK OF QUEUES

3.1 Introduction

In this chapter, we study the daily orienteering problem in the context of routing a textbook salesperson who visits professors located in one or more buildings on campus. Similar to the pharmaceutical case discussed in §2, each professor is associated with 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. However, unlike in §2 where we consider a non-priority queue of other sales representatives, in this study, we consider

a queue of students at each professor and assume that the salesperson has the 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 study 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 §3.2, we provide a MDP model for the problem. We investigate the existence of optimal control limit policies and present the corresponding results in §3.3. We describe our rollout algorithms in §3.4 and present computational results of our rollout methodology in §3.5. In §3.6, we conclude the study and discuss future work.

3.2 Problem Formulation

We model the DOPTW as a Markov decision process (MDP). We refer the readers to Puterman (2005) for an introduction to MDPs. We define the problem on a complete graph $G = (V, E)$. The set V of n nodes corresponds to n potential customers 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 professor 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.

3.2.1 Decision Epochs

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, $\xi_k = \xi_{k-1} + \min\{x_i, y_i, \delta\}$.

3.2.2 States

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 $S_K = \Big\{(t, d, q, H, U, W, (\breve{t}, \breve{q})) : H = V$ or $t + c_{di} \geq l_i, \forall i \in \{V \setminus H\}\Big\}$.

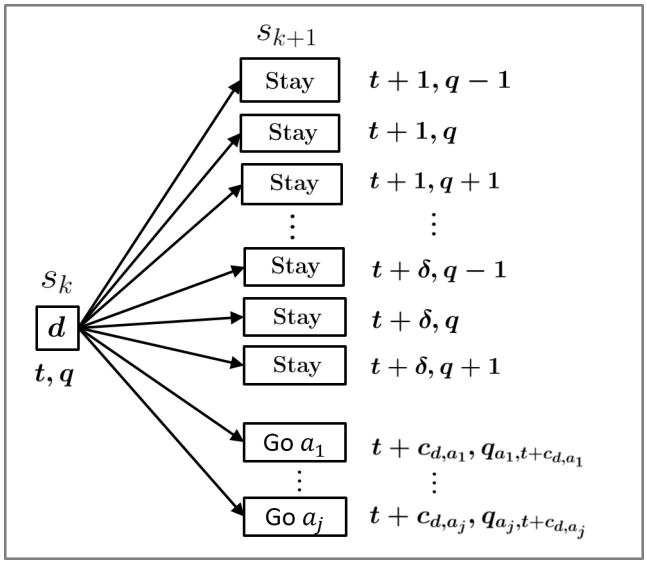

Figure 3.1: State Transition

3.2.3 Actions

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 $A(s) = \Big\{a \in V : t + c_{da} \leq l_a, a \in \{U \cup W\}\Big\}$.

3.2.4 State Transition

Figure 3.1 depicts the state transition that occurs upon action selection. 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 §3.2.5, 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 §3.2.5.

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 $q' = 0$.

3.2.5 Transition Probabilities

We derive the state transition probabilities, $P\{s_{k+1}|s_k, a\}$, for the various state-action pairs. 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 §3.2, 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 §3.2, 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.

3.2.6 Criterion and Objective

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. In this section, we first 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. 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.

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 §3.2.5, 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:

$$\begin{aligned} Stay(i,t,q) &= \sum_{q'} g_i(q'|q,1)v(i,t+1,q'), \\ Leave(i,t,q) &= \sum_{q'} P(Q_{i+1,t+c_{i,i+1}} = q')v(i+1,t+c_{i,i+1},q'). \end{aligned} \tag{3.1}$$

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:

$$v(i,t,q) = \max\big\{Stay(i,t,q), Leave(i,t,q)\big\}. \tag{3.2}$$

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 §3.2.5, there is $\sum_{q=k}^{L} g_i(q|\tilde{q},1) \leq \sum_{q=k}^{L} g_i(q|q',1)$ for all k and $q' > \tilde{q}$ ($\tilde{q}, q' \in [0, L]$). Assuming $\sum_{q=k}^{L} g_i(q|\tilde{q},n) \leq \sum_{q=k}^{L} g_i(q|q',n)$ for all k and $q' > \tilde{q}$, now we show $\sum_{q=k}^{L} g_i(q|\tilde{q},n+1) \leq \sum_{q=k}^{L} g_i(q|q',n+1)$ for all k and $q' > \tilde{q}$. Then,

$$\sum_{q=k}^{L} g_i(q|\tilde{q},n+1) = \sum_{q=k}^{L}\sum_{r=0}^{L} g_i(q|r,n) g_i(r|\tilde{q},1) \tag{3.3}$$

$$\leq \sum_{q=k}^{L}\sum_{r=0}^{L} g_i(q|r,n) g_i(r|q',1) \tag{3.4}$$

$$= \sum_{q=k}^{L} g_i(q|q',n+1), \tag{3.5}$$

Equalities (3.3) and (3.5) follow the definition of n-step transition probability. Inequality (3.4) 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:

$$\begin{aligned} & Stay(i,t,\tilde{q}) - Stay(i,t,q') \\ =& \sum_q g_i(q|\tilde{q},1)v(i,t+1,q) - \sum_q g_i(q|q',1)v(i,t+1,q) \qquad (3.6) \\ \geq& 0. \qquad (3.7) \end{aligned}$$

Equality (3.6) 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(3.7). 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 1. *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 1 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

$$\begin{aligned} & Leave(i,t,\tilde{q}) - Stay(i,t,\tilde{q}) \\ =& \sum_q P(Q_{i+1,t+c_{i,i+1}} = q)v(i+1,t+c_{i,i+1},q) - \sum_q g_i(q|\tilde{q},1)v(i,t+1,q) \qquad (3.8) \\ \geq& 0. \end{aligned}$$

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

$$
\begin{aligned}
& Leave(i,t,q') - Stay(i,t,q') \\
=& \sum_q P(Q_{i+1,t+c_{i,i+1}} = q) v(i+1, t+c_{i,i+1}, q) - \sum_q g_i(q|q',1) v(i,t+1,q) && (3.9) \\
\geq& \sum_q P(Q_{i+1,t+c_{i,i+1}} = q) v(i+1, t+c_{i,i+1}, q) - \sum_q g_i(q|\tilde{q},1) v(i,t+1,q) && (3.10) \\
\geq& 0. && (3.11)
\end{aligned}
$$

Equality (3.9) follows by definition. Inequality (3.10) results from Lemma 2, and Inequality (3.11) 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 3.1 and 3.2 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

3.1, 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 3.2. 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 3.2, the first column includes the arrival times reported in the sixth and ninth columns of Table 3.1. 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

Table 3.1: Professor 1 with time window [0, 40]

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				

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 3.1, for an arrival time of 10, it is optimal to balk and go to professor 2 because E[Stay at 1] $= 30 < E$[Leave 1]$= 34$. For an arrival time of 25, with E[Stay at 1] $= 6 > E$[Leave 1] $= 2$, it is optimal to stay in queue at professor 1.

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*

Table 3.2: Professor 2 with time window [0, 60]

Arrive Time A_2	Queue Dist. $q_{2t}, P(q_{2t})$	Wait Time Dist. $\omega_{2t}, P(\omega_{2t})$	Meet	R_2	$E[R_2]$
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	

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.

3.4 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. (2014).

In §3.4.1, we present a heuristic a-priori-route policy for estimating the reward-to-go. In §3.4.2, we briefly summarize existing rollout algorithm methodology from the literature. In §3.4.3, we propose a compound rollout algorithm that is based on a partitioned action space and hierarchal application of heuristic policies.

3.4.1 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 (2008b) 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 $\big\{\{U \cup W\} \setminus \{d\}\big\}$. 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 §3.4.1.1, 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 in Algorithm 3.1. Line 3 of Algorithm 2.2 initializes the search with an a-priori-route policy from state s with randomly ordered professors. Line 5 states the termination criteria for the VND. As mentioned in §3.5.2, 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 18 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 19 through Line 23 update the current solution and the value of *level*. Line 25 advances the interaction counters.

We denote by $V^{\pi(\nu)}(s)$ the expected value of following a-priori-route policy

Algorithm 3.1 Variable Neighborhood Descent (VND) for the DOPTW

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
      $improving \leftarrow false$
    end if
  end while
  if $V^{\pi''(\nu)}(s) > V^{\pi(\nu)}(s)$ **then**
    $\pi(\nu) \leftarrow \pi''(\nu)$
    $level \leftarrow 1$
  else
    $level \leftarrow level + 1$
  end if
  $iter \leftarrow iter + 1$
end while

$\pi(\nu)$ 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.

3.4.1.1 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 §3.2.5. Let $f^i_{q0}(n)$ 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 $P^i_{qj}(n)$ 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

$$P^i_{q0}(n) = P^i_{00}(n-1)f^i_{q0}(1) + \ldots + P^i_{00}(1)f^i_{q0}(n-1) + f^i_{q0}(n). \tag{3.12}$$

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

$$\begin{aligned} f^i_{q0}(1) =& P^i_{q0}(1) \\ & \vdots \\ f^i_{q0}(n) =& P^i_{q0}(n) - f^i_{q0}(1)P^i_{00}(n-1) - \ldots - f^i_{q0}(n-1)P^i_{00}(1). \end{aligned} \tag{3.13}$$

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 $q = 0, \ldots, L$.

3.4.2 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 §3.4.1. 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. (2014) 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 Algorithm 3.1 in §3.4.1. 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}:

$$E\big[V^{\pi(\nu,s_{k+1})}(s_{k+1})|s_k,a\big] = \sum_{s_{k+1}\in S(s_k,a)} P\{s_{k+1}|s_k,a\} \times V^{\pi(\nu,s_{k+1})}(s_{k+1}), \qquad (3.14)$$

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 3.2a 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.

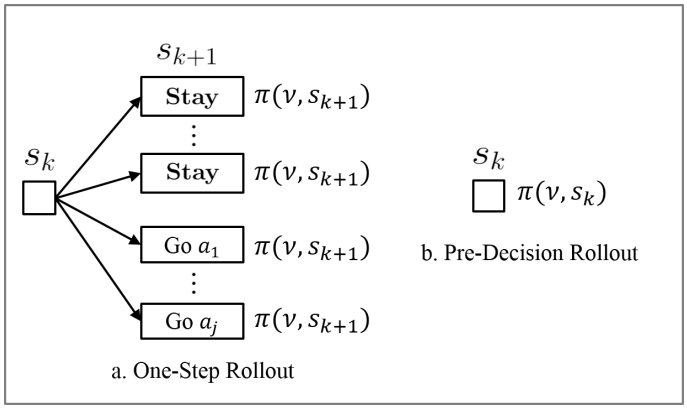

Figure 3.2: Rollout Decision Rule

The formalization of the pre-decision rollout is motivated by the computational issues associated with one-step rollout (Goodson et al. (2014)). As shown in Figure 3.2b, 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, 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.

3.4.3 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,

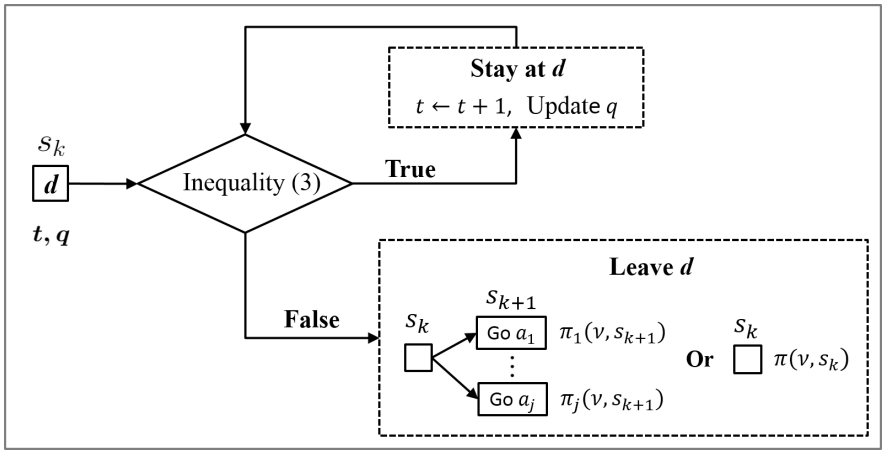

Figure 3.3: Compound Rollout

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 §3.2, 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

$$P(\Omega_{dt} < l_d - t|q,t) \times r_d + \sum_{\omega_{dt} \leq l_d - t} P(\Omega_{dt} = \omega_{dt}) V_{go}(t + \omega_{dt}) > V_{go}(t), \tag{3.15}$$

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 §3.4.1.1. The comparison in Inequality (3.15) 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 (3.15) with

$$P(\Omega_{dt} < l_d - t|q,t) \times r_d + V_{go}(t + E[\Omega_{dt}|q,t]) > V_{go}(t). \tag{3.16}$$

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 3.3 shows, the salesperson will stay and wait at professor d while Inequality (3.16) 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 §3.2.5), and re-evaluate Inequality (3.16) with the updated information. The salesperson will leave professor d once Inequality (3.16) 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 3.3 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 3.2, 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 (3.16) 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 3.2 Compound Rollout

1: **if** $d \in A(s_k)$ and Inequality (3.16) 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 $\nu_2 \in \pi(\nu, s_k)$
7: **end if**

3.5 Computational Experiments

In this section, we present computational results from applying the compound rollout procedure of §3.4.3. In §3.5.1, we detail the generation of problem instances for the DOPTW from existing benchmark problems. In §3.5.2, we provide implementation details, and in §4.4.2 and §3.5.4, we present and discuss our computational results.

3.5.1 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.

3.5.2 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 §3.4.1, 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 §3.2.5. 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 $i \in V$.

3.5.3 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. 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 3.3 through 3.5 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

Table 3.3: DOPTW Results for R Instances with 20 Professors

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

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 ($Gap = \frac{Dynamic - A\ Priori}{A\ Priori} \times 100\%$).

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

Table 3.4: DOPTW Results for C Instances with 20 Professors

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 3.5: DOPTW Results for RC 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

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 §3.5.2, 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 professors, 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 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 3.2 is implemented when using one-step rollout in the second stage, and line 6 is implemented when using

Table 3.6: Comparison between Compound and Pre-Decision Rollout

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%]		

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 3.6, 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.

3.5.4 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 3.4 and 3.5. 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, 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 3.4, 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

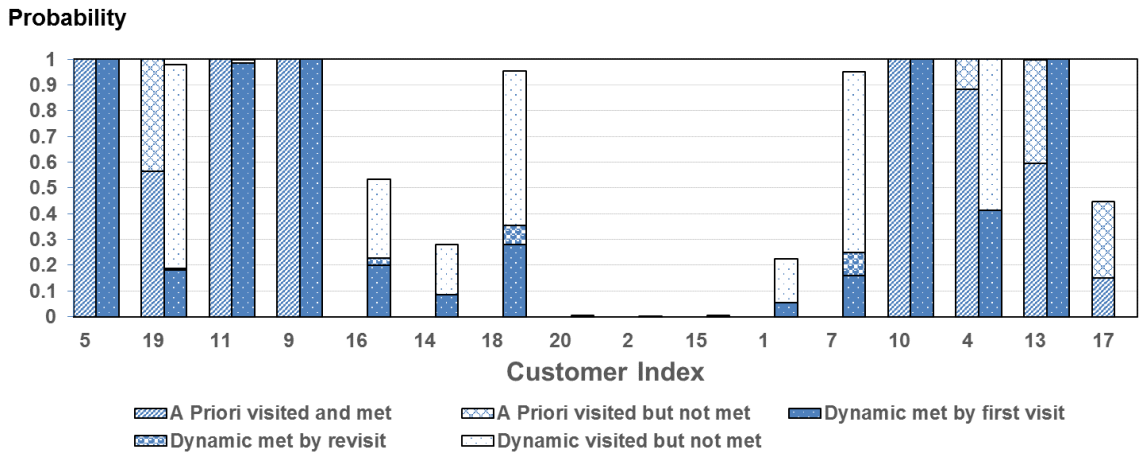

Figure 3.4: Compound Rollout vs. A Priori for Dataset R107

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 3.5 shows, professors 16, 4, 15, 13, 9, and 8 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.

3.6 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) character-

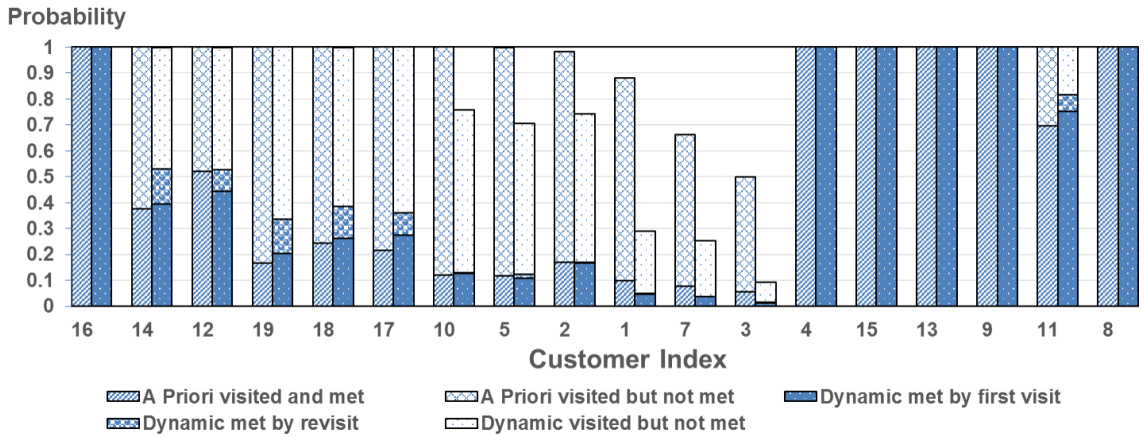

Figure 3.5: Compound Rollout vs. A Priori for Dataset C204

ized 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 §4.1, 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.

CHAPTER 4
MULTI-PERIOD ORIENTEERING WITH UNCERTAIN ADOPTION LIKELIHOOD AND WAITING AT CUSTOMERS

4.1 Introduction

In this chapter, we introduce a multi-period orienteering problem where a salesperson visits customers over a multi-period horizon to observe and influence the chance of customer adoption. Each customer's likelihood of adopting the product is uncertain and evolves stochastically over the multi-period horizon. The salesperson's visits to a customer serve two primary purposes: (i) to potentially increase the adoption likelihood by increasing the customer's awareness and knowledge of the product, and (ii) to gauge the customer's adoption likelihood. Thus, the salesperson may revisit a customer after a previous meeting, trying to develop a stronger relationship that leads to greater chance for adoption (personal communication, Tracy Ward, Senior Learning Technology Representative at McGraw-Hill Education, April 20, 2015). Further, the customers' adoption probabilities evolve over time, influenced by other peers or the efforts of the salesperson's competitors, whether or not the customer is visited by the salesperson. When meeting with a customer, the salesperson may not be able to fully observe the customer's adoption likelihood. Instead, she estimates the underlying likelihood based on the information obtained via communicating with the customer, but the estimation may not be accurate.

In each period, we consider that customers have limited availability and can only be accessed in pre-determined time windows (e.g., office hours of professors).

To meet with a customer, the salesperson must arrive within the customer's time windows. However, arriving within a customer's time windows does not guarantee that the salesperson will meet with the customer. Upon arrival, the salesperson may find a queue of others waiting or meeting with the customer, in which case the salesperson has to wait in the queue if she wants to meet the customer. If observing a queue after arrival, the salesperson needs to decide whether to join the queue or balk (leave immediately after arrival). While waiting in the queue, the salesperson needs to decide whether to stay in the queue or renege (leave the queue after waiting for a while). We assume that the queue length is unknown to the salesperson before her arrival at a customer and the wait time the salesperson needs to spend in the queue is uncertain.

Due to the stochastically-evolving adoption likelihood and uncertain ability to visit customers as a result of the wait times induced by queues, in general, this situation can be viewed as a multi-period orienteering problem with uncertain reward collection. Each period, the salesperson must decide which customers to schedule and in which order to visit them, knowing that, due to uncertain wait times, the salesperson may not be able to meet customers even if they are on the schedule. We propose two models for the problem. First, we model the problem as a Markov decision process (MDP), assuming that each customer's adoption likelihood is perfectly known by the salesperson at all epochs. We then model the problem as a partially observed Markov decision process (POMDP), which accounts for the fact that the salesperson can only estimate the adoption likelihood and the accuracy of the esti-

mation is uncertain. The objective of both models is to maximize the expected sales accrued from product adoptions at the end of the horizon.

This study makes the following contributions to the literature. First, we introduce a new multi-period orienteering problem motivated by salespeople visiting customers. The salespeople may experience queues at customers and the likelihood that each customer will adopt the product is uncertain and not fully observed by the salesperson. Second, the formulation explicitly accounts for the stochastic evolution of each customer's adoption likelihood resulting from the salesperson and her competitors' marketing efforts as well as the influence from peers of the customer. Third, we propose a heuristic approach to facilitate decision making. The approach iteratively solves an assignment problem to determine which customers to visit in a period and a routing problem to visit the selected customers within the period.

In §4.2, we present the MDP model (§4.2.1) and the POMDP model (§4.2.2). We provide an example of the POMDP to illustrate the computational complexity in obtaining an optimal solution. In §4.3, we propose the heuristic solution approach, and in §4.4, we demonstrate the effectiveness of the heuristic approach via computational results.

4.2 Problem Formulation

We define the problem on a complete graph (N, E), where the node set N consists of n potential customers that the salesperson may visit over T time periods, and the set E consists of edges associated with each pair of nodes. We consider

deterministic travel time on edge (i, j), denoted as c_{ij}. For $i = j$, we set $c_{ij} = 0$. A time window $[e_i^t, l_i^t]$ is associated with each customer $i \in N$ in period t. We denote by r_i the reward collected by the salesperson if customer $i \in N$ adopts the product. In this section, we present two models for the problem. In §4.2.1, we first model the problem as a Markov decision process (MDP) assuming that each customer's adoption likelihood is perfectly known by the salesperson at all epochs. In §4.2.2, we present a partially observed Markov decision process (POMDP) that models the situation in which the salesperson may not perfectly gauge the customer's adoption likelihood.

4.2.1 Markov Decision Process

In this section, we present a model for the problem in which we assume that the salesperson can fully observe each customer's adoption likelihood when meeting with the customer. We consider a discrete-time planning horizon $0, 1, \ldots, T$. In the MDP, a decision epoch occurs at each period $t \in [0, T]$. Let X_t^i be the discrete random variable whose finite set of ordered states represents the chance at the beginning of period t that customer i will adopt the product in period T. We denote a realization of X_t^i as x_t^i and assume that x_t^i takes on a value from the finite set $\mathfrak{P}$ of categories which can be mapped to an ordered set of probabilities $\{\alpha_k\}$, where $0 \leq \alpha_k \leq 1$ and $\alpha_k < \alpha_{k+1}$ for $k = 1, \ldots, |\mathfrak{P}|$. For example, $\mathfrak{P} = \{\text{certainly not}, \text{unlikely}, \text{maybe}, \text{very likely}, \text{certainly will}\}$ could be mapped to likelihoods $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\} = \{0, 0.25, 0.5, 0.75, 1\}$.

The state of the system includes information needed for the salesperson to

decide which customers to schedule in the current period. Let $[x_t^i]_{i\in N}$ be a vector of customers' adoption likelihood in period t. We represent the system state s_t by $[x_t^i]_{i\in N}$. The state space S is defined on $[\alpha_1, \alpha_2, \ldots, \alpha_{|\mathfrak{P}|}]^N$. The initial state is $s_0 = [x_0^i]_{i\in N}$, where x_0^i is the original adoption likelihood associated with customer i before the salesperson's marketing efforts.

At each decision epoch t, an action a_t taken by the salesperson indicates which customers to visit in the period, the order to visit them, and the queueing strategy (whether to wait and how long to wait) at the visited customers. Thus, the action space for state s_t, $A(s_t)$, includes all the policies for scheduling and routing customers in period t. In this study, we consider a restricted policy class in which the sequencing and queueing strategies are determined a priori given a set of scheduled customers to visit. We refer the readers to §2 for discussions about the a priori routing and queueing strategies. This restricted policy class allows us to characterize an action a_t by the set $\{v_t^1, v_t^2, \ldots, v_t^N\}$, where v_t^i is a binary variable such that $v_t^i = 1$ if customer i is selected for a visit in period t and $v_t^i = 0$ otherwise.

The system evolves in the following manner. At decision epoch t with state $s_t = [x_t^i]_{i\in N}$, an action a_t is selected to visit customers in period t, initiating the stochastic transition from state s_t to state $s_{t+1} = [x_{t+1}^i]_{i\in N}$. Due to the uncertain wait time and the limited availability associated with each customer, the salesperson may not be able to meet with all the selected customers. Let W_t^i be a random variable indicating whether the salesperson met with customer i in period t and w_t^i be a realization of W_t^i. If the salesperson meets with customer i in period t, $w_t^i = 1$,

otherwise $w_t^i = 0$. The distribution of W_t^i depends on the action (routing policy) a_t taken in period t, denoted as $\Pr\{W_t^i = w_t^i | a_t\}$. In this study, we assume that the routing policy taken by the salesperson in each decision epoch is an a priori routing policy, which specifies an a priori route (a pre-planned order of visits to customers) and the decision rules for executing recourse actions in the a priori route. We refer the readers to §3.4.1 for a detailed discussion about a priori route policies.

Let $\Pr\{s_{t+1}|s_t, a_t\}$ be the transition probability from state s_t to s_{t+1} when taking action a_t in s_t. The distribution of $\Pr\{s_{t+1}|s_t, a_t\}$ is defined over $\mathfrak{P}^N$ possible states and is specified by $\prod_{i\in N} \Pr\{X_{t+1}^i = x_{t+1}^i | X_t^i = x_t^i, a_t\}$ for all $[x_{t+1}^1, \ldots, x_{t+1}^i] \in \mathfrak{P}^N$. Specifically,

$$\begin{aligned} &\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, a_t\} \\ &= \sum_{w_t^i \in \{0,1\}} \Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, w_t^i\} \Pr\{W_t^i = w_t^i | a_t\}, \end{aligned} \tag{4.1}$$

where we assume $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, w_t^i\}$ is known and $\Pr\{W_t^i = w_t^i | a_t\}$ is given by the likelihood that customer i is met by the salesperson when the a priori route policy a_t is executed in period t.

We represent by $R_t(s_t, a_t)$ the reward collected in period t given state s_t and action a_t. Because the salesperson can only collect reward from customers at the end of the horizon T, we have $R_t(s_t, a_t) = 0$ for any $t < T$. The expected reward at period T, $R_T(s_T, a_T)$, is given by $\sum_{i\in N} r_i \alpha_k^i$, where $\alpha_k^i (1 \le k \le \mathfrak{P})$ is the adoption probability of customer i after taking action a_T at state s_T.

We denote by π a policy for the problem, which is a sequence of decision rules at each epoch t: $\pi = (\rho_0^\pi(s_0), \rho_1^\pi(s_1), \ldots, \rho_T^\pi(s_T))$, where $\rho_t^\pi(s_t) : s_t \mapsto A(s_t)$ is

a function specifying the action to select at decision epoch t while following policy π. The objective is to find a Markovian decision policy π that maximizes $V_0^\pi = E_\pi\big[\sum_{t=0}^{T} R_t(s_t, \rho_t^\pi(s_t))|s_0\big] = E_\pi\big[R_T(s_T, \rho_T^\pi(s_T))|s_0\big]$.

We note that even though a restricted policy class is considered in the MDP, the action space is still intractably large, because besides sequencing customers, queueing strategy has to be developed explicitly for each customer in an a priori policy. As such, the state space will explode and obtaining an optimal solution for the MDP becomes computationally challenging.

4.2.2 Partially Observed Markov Decision Process

Unlike in §4.2.1 where we assume that the state information is fully observable, in this section, we consider the case that the adoption probability is partially observed. We model the problem as a partially observed markov decision process (POMDP). The POMDP has been widely used to model problems where only partial knowledge of a system state is available. A POMDP is a generalization of the fully observed Markov decision process (MDP). It allows imperfect information about the state of the system. Instead of observing the state s perfectly as in the MDP, the salesperson now perceives an observation of s. The observations that the salesperson receives depend on the state s and the action a taken at the state, and is drawn according to a function Q that specifies the distribution of observations given the action a taken at state s. For a detailed discussion of POMDP, we refer the readers to Bertsekas (2005) and Spaan (2012). For surveys on the POMDP, see Monahan (1982) and Lovejoy

(1991).

The *core process* of our POMDP is the evolution of the true adoption likelihood at each customer $i \in N$, i.e., the MDP presented in §4.2.1. However, we now assume that the salesperson can only estimate the customer's adoption probability. Following the notation in §4.2.1, we use X_t^i to represent the true adoption likelihood of customer i. We denote the salesperson's estimation of X_t^i by a random variable Y_t^i and let y_t^i be a realization of Y_t^i. We consider that there is a probabilistic relationship between X_t^i and Y_t^i, which depends on the action a_{t-1} taken in period $t-1$ that induces the transition from X_{t-1}^i to X_t^i. We denoted the probabilistic relationship by a matrix $Q^{it}\big(a_{t-1}\big) = [q_{\alpha\beta}^{it}(a_{t-1})]$, where $q_{\alpha\beta}^{it}(a_{t-1}) \equiv \Pr\{Y_t^i = \beta | X_t^i = \alpha, a_{t-1}\}$.

The support of the conditional distribution of Y_t^i is given by $\{\mathfrak{P} \cup \{?\}\}$, where $y_t^i =?$ denotes that the salesperson has not met with customer i in period $t-1$. The probabilistic relationship $q_{\alpha\beta}^{it}\big(a_{t-1}\big)$ is derived as

$$\begin{aligned} &\Pr\{Y_t^i = \beta | X_t^i = \alpha, a_{t-1}\} \\ &= \sum_{w_{t-1}^i \in \{0,1\}} \Pr\{Y_t^i = \beta | X_t^i = \alpha, w_{t-1}^i\} \Pr\{W_{t-1}^i = w_{t-1}^i | a_{t-1}\}, \end{aligned} \tag{4.2}$$

where we assume that $\Pr\{Y_t^i = \beta | X_t^i = \alpha, w_{t-1}^i\}$ is known and $\Pr\{W_{t-1}^i = w_{t-1}^i | a_{t-1}\}$ is given by the likelihood of meeting customer i when taking action a_{t-1} in period $t-1$. We note that $\Pr\{Y_t^i =? | X_t^i = \alpha, w_{t-1}^i = 0\} = 1$, for all $\alpha \in \mathfrak{P}$.

As noted in Bertsekas (2005), the problems with imperfect state information can be reduced to ones with perfect state information and thereby solved by dynamic programming algorithms. In the following discussion, we provide two different formulations for the POMDP. In §4.2.2.1, we reduce the POMDP to a MDP in which

the system maintains memory of all past observations and actions. In §4.2.2.2, we formulate the POMDP by taking advantage of the fact that the conditional probability of being in each possible state at a decision epoch is a sufficient statistic of the past observations and actions.

4.2.2.1 Formulation I

In this section, we formulate the POMDP as a MDP in which the system state stores the history of all previous observations and actions. Let $d_t^i \equiv (y_1^i, \ldots, y_t^i, a_1, \ldots, a_{t-1})$ be the vector that consists of all past actions taken and observations made at customer i before period t. By definition, $d_{t+1}^i = d_t^i \cup \{a_t, y_{t+1}^i\}$ for all $t \in [1, T-1]$. We define $D_t \equiv [d_t^i]_{i \in N}$ as the *information vector* of the system at period t. The state of the system is given by D_t, which represents all the information available for decision making at period t. By taking action a at state D_t, a set of realizations $[y_t^i]_{i \in N}$ is observed and the system transitions to the next state D_{t+1}. The state transition probability, $\Pr\{D_{t+1}|D_t, a_t\}$, is given by the joint distribution of observations y_{t+1}^i given the action a_t and history information d_t^i, denoted as $\prod_{i=1}^N \Pr\{y_{t+1}^i|d_t^i, a_t\}$. The probability $\Pr\{y_{t+1}^i|d_t^i, a_t\}$ is computed as

$$\Pr\{y_{t+1}^i|d_t^i, a_t\} = \sum_{\gamma \in \mathfrak{P}} q_{\gamma, y_{t+1}^i}^{i,t+1}(a_t) \cdot \sum_{\kappa \in \mathfrak{P}} p_{\kappa\gamma}^{i,t+1}(a_t) \Pr\{X_t^i = \kappa|d_t^i\}, \tag{4.3}$$

where $p_{\kappa\gamma}^{i,t+1}(a_t) \equiv \Pr\{X_{t+1}^i = \gamma|X_t^i = \kappa, a_t\}$ and $\Pr\{X_t^i = \kappa|d_t^i\}$ is the posterior distribution of customer i's true adoption likelihood given the information d_t^i. According

to the Bayes' rule, we derive $\Pr\{X_t^i = \kappa | d_t^i\}$ as

$$\begin{aligned}
&\Pr\{X_t^i = \alpha | d_t^i\} = \Pr\{X_t^i = \alpha | d_{t-1}^i, a_{t-1}, y_t^i\} \\
&= \frac{q_{\alpha,y_t^i}^{i,t}(a_{t-1}) \sum_{\kappa \in \mathfrak{P}} p_{\kappa\alpha}^{i,t}(a_{t-1}) \Pr\{X_{t-1}^i = \kappa | d_{t-1}^i\}}{\sum_{\gamma \in \mathfrak{P}} q_{\gamma,y_t^i}^{i,t}(a_{t-1}) \cdot \sum_{\kappa \in \mathfrak{P}} p_{\kappa\gamma}^{i,t}(a_{t-1}) \Pr\{X_{t-1}^i = \kappa | d_{t-1}^i\}}, \forall i \in N, \alpha \in \mathfrak{P}. \qquad (4.4)
\end{aligned}$$

For $t = 0$, we assume that $\Pr\{X_0^i = \kappa | d_0^i\}$ is given by $\Pr\{X_0^i = \kappa\}$.

Let $R_T(D_T, s_T)$ represent the reward collected at the end of the horizon given the state D_T and underlying adoption probabilities $s_T = [x_T^i]_{i \in N}$. The expected reward collected at the end of the horizon is then given by $E_{s_T}[R_T(D_T, s_T)]$. We note that $\Pr\{s_T | D_T\} = \prod_{i=1}^N \Pr\{x_T^i | d_T^i\}$. The objective is seek a policy π that maximizes $E_\pi\big[E_{s_T}[R_T^\pi(D_T, s_T)]\big]$.

An Example

We consider an example with two customers and two periods. For illustrative purposes, we define $\mathfrak{P} = \{L, H\}$, where L represents a low chance ($\alpha_1 = 0.3$) that the customer will adopt and H stands for a high chance ($\alpha_2 = 0.7$). Customer 1 is associated with value 10 and customer 2 is associated with value 15. We assume that in each period, the salesperson can only visit one customer. The probability distribution $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, w_t^i\}$ for $i = 1, 2$ is given in Table 4.1.

For $t = 1, 2$, we assume that the conditional probability of meeting a customer is given in Table 4.2. According to Equation (4.1), we derive the transition distribution of the core process $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, a_t\}$, which is presented in Table 4.3.

In Table 4.4, we define the conditional distribution of $\Pr\{Y_t^i = \beta | X_t^i = \alpha, w_{t-1}^i\}$

Table 4.1: Distribution of $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, w_t^i\}$

X_{t+1}^i	X_t^i	w_t^i	**Prob.**	X_{t+1}^i	X_t^i	w_t^i	**Prob.**
L	L	0	0.7	H	L	0	0.3
L	L	1	0.6	H	L	1	0.4
H	H	0	0.6	L	H	0	0.4
H	H	1	0.7	L	H	1	0.3

Table 4.2: Distribution of $\Pr\{W_t^i = w_t^i | a_t\}$

v_t^1	v_t^2	w_t^1	$\Pr\{W_t^1 = w_t^1 \| a_t\}$	v_t^1	v_t^2	w_t^2	$\Pr\{W_t^2 = w_t^2 \| a_t\}$
1	0	1	0.75	0	1	1	0.75
1	0	0	0.25	0	1	0	0.25
0	1	1	0	1	0	1	0
0	1	0	1	1	0	0	1

Table 4.3: Distribution of $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, a_t\}$

i	X_{t+1}^i	X_t^i	v_t^1, v_t^2	**Prob.**	i	X_{t+1}^i	X_t^i	v_t^1, v_t^2	**Prob.**
1	L	L	0, 1	0.7	1	H	L	0, 1	0.3
1	L	L	1, 0	0.625	1	H	L	1, 0	0.375
1	H	H	0, 1	0.6	1	L	H	0, 1	0.4
1	H	H	1, 0	0.675	1	L	H	1, 0	0.325
2	L	L	1, 0	0.7	1	H	L	1, 0	0.3
2	L	L	0, 1	0.625	1	H	L	0, 1	0.375
2	H	H	1, 0	0.6	1	L	H	1, 0	0.4
2	H	H	0, 1	0.675	1	L	H	0, 1	0.325

Table 4.4: Distribution of $\Pr\{Y_t^i = \beta | X_t^i = \alpha, w_{t-1}^i\}$

Y_1^i	X_1^i	w_0^i	**Prob.**	Y_1^i	X_1^i	w_0^i	**Prob.**	Y_1^i	X_1^i	w_0^i	**Prob.**
?	H	0	1	H	H	0	0	L	H	0	0
?	L	0	1	H	L	0	0	L	L	0	0
?	H	1	0	H	H	1	0.7	L	H	1	0.3
?	L	1	0	L	L	1	0.6	H	L	1	0.4
Y_2^i	X_2^i	w_1^i	**Prob.**	Y_2^i	X_2^i	w_1^i	**Prob.**	Y_2^i	X_2^i	w_1^i	**Prob.**
?	H	0	1	H	H	0	0	L	H	0	0
?	L	0	1	H	L	0	0	L	L	0	0
?	H	1	0	H	H	1	0.75	L	H	1	0.25
?	L	1	0	H	L	1	0.55	L	L	1	0.45

Table 4.5: Distribution of $\Pr\{Y_t^i = \beta | X_t^i = \alpha, a_{t-1}\}$

i	Y_1^i	X_1^i	v_0^1, v_0^2	**Prob.**	i	Y_1^i	X_1^i	v_0^1, v_0^2	**Prob.**	i	Y_1^i	X_1^i	v_0^1, v_0^2	**Prob.**
1	?	H	0, 1	1	1	H	H	0, 1	0	1	L	H	0, 1	0
1	?	L	0, 1	1	1	H	L	0, 1	0	1	L	L	0, 1	0
1	?	H	1, 0	0.25	1	H	H	1, 0	0.525	1	L	H	1, 0	0.225
1	?	L	1, 0	0.25	1	H	L	1, 0	0.45	1	L	L	1, 0	0.3
2	?	H	1, 0	1	2	H	H	1, 0	0	2	L	H	1, 0	0
2	?	L	1, 0	1	2	H	L	1, 0	0	2	L	L	1, 0	0
2	?	H	0, 1	0.25	2	H	H	0, 1	0.525	2	L	H	0, 1	0.225
2	?	L	0, 1	0.25	2	H	L	0, 1	0.45	2	L	L	0, 1	0.3
i	Y_2^i	X_2^i	v_1^1, v_1^2	**Prob.**	i	Y_2^i	X_2^i	v_1^1, v_1^2	**Prob.**	i	Y_2^i	X_2^i	v_1^1, v_1^2	**Prob.**
1	?	H	0, 1	1	1	H	H	0, 1	0	1	L	H	0, 1	0
1	?	L	0, 1	1	1	H	L	0, 1	0	1	L	L	0, 1	0
1	?	H	1, 0	0.25	1	H	H	1, 0	0.562	1	L	H	1, 0	0.188
1	?	L	1, 0	0.25	1	H	L	1, 0	0.412	1	L	L	1, 0	0.338
2	?	H	1, 0	1	2	H	H	1, 0	0	2	L	H	1, 0	0
2	?	L	1, 0	1	2	H	L	1, 0	0	2	L	L	1, 0	0
2	?	H	0, 1	0.25	2	H	H	0, 1	0.562	2	L	H	0, 1	0.188
2	?	L	0, 1	0.25	2	H	L	0, 1	0.412	2	L	L	0, 1	0.338

for $i = 1, 2$. According to Equation (4.2), the probabilistic relationship between the estimation of chance of adoption Y_t^i and the true underlying chance X_t^i given the actions taken in period $t-1$, $\Pr\{Y_t^i = \beta | X_t^i = \alpha, a_{t-1}\}$, is derived in Table 4.5.

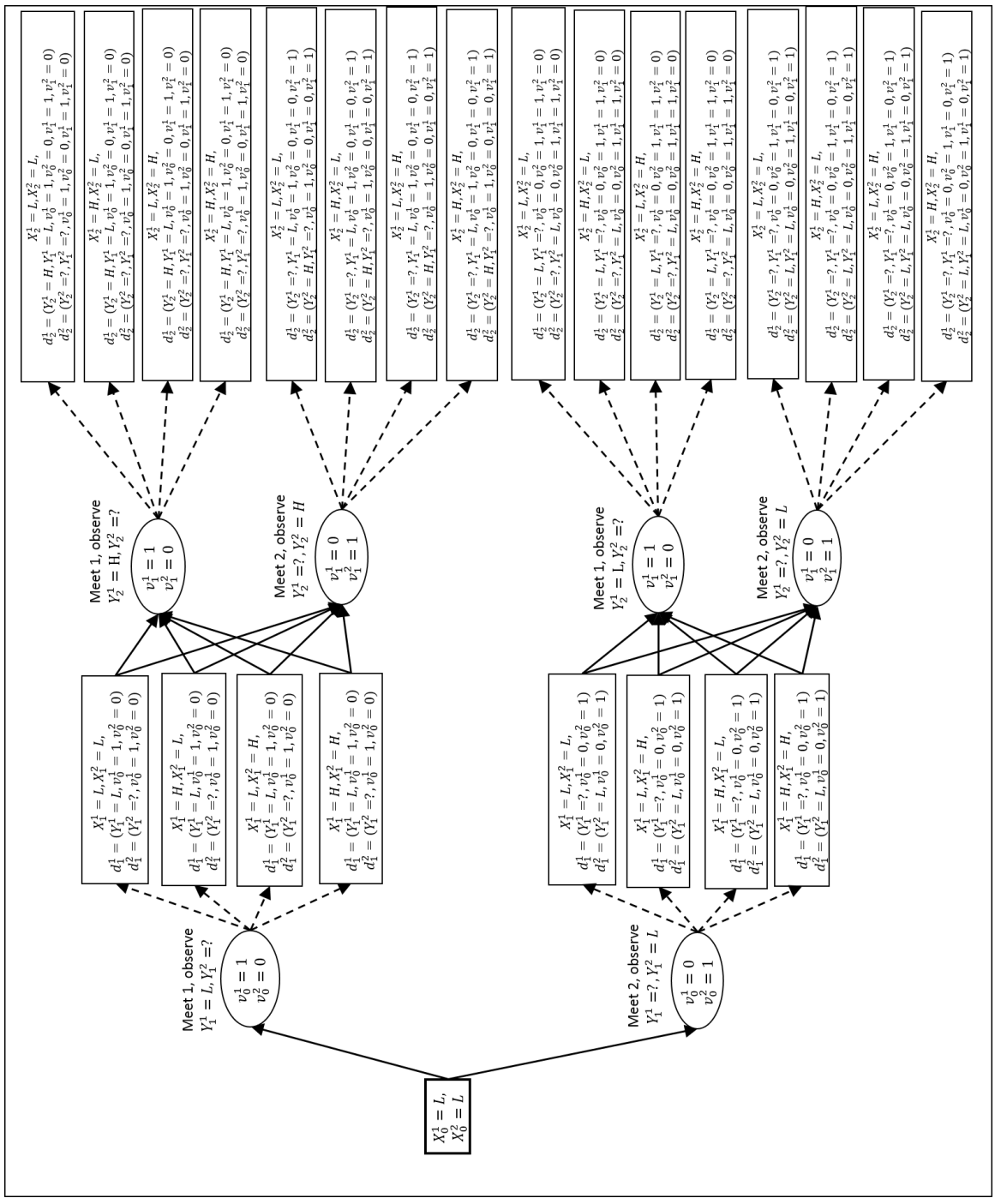

Figure 4.1: An Example of the POMDP

For illustrative purposes, we consider a scenario as depicted in Figure 4.1. We assume that the initial adoption likelihood for each customers is L. If the salesperson visits customer 1 in period 0, we assume that she will meet the customer and estimate the customer's adoption likelihood in period 1 as L. She will either visit customer 2 or revisit customer 1 in period 1. If she revisits customer 1 in period 1, we assume that she estimates customer 1's adoption likelihood in period 2 as H. If she visits customer 2 in period 1 after visiting customer 1 in period 0, we assume that she estimates customer 2's adoption likelihood in period 2 as H.

If the salesperson visits customer 2 in period 0, we assume that she will meet with customer 2 and estimate customer 2's adoption likelihood in period 1 as L. She will either visit customer 1 or revisit customer 2 in period 1. If she visits customer 1 in period 1, we assume that she estimates customer 1's adoption likelihood in period 2 as L. If she revisits customer 2 in period 1, we assume that she estimates customer 2's adoption likelihood in period 2 as L. According to Equation (4.4), we compute the transition probability of the core process in Table 4.6 and Table 4.7 and compute $\Pr\{X_t = x_t | D_t\}$ in Table 4.8.

Given that $r_1 = 10$, $r_2 = 15$, $\alpha_1 = 0.3$ and $\alpha_2 = 0.7$, for any d_2^1, d_2^2, we have $\bar{R}(X_2^1 = L, X_2^2 = L, d_2^1, d_2^2) = 10 \times 0.3 + 15 \times 0.3 = 7.5$, $\bar{R}(X_2^1 = H, X_2^2 = L, d_2^1, d_2^2) = 10 \times 0.7 + 15 \times 0.3 = 11.5$, $\bar{R}(X_2^1 = L, X_2^2 = H, d_2^1, d_2^2) = 10 \times 0.3 + 15 \times 0.7 = 14.5$ and $\bar{R}(X_2^1 = H, X_2^2 = H, d_2^1, d_2^2) = 10 \times 0.7 + 15 \times 0.7 = 17.5$. Then we compute $\bar{R}(s_t)$ for $t = 1, 2$ as,

Table 4.6: Distribution of $\Pr\{X_1^i = \alpha | d_1^i\}$

i	α	d_1^i	$\Pr\{X_1^i = \alpha \mid d_1^i\}$
1	L	$Y_1^1 = L, v_0^1 = 1, v_0^2 = 0$	0.69
1	H	$Y_1^1 = L, v_0^1 = 1, v_0^2 = 0$	0.31
1	L	$Y_1^1 =?, v_0^1 = 0, v_0^2 = 1$	0.7
1	H	$Y_1^1 =?, v_0^1 = 0, v_0^2 = 1$	0.3
2	L	$Y_1^2 =?, v_0^1 = 1, v_0^2 = 0$	0.7
2	H	$Y_1^2 =?, v_0^1 = 1, v_0^2 = 0$	0.3
2	L	$Y_1^2 = L, v_0^1 = 0, v_0^2 = 1$	0.69
2	H	$Y_1^2 = L, v_0^1 = 0, v_0^2 = 1$	0.31

Table 4.7: Distribution of $\Pr\{X_2^i = \alpha | d_2^i\}$

i	α	d_2^i	$\Pr\{X_2^i = \alpha \mid d_2^i\}$
1	L	$Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0$	0.454
1	H	$Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0$	0.546
1	L	$Y_2^1 =?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1$	0.607
1	H	$Y_2^1 =?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1$	0.393
1	L	$Y_2^1 = L, Y_1^1 =?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0$	0.605
1	H	$Y_2^1 = L, Y_1^1 =?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0$	0.395
1	L	$Y_2^1 =?, Y_1^1 =?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1$	0.61
1	H	$Y_2^1 =?, Y_1^1 =?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1$	0.39
2	L	$Y_2^2 =?, Y_1^2 =?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0$	0.61
2	H	$Y_2^2 =?, Y_1^2 =?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0$	0.39
2	L	$Y_2^2 = H, Y_1^2 =?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1$	0.561
2	H	$Y_2^2 = H, Y_1^2 =?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1$	0.439
2	L	$Y_2^2 =?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0$	0.607
2	H	$Y_2^2 =?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0$	0.393
2	L	$Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1$	0.632
2	H	$Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1$	0.368

Table 4.8: Distribution of $\Pr\{X_t = x_t | D_t\}$

t	X_t^1	X_t^2	d_t^1, d_t^2	$\Pr\{X_t = x_t \mid D_t\}$
1	L	L	$d_1^1 = \{Y_1^1 = L, v_0^1 = 1, v_0^2 = 0\}$ $d_1^2 = \{Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0\}$	0.483
1	H	L	$d_1^1 = \{Y_1^1 = L, v_0^1 = 1, v_0^2 = 0\}$ $d_1^2 = \{Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0\}$	0.217
1	L	H	$d_1^1 = \{Y_1^1 = L, v_0^1 = 1, v_0^2 = 0\}$ $d_1^2 = \{Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0\}$	0.207
1	H	H	$d_1^1 = \{Y_1^1 = L, v_0^1 = 1, v_0^2 = 0\}$ $d_1^2 = \{Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0\}$	0.093
1	L	L	$d_1^1 = \{Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1\}$ $d_1^2 = \{Y_1^2 = L, v_0^1 = 0, v_0^2 = 1\}$	0.483
1	L	H	$d_1^1 = \{Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1\}$ $d_1^2 = \{Y_1^2 = L, v_0^1 = 0, v_0^2 = 1\}$	0.217
1	H	L	$d_1^1 = \{Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1\}$ $d_1^2 = \{Y_1^2 = L, v_0^1 = 0, v_0^2 = 1\}$	0.207
1	H	H	$d_1^1 = \{Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1\}$ $d_1^2 = \{Y_1^2 = L, v_0^1 = 0, v_0^2 = 1\}$	0.093
2	L	L	$d_2^1 = \{Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$	0.277
2	H	L	$d_2^1 = \{Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$	0.333
2	L	H	$d_2^1 = \{Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$	0.177
2	H	H	$d_2^1 = \{Y_2^1 = H, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 1, v_1^2 = 0\}$	0.213
2	L	L	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = H, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$	0.341
2	H	L	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = H, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$	0.220
2	L	H	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = H, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$	0.266
2	H	H	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = L, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = H, Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0, v_1^1 = 0, v_1^2 = 1\}$	0.173
2	L	L	$d_2^1 = \{Y_2^1 = L, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$	0.367
2	H	L	$d_2^1 = \{Y_2^1 = L, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$	0.240
2	L	H	$d_2^1 = \{Y_2^1 = L, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$	0.238
2	H	H	$d_2^1 = \{Y_2^1 = L, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$ $d_2^2 = \{Y_2^2 = ?, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 1, v_1^2 = 0\}$	0.155
2	L	L	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$	0.386
2	H	L	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$	0.246
2	L	H	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$	0.224
2	H	H	$d_2^1 = \{Y_2^1 = ?, Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$ $d_2^2 = \{Y_2^2 = L, Y_1^2 = L, v_0^1 = 0, v_0^2 = 1, v_1^1 = 0, v_1^2 = 1\}$	0.144

$$\bar{R}(X_1^1 = L, X_1^2 = L, d_1^1 = (Y_1^1 = L, v_0^1 = 1, v_0^2 = 0), d_1^2 = (Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0))$$

$$= \bar{R}(X_1^1 = L, X_1^2 = H, d_1^1 = (Y_1^1 = L, v_0^1 = 1, v_0^2 = 0), d_1^2 = (Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0))$$

$$= \bar{R}(X_1^1 = H, X_1^2 = L, d_1^1 = (Y_1^1 = L, v_0^1 = 1, v_0^2 = 0), d_1^2 = (Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0))$$

$$= \bar{R}(X_1^1 = H, X_1^2 = H, d_1^1 = (Y_1^1 = L, v_0^1 = 1, v_0^2 = 0), d_1^2 = (Y_1^2 = ?, v_0^1 = 1, v_0^2 = 0))$$

$$= \max\{7.5 \times 0.277 + 11.5 \times 0.333 + 14.5 \times 0.177 + 17.5 \times 0.213, 7.5 \times 0.341 + 11.5 \times 0.220 + 14.5 \times 0.266 + 17.5 \times 0.173\}$$

$$= \max\{12.201, 11.972\} = 11.972$$

$$\bar{R}(X_1^1 = L, X_1^2 = L, d_1^1 = (Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1), d_1^2 = (Y_1^2 = L, v_0^1 = 0, v_0^2 = 1)\})$$

$$= \bar{R}(X_1^1 = L, X_1^2 = H, d_1^1 = (Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1), d_1^2 = (Y_1^2 = L, v_0^1 = 0, v_0^2 = 1))$$

$$= \bar{R}(X_1^1 = H, X_1^2 = L, d_1^1 = (Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1), d_1^2 = (Y_1^2 = L, v_0^1 = 0, v_0^2 = 1))$$

$$= \bar{R}(X_1^1 = H, X_1^2 = H, d_1^1 = (Y_1^1 = ?, v_0^1 = 0, v_0^2 = 1), d_1^2 = (Y_1^2 = L, v_0^1 = 0, v_0^2 = 1)\})$$

$$= \max\{7.5 \times 0.367 + 11.5 \times 0.240 + 14.5 \times 0.238 + 17.5 \times 0.155, 7.5 \times 0.386 + 11.5 \times 0.246 + 14.5 \times 0.224 + 17.5 \times 0.144\}$$

$$= \max\{11.676, 11.492\} = 11.676$$

$$\bar{R}(X_0^1 = L, X_0^2 = L) = \max\{11.972, 11.676\} = 11.972.$$

Therefore, the salesperson should visit customer 1 in both period 0 and period 1.

4.2.2.2 Formulation II

The example in the previous section illustrates the expanding dimension of the state space if each state stores all history information of observations and actions. In this section, we provide a formulation in which rather than storing all history information in state D_t, we summarize the past information using a function of D_t, denoted as $B_t(D_t)$. In general, $B_t(D_t)$ is a probability distribution over D_t. We call $B_t(D_t)$ the *belief state* and treat the POMDP as a random process of evolving belief states (Chong et al. (2009)).

Let $B_t^i(d_t^i)$ be a function the information d_t^i for customer i in period t. We

define $B_t^i(d_t^i)$ as the posterior distribution of the underlying true adoption probability associated with customer i in period t given the information vector d_t^i, i.e.,

$$B_t^i(d_t^i) = \left[\Pr\{X_t^i = \alpha | d_t^i\}\right]_{\alpha \in \mathfrak{P}}, \tag{4.5}$$

where the distribution of $\Pr\{X_t^i = \alpha | d_t^i\}$ for $\alpha \in \mathfrak{P}$ is given in Equation 4.4. Assuming $\{X_t^i, t = 0, 1, \ldots, T\}$ and $\{X_t^j, t = 0, 1, \ldots, T\}$ are independent Markov chains for $i \neq j$, we have

$$B_t(D_t) = B_t(d_t^1, d_t^2, \ldots, d_t^N) = \prod_{i=1}^{N} B_t^i(d_t^i). \tag{4.6}$$

The support of distribution $B_t(D_t)$ is given by the Cartesian product of the set $\mathfrak{P}$, i.e., $\mathfrak{P}^N$. It is well established that the sequence of conditional probabilities, $\{B_t, t \in [0, T]\}$, is a Markov process (Bertsekas (2005), Monahan (1982)). Thus, $B_t(D_t)$ is a *sufficient statistic* of the information vector D_t. That is, the salesperson can not make better decision by having information D_t than having $B_t(D_t)$. The POMDP is then converted to a continuous-space "belief MDP", where the salesperson's beliefs about the underlying true states are encoded through a set of belief states.

Given a state $B_t(D_t)$ in period t, after taking an action a_t and making observations $[y_t^i]_{i \in N}$, we set $d_{t+1}^i = d_t^i \cup \{a_t, y_{t+1}^i\}$ for all $i \in N$ and update the belief state $B_{t+1}(D_{t+1})$ through Equations 4.5 and 4.6. For notational simplicity, we denote $B_t(D_t)$ by B_t in the following discussion. Let $R_t(D_t, a_t)$ be the reward earned in period t given the information vector D_t and action a_t. As no reward will be collected until the end of the horizon, we have $R_t(D_t, a_t) = 0$ for all $t < T$. A policy $\pi(B)$ of the POMDP maps the beliefs into actions. We note that in the POMDP, the policy $\pi(B)$

is a function over a continuous set of probability distributions as there are uncountably number of belief states. The policy $\pi(B)$ is characterized by the expected value function $V^{\pi}(B_0) = E_{\pi}\big[R_T^{\pi}(B_T, a_T)|B_0\big]$, where $R_T(B_T, a_T) = \sum_{D_t} R_T(D_T, a_T)B_T$ is the expected reward that can be collected at the end of the horizon. The optimal policy $\pi^*(B)$ is a policy that maximizes the value function $V^{\pi}(B_0)$. The Bellman equation for the POMDP is

$$V_t\big(B_t\big) = \max_{a_t \in A(B_t)} \big\{ \sum_{D_t} R_t(D_t, a_t)B_t + \sum_{D_{t+1}} \Pr\{D_{t+1}|D_t, a_t\} V_{t+1}\big(B_{t+1}\big) \big\}. \tag{4.7}$$

4.3 Solution Approach: A Heuristic

Due to the well-known curses of dimensionality (Powell, 2007), it is not computationally tractable to solve the MDP and POMDP optimally. In this section, we propose a heuristic approach to develop heuristic solutions. In the heuristic, for each period, we decompose the problem into an assignment problem and a routing problem to determine which customers to visit in each period and how to visit the selected customers within the period, respectively. We provide an overview of the heuristic in §4.3.1 and present the model for the assignment problem in §4.3.2.

4.3.1 Heuristic Approach

In developing the heuristic, we assume that the representative can observe a customer's adoption likelihood by meeting the customer. We assume that $\Pr\{X_{t+1}^i = \beta | X_t^i = \alpha, w_t^i\}$ is known for $i \in N$, and $\sum_{m=k}^{|\mathfrak{P}|} \Pr\{X_{t+1}^i = \alpha_m | X_t^i = \alpha, w_t^i = 0\} \leq \sum_{m=k}^{|\mathfrak{P}|} \Pr\{X_{t+1}^i = \alpha_m | X_t^i = \alpha, w_t^i = 1\}$, for all $i, t, k, \alpha \in \mathfrak{P}, \alpha_m \in \mathfrak{P}$.

The heuristic is presented in Algorithm 4.1. Following discussion in §4.2.2.2,

Algorithm 4.1 Heuristic Approach

1: **Input:** A set of customers N, B_0 and T
2: **Output:** A scheduling and routing policy $\Pi = (\pi_1, \ldots, \pi_T)$, B_T, and R_T
3: **for** $t = 1$ to T **do**
4: $T_a \leftarrow T - t + 1$
5: $\{\vec{v_1}, \ldots, \vec{v}_{T_a}\} \leftarrow Assign(N, B_t, T_a)$
6: $a_1 \leftarrow A\text{-}Priori\text{-}Routing(\vec{v_1})$
7: **for** $i \in N$ **do**
8: **if** $v_1^i = 1$ and $w_1^i = 1$ **then**
9: $B_t^i \leftarrow B_t^i \times \Psi_m^i$
10: **else**
11: $B_t^i \leftarrow B_t^i \times \Psi_u^i$
12: **end if**
13: **end for**
14: $\pi_t \leftarrow a_1$
15: **end for**
16: $R_T \leftarrow \sum_{i=1}^{N} r_i \sum_{k=1}^{|\mathfrak{P}|} \alpha_k \Pr\{X_T^i = \alpha_k\}$

we denote by $B_t = (B_t^1, \ldots, B_t^N)$ with $B_t^i \equiv [\Pr\{X_t^i = \alpha\}]$ the distribution of adoption likelihood at customers in period t. We denote by $\Pi = (\pi_1, \ldots, \pi_T)$ the scheduling and routing policy over horizon T, where π_t is a policy for selecting and visiting customers in period $t \in [1, T]$. We assume that the initial distribution of adoption likelihood $B_0 = [\Pr\{X_0^i = \alpha\}]$ is known for all $\alpha \in \mathfrak{P}$ and $i \in N$. We denote by R_T the expected reward from all customers at the end of horizon T, which is also the value of the heuristic policy Π.

In Line 5, we solve an assignment problem $Assign(N, B_t, T_a)$ for each $t \in [1, T]$, to determine the customers to schedule in period t, where T_a is the time horizon considered in the assignment problem and N is the set of customers available for assigning to T_a periods. As depicted in Line 4, we start with $T_a = T$ and update T_a in

each iteration of assignment and routing. We discuss the assignment problem in more detail in the following section. The solution to $Assign(N, B_t, T_a)$ is $\{\vec{v_1}, \ldots, \vec{v}_{T_a}\}$, where $\vec{v_k}$ ($k \in [1, T_a]$) represents the set of customers scheduled for period k. The customers to visit in the current period, is then given by set $\vec{v_1}$.

In Line 6, we develop an a priori route to visit customers $i \in \vec{v_1}$. We consider recourse actions of skipping customers, balking and reneging at customers in the a priori routing and implement the static rules proposed by Zhang et al. (2014) to execute the recourse actions. We implement the variable neighborhood search (VNS) heuristic presented in Zhang et al. (2014) to find an a priori route. In Line 7 through Line 13, we update the distribution of adoption likelihood. We define $\Psi^i_m = [\Pr\{X^i_{t+1} = \beta | X^i_t = \alpha, w^i_t = 1\}]$ and $\Psi^i_u = [\Pr\{X^i_{t+1} = \beta | X^i_t = \alpha, w^i_t = 0\}]$, corresponding to the one-step transition matrix given that the salesperson has and has not met customer i, respectively. If customer i is scheduled to visit and met by the salesperson in period t, we update the distribution of adoption likelihood for the next period by $B^i_t \times \Psi^i_m$. Otherwise, we update the distribution by $B^i_t \times \Psi^i_u$. In Line 14, we set the scheduling and routing policy in period t as a_1. We compute the value R_T of the heuristic policy Π in Line 16.

4.3.2 Assignment Problem

In this section, we present the integer programming model for the assignment problem. As discussed above, in each iteration of the heuristic, the input to the assignment model is the horizon T_a, the set of customers N considered for visit over horizon T_a,

and the distribution of adoption likelihood in period t, B_t. We note that T_a may be different from the problem horizon T and T_a is specified in Line 4 of Algorithm 4.1. For notational simplicity, in the following discussion, we use $B_o(= B_t)$ to denote the distribution of adoption likelihood that is given as input to each assignment problem, where $B_o = (B_o^1, \ldots, B_o^N)$.

Let $(u_1, u_2, \ldots, u_{T_a})$ be a vector representing the assignment to a customer, where $u_t(t \in [1, T_a])$ is a binary variable with $u_t = 1$ denotes that the customer is visited in period t and $u_t = 0$ otherwise. For example, considering a five-period horizon, $(0, 1, 1, 0, 1)$ represents that the customer is visited in periods 2, 3, and 5. We note that there are 2^{T_a} realizations of $(u_1, u_2, \ldots, u_{T_a})$. Let z_{ij} be a binary variable indicating the visiting plan $j \equiv (u_1, u_2, \ldots, u_{T_a}) \in \{0, 1\}^{T_a}$ for customer i. Specially, $z_{ij} = 1$ indicates that the visiting plan to customer i is j and $z_{ij} = 0$ otherwise. Let m_{ij} be the expected value associated with z_{ij}. An important assumption in the assignment model is that the salesperson will meet with each customers scheduled in her trip. Thus, we ignore the possibility that the salesperson may not meet with a customer who was scheduled and compute m_{ij} as $m_{ij} = r_i B_o^i (\Psi_m^i)^{u_1} (\Psi_u^i)^{u_1} \cdot \ldots \cdot (\Psi_m^i)^{u_{T_a}} (\Psi_u^i)^{u_{T_a}} \alpha$, where $\alpha = \{\alpha_k\}, k = 1, \ldots, |\mathfrak{P}|$ is the vector of adoption probabilities defined in §4.2.1. We assume that the salesperson can visit at most θ_t customers in period t. We also assume that the N customers are divided into K geographical regions, each denoted as $N_k(1 \leq k \leq K)$, such that during each period only customers from the same region can be visited. We define the binary variable y_{tk} to indicate whether the region N_k is assigned in period t. The integer programming model is as follows:

$$\max \sum_{i \in N} \sum_{j \in \{0,1\}^{T_a}} m_{ij} z_{ij}$$

$$\text{s.t.} \quad \sum_{j \in \{0,1\}^{T_a}} z_{ij} = 1, \quad \forall i \in N, \tag{4.8}$$

$$\sum_{i \in N} \sum_{j | u_t = 1 \in \{0,1\}^{T_a - 1}} z_{ij} \leq \theta_t, \quad \forall t \in [1, T_a], \tag{4.9}$$

$$z_{ij} \leq y_{kt}, \quad \forall k \in [1, K]; i \in N_k; t \in [1, T_a]; j | u_t = 1 \in \{0,1\}^{T_a - 1}, \tag{4.10}$$

$$y_{kt} \leq \sum_{j | u_t = 1 \in \{0,1\}^{T_a - 1}} z_{ij}, \quad \forall k \in [1, K]; i \in N_k; t \in [1, T_a], \tag{4.11}$$

$$\sum_{k=1}^{K} y_{kt} = 1, \quad \forall t \in [1, T_a], \tag{4.12}$$

$$z_{ij} \in \{0, 1\}, \quad \forall i \in N; t \in [1, T_a]; j \in \{0,1\}^{T_a}, \tag{4.13}$$

$$y_{kt} \in \{0, 1\}, \quad \forall t \in [1, T_a]; k \in [1, K]. \tag{4.14}$$

The objective of the model is to maximize the expected rewards collected from all customers at the end of the horizon T_a. Constraints (4.8) ensures that only one visiting plan is selected for each customer i. Constraints (4.9) bound the total number of customers visited in period t by θ_t. Constraints (4.10) and (4.11) ensure the consistency of customers selection and region assignment. Constraints (4.12) illustrate that in each period the salesperson can only visit customers within the same region. Constraints (4.13) and (4.14) specify the domains of the decision variables.

4.4 Computational Experiments

In this section, we discuss computational experiments for the problem. In §4.4.1, we present the problem instances generated from existing benchmark datasets

and provide the implementation details. In §4.4.2, we present benchmark solutions to compare with our heuristic solutions and provide computational results.

4.4.1 Problem Instances and Implementation

In the experiments, we consider the application of the multi-period orienteering to the case of routing a salesperson who visits customers on campus. We generate our datasets from Solomon's VRPTW instances (Solomon, 1987). As we assume that customers are located in several geographical regions in the problem, we modify Solomon's C sets with clustered customers. We maintain each customer's location and demand information as in Solomon's instances. We use the method presented in §3.5.1 to modify the time windows to either 60 minutes, 90 minutes or 120 minutes to reflect the durations of customers' time windows. We divide the 100 customers in each of Solomon's instances into three groups, each representing a geographical region and with 35, 30, and 35 customers, respectively. As mentioned in §4.3.2, we assume that the sales representative can only visit customers within the same region in each period.

We program Algorithm 4.1 in C++ and execute the experiments on 2.6-GHz Intel Xeon processor with 64-512 GB of RAM. We obtain an a priori route via the VNS presented in Zhang et al. (2014) and set the parameters based on the discussion in §2.7.1. For each problem instance, we report the average expected objective of 500 runs. We set $\mathfrak{P} = \{0, 0.25, 0.5, 0.75, 1\}$ and $\Psi_0^i = [0, 0, 1, 0, 0]$ for each $i \in N$. We set Ψ_m^i and Ψ_u^i as

Table 4.9: Heuristic Results for Instances with 100 customers

Dataset	Heuristic	CI_h	Greedy	CI_{lb}	Gap
C101	1089.20	[1088.06, 1090.34]	1081.57	[1080.58, 1082.55]	0.71%
C102	1045.33	[1043.88, 1046.78]	1044.34	[1042.85, 1045.83]	0.10%
C103	989.65	[988.06, 991.24]	990.44	[988.81, 992.06]	-0.08%
C104	932.92	[931.24, 934.60]	930.83	[929.07, 932.59]	0.22%
C105	1070.33	[1068.85, 1071.81]	1069.71	[1068.36, 1071.05]	0.06%
C106	1062.95	[1060.46, 1065.44]	1061.78	[1059.09, 1064.46]	0.11%
C107	1048.23	[1046.79, 1049.68]	1045.57	[1044.29, 1046.84]	0.25%
C108	1036.34	[1034.01, 1038.66]	1034.35	[1032.25, 1036.46]	0.19%
C109	1010.16	[1008.25, 1012.07]	1009.21	[1007.19, 1011.24]	0.09%

$$\Psi_m^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5 & 0.3 & 0.2 & 0 & 0 \\ 0.1 & 0.4 & 0.3 & 0.2 & 0 \\ 0 & 0.1 & 0.4 & 0.4 & 0.1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \Psi_u^i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0.3 & 0.4 \\ 0 & 0 & 0.1 & 0.5 & 0.4 \\ 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

for customer i, and let $\theta_t = 20$ for each $t \in [0, T]$.

4.4.2 Computational Results

To benchmark our heuristic approach, we consider a greedy approach to select the customers to visit in each period, where instead of solving an assignment problem as in Line 5 of Algorithm 4.1, we select customers based on the marginal values achieved by meeting with them. Specifically, for each customer i, we compute the marginal value, denoted by M_t^i, via $M_t^i = r_i B_t^i \Psi_m^i \alpha - r_i B_t^i \Psi_u^i \alpha$, and select the θ_t customers with the greatest marginal values. We note that the θ_t customers can only be selected from the same geographical region.

Table 4.9 presents the computational results. The second and the fourth columns report the average expected objectives over 500 sample paths from the heuristic approach and the greedy method, respectively. The third and fifth columns present the 95% confidence intervals for the two methods, respectively. The sixth column shows the gap between average expected objectives of the heuristic and the greedy methods ($Gap = \frac{\text{Heuristic}-\text{LB}}{LB} \times 100\%$).

Overall, the heuristic method performs similar to the greedy approach in most instances except instance C101. For C101, the heuristic solution is statistically better than the greedy solution based on a 95% confidence interval, and the gap between the heuristic and greedy solutions is 0.71%. In all the other instances, however, the heuristic approach is not statistically better than the greedy method based on a 95% confidence interval. In part, the lack of difference between results from the two approaches is due to the constraint that the representative can only visit customers within the same geographical regions in a period. This prevents the heuristic approach from taking advantage of considering the evolution of customers' adoption probabilities ahead when scheduling customers, which may results in greater expected reward at the end of the horizon.

4.5 Conclusion and Future Research

This work introduces a multi-period orienteering problem motivated by sales representatives visiting customers to observe and influence the likelihood that the customers will adopt the product. Each customer's adoption probability is uncertain

and may evolve over time. By meeting with the customer, however, the representative may not be able to fully observe the likelihood. Instead, she has to estimate the adoption probability and make decisions based on the estimations, which may not be accurate. We propose two models for the problem, one is a Markov decision process (MDP), in which we assume that in which we assume that the representative can obtain perfect information of the customers' adoption likelihood by meeting with the customers. The other is a partially observed Markov decision process (POMDP) which accounts for the inaccurate estimation of customers' adoption likelihood. Due to the fact that it is computational intractable to solve neither the MDP nor the POMDP via dynamic programming exactly, we propose a heuristic approach for the problem. In the heuristic, we first solve an assignment problem to select customers to visit for a period and then solve a routing problem to visit the selected customers within the period.

One future research avenue is to develop approximate dynamic programming (ADP) approach for the problem. Instead of computing exactly the reward associated with each state as in the exact dynamic programming, we approximate the reward-to-go at each decision epoch via heuristic methods in the ADP. Our future research will be focused on investigating heuristic methods for reward-to-go approximation.

Another research avenue is to transform the dynamic programming models into equivalent integer programs, where the latter can be solved by readily available solvers. However, directly converting the current MDP and POMDP models may results in integer programs that is not practically solvable via commonly available

solution engines. Thus, we will investigate additional assumptions and restrictions that can be introduced to the original dynamic programming models to make the transformation validated.

CHAPTER 5
CONCLUSIONS

In this thesis, we introduce a new variety of orienteering problems motivated by routing sales representatives, but generalizable to other setting in which a routed entity may experience queues at a series of locations and the reward collectable from each location may evolve over time. We present models that explicitly accounts for the stochasticity induced by uncertain wait times at customers and uncertain adoption likelihood associated with customers. We focus on developing solution approaches that provide good quality solutions within reasonable computational efforts.

In §2, we propose customer-specific recourse actions corresponding to the queueing notions of balking and reneging, which is a novel addition to routing models. We establish decision rules to execute the recourse actions and derive an analytical formula to compute the expected sales from an a priori tour with recourse. Via computational results, we demonstrate the value of incorporating stochastic information into the routing model.

Considering a priority queue at each customer, in §3 we propose the dynamic orienteering problem on a network of queues. We model the problem as a MDP and focus on developing dynamic policies that enable the salesperson to decide which customer to visit and whether to stay in queue at a customer based on realized information at each decision epoch. We investigate the existence of optimal control limits and identify the limited existence of optimal control limit policies. To facilitate decision making, we propose a novel compound rollout heuristic, which differs

from existing approximate dynamic programming approaches in implementing different mechanisms for reward-to-go approximation based on explicitly partitioning the action space. We demonstrate the capability of our compound rollout in making high quality decisions with reasonable computation efforts by the computational experiments that comparing our dynamic policies to benchmark a priori routing solutions.

As an extension from the daily orienteering, in §4, we introduce a multi-period orienteering problem in which a traveler visits customers over a multi-period horizon to influence and observe the chance of customer adoption. We consider that each customer's likelihood of adopting the product is uncertain and may evolve over time. The representative is not able to fully observe a customer's adoption likelihood by meeting with the customer. We assume that the representative can only estimate each customer's adoption likelihood and the accuracy of estimation is uncertain. We model the problem as a partially observed Markov decision process (POMDP) with an objective to maximize the expected sales at the end of the horizon. Our model explicitly accounts for partially observable nature of customers' adoption likelihood. We propose a heuristic approach to facilitate decision making, which iteratively solves an assignment problem to determine which customers to visit in a period and a routing problem to visit the selected customers within the period.

REFERENCES

Arnold, M. 2012. Big cuts for AstraZeneca sales force. Retrieved August 18, 2012, http://www.mmm-online.com/big-cuts-for-astrazeneca-sales-force/article/220025/.

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. 2008a. Challenges and advances in a priori routing. B. Golden, S. Raghavan, E. Wasil, eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*, *Operations Research/Computer Science Interfaces Series*, vol. 43. Springer, New York, 123–142.

Campbell, A. M., B. W. Thomas. 2008b. 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.

Chao, I., B. L. Golden, E. A. Wasil. 1996. The team orienteering problem. *European Journal of Operational Research* **88** 464–474.

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.

Chong, E. K. P., C. M. Kreucher, A. O. Hero III. 2009. Partially observable markov decision process approximations for adaptive sensing. *Discrete Event Dynamic Systems* **19** 377–422.

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.

Falk, N.J., J.E. Gariepy, T. A. Wrobleski. 2012. 8 steps to a "servitized" supply chain. http://www.supplychainquarterly.com/topics/Strategy/20121001-8-steps-to-a-servitized-supply-chain/.

Feillet, D., P. Dejax, M. Gendreau. 2005. Traveling salesman problems with profits. *Transportation Science* **21**(2) 241–257.

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. 2014. 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. 2015. Restocking-based rollout policies for the vehicle routing problem with stochastic demand and duration limits. To appear in *Transportation Science*.

Hansen, P., N. Mladenovic. 2003. Variable neighborhood search. *Handbook of Metaheuristics*, *International Series in Operations Research & Management Science*, vol. 57. Springer, New York, 145–184.

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

Hu, Q., A. Lim. 2014. An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research* **232** 276–286.

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.

Labadie, N., R. Mansini, J. Melechovsky, R. W. Calvo. 2012. The team orienteering problem with time windows: An lp-based granular variable neighborhood search. *European Journal of Operational Research* **220** 15–27.

Labadie, N., J. Melechovsky, R. W. Calvo. 2011. Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. *Journal of Heuristics* **17** 729–753.

Lovejoy, W. S. 1991. A survey of algorithmic methods for partially observed markov decision processes. *Annals of Operations Research* **28** 47–66.

Monahan, G. E. 1982. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science* **28**(1) 1–16.

Montemanni, R., L. M. Gambardella. 2009. Ant colony system for team orienteering problem with time windows. *Foundations of Computing and Decision Sciences* **34**(4) 287–306.

Neely, A. 2014. The servitization of manufacturing: An analysis of global trends. Retrieved April 12, 2014, http://www.cambridgeservicealliance.org/uploads/downloadfiles/BerlinServicesKeynote.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.

Puterman, M. L. 2005. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. NY: Wiley, New York.

Rockoff, J. D. 2012. Drug reps soften their sales pitches. *The Wall Street Journal.* http://online.wsj.com/article/SB10001424052970204331304577142763014776148.html.

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

Schrimpf, G., J. Schneider, H. Stamm-Wilbrandt, G. Dueck. 2000. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* **159** 139–171.

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.

Souffriau, W., P. Vansteenwegen, G. V. Berghe, D. V. Oudheusden. 2013. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science* **47**(1) 53–63.

Spaan, M. T. J. 2012. Partially observable markov decision processes. *Reinforcement Learning*, *Adaptation, Learning, and Optimization*, vol. 12. Springer Berlin Heidelberg, 387–414.

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.

Tricoire, F., M. Romauch, K. F. Doerner, R. F. Hartl. 2010. Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research* **37** 351–367.

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

Vansteenwegen, P., W. Souffriau, D. V. Oudheusden. 2011. The orienteering problem: A survey. *European Journal of Operational Research* **209** 1–10.

Verbeeck, C., P. Vansteenwegen, E. H. Aghezzaf, M. Vanlommel, B. W. Thomas. 2015. Solving the stochastic time-dependent orienteering problem with time windows. Working paper.

Voccia, S., A. M. Campbell, B. W. Thomas. 2013. Probabilistic traveling salesman problem with time windows. *EURO Journal on Transportation and Logistics* **2** 89–107.

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.