The Dissertation Committee for Tarun Rambha
certifies that this is the approved version of the following dissertation:

Dynamic Congestion Pricing in Within-Day and Day-to-Day Network Equilibrium Models

Committee:

Stephen D. Boyles, Supervisor

Chandra Bhat

Christian Claudel

John Hasenbein

Peter Stone

Dynamic Congestion Pricing in Within-Day and Day-to-Day Network Equilibrium Models

by

Tarun Rambha, B.Tech, M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2016

To my teachers.

Acknowledgments

Several individuals who were a part of my six long years at UT deserve a special mention. First, I would like to express my gratitude to Steve Boyles for patiently helping me navigate my graduate studies with his constant guidance and encouragement. His efforts in creating a secure environment, in which I could explore without the fear of failing, are greatly appreciated and I consider myself very lucky to have worked with him. I would also like to thank members of my committee—Chandra Bhat, Christian Claudel, John Hasenbein, and Peter Stone—for their comments on my research, which significantly improved the content and structure of this dissertation. Thanks again to Steve Boyles and John Hasenbein, and to Michael Starbird for teaching courses in impressionable ways. I am also grateful for the financial support from NSF, TxDOT, D-STOP and SWUTC.

I thank my former advisors Travis Waller and Karthik Srinivasan for motivating me to work in the area of traffic networks. I also thank my collaborators, especially Avi for sharing his advice on academia and beyond; Ehsan for being a wonderful office-mate and for the countless intellectual conversations; and Kai for having an enthusiasm that was contagious and for pointing me to several interesting papers in obscure areas. I would also like to thank current and previous members of my group—Michael, Alireza, Shoupeng, Venktesh, Rachel, John, Ravi, Chris, Sudesh, Rohan, Ruoyu, Mark, and the NMC staff—for offering some great company during my stay at UT and at conferences. Special thanks to Steve and Hyojin for hosting several fun-filled end-of-semester potlucks. I would also like to acknowledge former students of CE 311S and the GLUE program for offering memorable teaching and mentoring experiences.

The last few years have also introduced me to many special friends. Thanks to my unpaid shrinks—Vivek and GT—for being just a call away; my partners in crime—Prem and Dinesh—for assisting in exploring and exploiting Austin; and my fellow graduates—Prateek, Subodh, Ankita, Rajesh, Raghu, Prasad, and Sundeep—for sharing many unforgettable moments.

I am also grateful to have found some amazing friends to play and compose music with. Thanks to Adith, Aaron, and Ashwin for the weekly jamming sessions which provided a much needed respite from my academic and worldly pursuits. I also thank Srikanth, Sastry, Anand, and Shruti for previously collaborating on a few live gigs.

I would not have come this far if not for the blessings and sacrifices of my parents, sister, and grandmother. Their presence has been a great source of strength. I am also particularly thankful to my mom for making sure that I was well fed and watered, especially when I fractured my thumb in a cricket match and was forced to type most of this dissertation with one hand. Finally, I am indebted to all the teachers in my life who have and continue to inspire me. This dissertation is dedicated to you all.

Dynamic Congestion Pricing in Within-Day and Day-to-Day Network Equilibrium Models

Tarun Rambha, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Stephen D. Boyles

This dissertation explores two kinds of dynamic pricing models which react to within-day and day-to-day variation in traffic. Traffic patterns vary within each day due to uncertainty in the supply-side that is caused by non-recurring sources of congestion such as incidents, poor weather, and temporary bottlenecks. On the other hand, significant day-to-day variations in traffic patterns also arise from stochastic route choices of travelers who are not fully rational. Using slightly different assumptions, we analyze the network performance in these two scenarios and demonstrate the advantages of dynamic pricing over static tolls. In both cases, traffic networks are characterized by a set of stochastic states. We seek optimal tolls that are a function of the network states which evolve within each day or across days.

In the within-day equilibrium models, travelers are assumed to be completely rational and have knowledge of stochastic link-states, which have different delay functions. At every node, travelers observe the link-states of downstream

links and select the next node to minimize their expected travel times. Collectively, such behavior leads to an equilibrium, which is also referred to as user equilibrium with recourse, in which all used routing policies have equal and minimal expected travel time. In this dissertation, we improve the system performance of the equilibrium flows using state-dependent marginal link tolls. These tolls address externalities associated with non-recurring congestion just as static marginal tolls in regular traffic assignment reflect externalities related to recurring congestion.

The set of tolls that improve system performance are not necessarily unique. Hence, in order to make the concept of tolling more acceptable to the public, we explore alternate pricing mechanisms that optimize social welfare and also collect the least amount of revenue in expectation. This minimum revenue toll model is formulated as a linear program whose inputs are derived from the solution to a novel reformulation of the user equilibrium with recourse problem.

We also study day-to-day dynamic models which unlike traditional equilibrium approaches capture the fluctuations or stochasticity in traffic due to route choice uncertainty. Travelers decisions are modeled using route choice dynamics, such as the logit choice protocol, that depend on historic network conditions. The evolution of the system is modeled as a stochastic process and its steady state is used to characterize the network performance. The objective of pricing in this context is to set dynamic tolls that depend on the state of the network on previous day(s) such that the expected total system travel time is minimized. This problem is formulated as an average cost Markov decision process. Approximation methods are suggested to improve computational tractability.

The day-to-day pricing models are extended to instances in which closed form dynamics are unavailable or unfit to represent travelers' choices. In such cases, we apply Q-learning in which the route choices may be simulated off-line or can be observed through experimentation in an online setting. The off-line methods were found to be promising and can be used in conjunction with complex discrete choice models that predict travel behavior with greater accuracy.

Overall, the findings in this dissertation highlight the pitfalls of using static tolls in the presence of different types of stochasticity and make a strong case for employing dynamic state-dependent tolls to improve system efficiency.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Urban transportation planning is traditionally carried out using a four-step method. The first three steps are used to estimate the number of travelers/users, their origin-destination (OD) pairs, and their mode of travel. The final step, also called *route choice* or *traffic assignment*, involves assigning travelers to different routes. This assignment procedure is done assuming that traffic networks are in a state of *user equilibrium* (UE) or *Nash equilibrium* (NE), which states that "*All used routes between an OD pair have equal and minimal travel times.*" The UE principle results from assuming that users selfishly choose routes so as to minimize their travel time. Many efficient algorithms exist for finding the UE solution to the traffic assignment problem (TAP) (Larsson and Patriksson, 1992; Jayakrishnan et al., 1994; Bar-Gera, 2002; Dial, 2006; Bar-Gera, 2010; Mitradjieva and Lindberg, 2013). Another state typically of interest is called the *system optimum* (SO), in which the sum total of travel time experienced by all travelers, also called the *total system travel time* (TSTT), is minimized. The equilibrium solution to the traffic assignment problem (TAP) can be expressed either in terms of link flows (volume of users on each roadway link) or path flows (volume of users on each path between every OD pair).

1.1 Traffic Assignment Problem

Consider a directed network $G = (N, A)$, where N and A are the set of nodes and arcs/links respectively. Assuming that the flow on an arc $(i, j) \in A$ is denoted by x_{ij}, let the function $t_{ij}(x_{ij})$ (also referred to as *link performance function* or *latency/delay function*) represent the travel time experienced by users on arc (i, j). Suppose the set of origins and destinations is denoted by $Z \subseteq N$ and the demand between origin and destination nodes $u \in Z$ and $v \in Z$ is represented by d_{uv}. We denote the set of paths between u and v using Π_{uv} (we consider only the set of simple paths, i.e., ones without any directed cycles) and the set of all paths in the network using $\Pi = \cup_{(u,v)\in Z^2} \Pi_{uv}$. We will use the notation $(i, j) \in \pi$ to denote that link (i, j) belongs to path π. Assume that y_π denotes the flow on a path π. Let δ_{ij}^π represent the arc-path incidence variable, i.e., δ_{ij}^π is 1 if $(i, j) \in \pi$ and is 0 otherwise. The following mathematical program, proposed by Beckmann et al. (1956) (and hence popularly known as the Beckmann formulation), describes the traffic assignment problem:

$$\min \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(x)\, dx \tag{1.1}$$

$$\text{s.t.} \quad x_{ij} = \sum_{\pi\in\Pi} \delta_{ij}^\pi y_\pi \qquad \forall\, (i, j) \in A \tag{1.2}$$

$$\sum_{\pi\in\Pi_{uv}} y_\pi = d_{uv} \qquad \forall\, (u, v) \in Z^2 \tag{1.3}$$

$$y_\pi \geq 0 \qquad \forall\, \pi \in \Pi \tag{1.4}$$

$$x_{ij} \geq 0 \qquad \forall\, (i, j) \in A \tag{1.5}$$

If the link performance functions are continuous, it is easy to verify that the objective of the TAP is continuous and differentiable. Hence, existence of an optimal solution follows directly from the fact that the objective is continuous and the constraints define a compact feasible region. If it is also assumed that the link performance functions are non-decreasing, the objective is convex and hence every equilibrium solution has equal link travel times. Furthermore, if the link performance functions are strictly increasing then there exists a unique solution in link flows to the TAP (Patriksson, 2015).

At low volumes, the travel time on a roadway link is usually insensitive to increase in flow but as more travelers use it, variability in driver behavior and speeds results in an increase in time taken to traverse the link. Hence, one expects that the link performance functions be non-decreasing. A widely used class of link performance functions are functions of the type $t_{ij}(x_{ij}) = \tau_{ij}\left(1 + \alpha(x_{ij}/\mu_{ij})^{\beta}\right)$ (also known as the Bureau of Public Roads (BPR) function), where μ_{ij} and τ_{ij} denote the capacity of link (i, j) and its free-flow travel time respectively, and α and β are parameters. Since these functions are strictly increasing an equilibrium solution obtained using BPR functions in unique in link flows. The mathematical elegance of the TAP has led to its widespread use in the transportation planning process.

Remark. The assumption that the travel time on an arc depends only on the flow on it is also known as the *separability condition*. Relaxing this assumption leads to a more general traffic assignment formulation which can help model the impacts of intersections. These problems are usually expressed as a variational inequality (VI) (see **?**Dafermos, 1980). Another widely studied variant of the TAP is called dynamic traffic assignment (DTA), which captures traffic queue dynamics (see Peeta and Ziliaskopoulos, 2001; Chiu et al., 2010). Unlike DTA models which represent traffic at a meso- or micro-scopic level, we restrict our attention to computing the equilibrium flows and optimal tolls in the presence of link delay functions.

1.2 Pricing in Transportation Networks

An equilibrium assignment is not optimal from a network wide perspective, i.e., it does not minimize the TSTT. In fact, for certain classes of link performance functions, one can bound the inefficiency of the UE solution (Roughgarden, 2002). Economists have suggested using tolls to drive a network of selfish users to an SO state. The idea of using tolls to control congestion dates to Pigou (1920), who proposed the concept of marginal tolls which are equal to the congestion externalities imposed by a traveler. To be more precise, suppose each arc in the network can be tolled and the toll paid by a traveler is the sum of tolls on the arcs along his/her path. Also suppose that the toll on arc (i, j) is given by $x_{ij}t'_{ij}(x_{ij})$, where x_{ij} is the flow on arc (i, j). If users act in a selfish manner but try to minimize the sum of travel time and toll, then the solution to the UE problem in the presence of tolls has an objective $\sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(x) + xt'_{ij}(x)\, dx = \sum_{(i,j)\in A} \int_0^{x_{ij}} d(xt(x)) = \sum_{(i,j)\in A} x_{ij}t(x_{ij})$. Thus congestion pricing with marginal tolls helps reduce the TSTT as the UE flow on the network with tolls results in a SO flow pattern in the original network.

Efficient algorithms and extensions to the TAP fueled modifications to marginal pricing to account for cases in which tolls may be collected only on a subset of links belonging to a cordon or a freeway (Verhoef et al., 1996); travelers have different values of time (Dial, 1999a); or the demand is elastic (Yildirim and Hearn, 2005). Models that maximize profit (Labbé et al., 1998), which could be of use to a private tolling firm, also exist. A thorough review on congestion pricing in theory and practice can be found in de Palma and Lindsey (2011). Despite these research efforts that span several decades, congestion pricing found its way into practice (in a limited way) with great difficulty for political and financial reasons. It was and continues to remain unpopular among the public as many consider it inequitable (Ecola and Light, 2009). Also, the set-up costs for toll collection requires considerable investment in infrastructure although the revenue from tolls can be used to maintain and operate such facilities.

However, with the advent of electronic roadway pricing systems some of these hurdles have been overcome and tolls have been instrumental in addressing congestion issues on freeways all around the world and on arterials in cities such as London, Hong Kong, and Singapore. The future is also likely to see innovative means of pricing in the form of vehicle miles traveled (VMT) taxes, credit based systems (Yang and Wang, 2011), and autonomous vehicles, all of which could make pricing a very practical and effective solution to deal with congestion.

1.3 Motivation and Research Objectives

One of the key issues to consider when setting tolls is the dynamic nature of traffic. On no two days are the flow patterns in a network the same. Furthermore, changes in network conditions due to non-recurring sources of congestion can result in significant variation of traffic within each day. Thus, in order to effectively manage congestion we could let the tolls be dynamic. The question that remains is how to set the right tolls to reduce congestion. In order address this question, it is first necessary to understand the sources of uncertainty that cause variability in traffic networks.

Uncertainty in network equilibrium models can be of three types: supply-side uncertainty, demand-side uncertainty, and route choice uncertainty and all three forms of uncertainty lead to traffic states that vary over time. Supply-side uncertainty mostly stems from non-recurring sources such as incidents and poor weather while demand-side uncertainty originates from changes in travelers' decisions to travel, mode and departure time choices. Both these forms of uncertainty can lead to uncertainty in travelers' route choices. However, by route choice uncertainty we refer to the inherent randomness associated with the lack of perfect rationality among travelers or that due to unobserved factors that impact route choice.

Modeling any of these three sources of uncertainty requires several assumptions on the distributions of uncertainty and the timing of observations. Thus, studying each type of uncertainty is a challenge in itself which makes a unified theory for modeling stochasticity in traffic networks and dynamic pricing sound far-fetched. We therefore restrict our attention to two different pricing models in this dissertation which improve network performance under two different sources of stochasticity: supply-side and route choice uncertainty. Hence, throughout this dissertation, the number of travelers in the network is assumed fixed and known. A study of these two problems will likely provide insights into developing hybrid models that capture more features of the route choice process.

More specifically, the two models proposed in this dissertation solve the problem of dynamic pricing in the following contexts:

- *Within-day pricing models:* When supply-side parameters such as capacity and free-flow travel time vary due to factors such as incidents, poor weather, and bottlenecks, we may assume that network arcs exist in a finite number of states with different delay functions with different probabilities. In such scenarios, travelers do not just choose paths but follow routing policies that respond to *en route* information. It is assumed that at every node in the network, travelers observe the true state of the immediate downstream arcs and select one of them to minimize their expected cost of travel. In this context, we find dynamic tolls that vary with the state of each link in order to minimize the total expected travel time of all the users. While humans would find it difficult to optimize their routes in the presence of such dynamic tolls, autonomous vehicles equipped with computers can find the best possible strategy that minimizes the expected generalized cost of travel. A marginal cost pricing scheme that leads to a socially optimal outcome and methods to compute it are discussed in detail. Further, alternate equilibrium formulations are explored and their solutions are used to find tolls that lead to a socially optimal outcome while generating the least amount of revenue in expectation.

- *Day-to-day pricing models:* A finite number of habitual drivers choose from a set of routes over different days. From an empirical standpoint, when a large number of selfish humans travel in a network, the chances of reaching an equilibrium are slim. User behavior in such settings can be modeled using probabilistic route choice models which define when and how travelers switch paths. This approach results in stochastic processes with steady state distributions containing multiple states in their support. Tolls are set on each day based on the route pattern(s) on previous day(s) and revealed to users before they make their trips. Travelers are also known to have access to historic travel times and choose routes in a probabilistic way as they do not precisely know how other travelers react to the travel time and toll information. The route choice probabilities of travelers are either assumed to be known to the system manager or may be inferred from real world data. The objective is to minimize the expected TSTT over an infinite horizon. In order to make this framework practical, approximation schemes for handling a large number of users are developed.

1.4 Examples

In this section, we demonstrate the core results of this dissertation using simple two link networks. The first example demonstrates how dynamic tolls can improve network performance in a within-day setting when the supply-side is uncertain and the second example illustrates a similar result in a day-to-day setting when route choices are uncertain.

1.4.1 Within-Day Pricing

To illustrate within-day pricing under supply-side uncertainty, consider the example in Figure 1.1. Suppose that the 1 unit of demand between nodes 1 and 2 is infinitely divisible. Since we model a nonatomic version of the problem, the terms 'travelers' and 'users' are to be interpreted as flow rates. The top link has a constant travel time of 1 unit. The bottom link on the other hand is congestible and exists in two states with link performance/delay functions x^2 and $2x$ with probabilities 0.6 and 0.4 respectively. These states on bottom link are referred to as s_1 and s_2. Of the 1 unit of demand arriving at node 1, 0.6 and 0.4 units of flow see the bottom arc in states s_1 and s_2 respectively. (In general, if η travelers arrived at node 1, 0.6η and 0.4η travelers observe

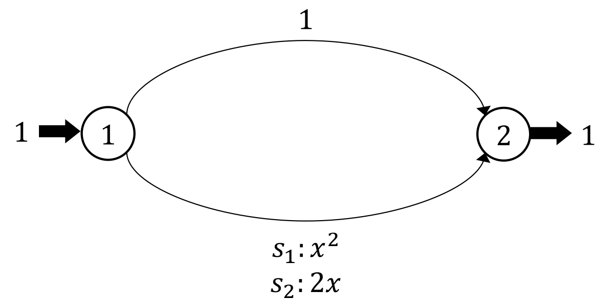


Figure 1.1: Demonstration of system optimal solutions with recourse.

the bottom arc in states s_1 and s_2 respectively.) A policy for a traveler is a complete contingent plan of action that selects a downstream node at each node, for each of the possible set of adjacent link-states (and tolls) at that node. For instance, a policy in the network in Figure 1.1 may require a traveler to head to node 2 via the top arc if the state s_1 is observed at node 1 and use the bottom arc otherwise. Thus, each traveler has 4 policies to choose from (see Table 1.1) and a feasible assignment involves dividing the 1 unit of demand across these policies. Let $y_1, \ldots, y_4$ represent the number of travelers using the 4 policies. The cost of a policy is a random variable and hence we suppose that travelers choose policies which minimize the expected travel time.

The system optimal solution may assign a positive demand to all four policies, whereas at equilibrium, all travelers select the bottom arc. We will refer to the system optimum and equilibrium states as system optimum with recourse (SOR) and user equilibrium with recourse (UER) respectively. Unless stated otherwise, we assume that all travelers have the same value of time (VOT)

Table 1.1: Flows on policies at UER and SOR states (T and B represent top and bottom arcs respectively).

Policy No.	*Arc* s_1	s_2	y^{SO}	y^{UE}
1	T	T	0.0185	0
2	B	B	0.6058	1
3	T	B	0.0192	0
4	B	T	0.3565	0

and units for the tolls are chosen such that VOT equals 1. Let the total expected travel time (TETT) represent the sum total of the expected travel times of all the users in the network. At the SOR state, the number of users on the bottom arc in states s_1 and s_2 are $0.6(0.6058 + 0.3565) = 0.5774$ and $0.4(0.6058+0.0192) = 0.25$ respectively. The number of users on the top arc is $0.6(0.0185 + 0.0192) + 0.4(0.0185 + 0.3565) = 0.1726$. Thus, the TETT of the SOR solution is $0.1726 + (0.5774)^3 + 2(0.25)^2 = 0.4901$. On the other hand, the TETT of the UER solution is $0.6^3 + 2(0.4)^2 = 0.536$. Our findings in this dissertation suggest that by collecting a marginal toll of $2(0.5774)^2 = 0.6667$ when the bottom link is in states s_1 and $2(0.25) = 0.5$ when it is in state s_2 would result in a socially optimal solution.

1.4.2 Day-to-Day Pricing

As explained in Section 1.2, congestion pricing helps reduce the TSTT as the UE flow on the network with marginal tolls results in a SO flow pattern in the original network. However, in a day-to-day setting, marginal prices are of little relevance. In fact, in some cases, they can result in increased TSTT as illustrated by the following example.

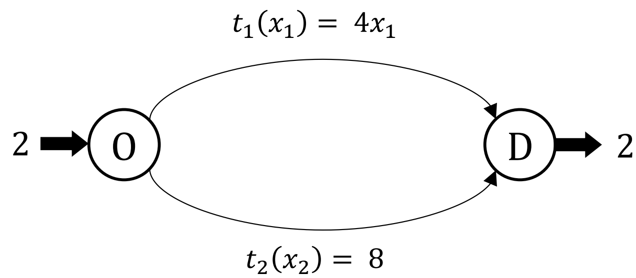


Figure 1.2: Sub-optimality of marginal prices in a day-to-day setting.

Consider two travelers from O to D in the network shown in Figure 1.2. Demand in a day-to-day traffic model represents actual travelers and not flow rates and is integral in nature. Let the vector (x_1, x_2) denote the state of the system, where x_1 and x_2 denotes the number of travelers on the top and the bottom path respectively. The above system has three states $(2, 0)$, $(0, 2)$, and $(1, 1)$, which we will refer to as states 1, 2 and 3 respectively. It is easy to verify that state 1 is a NE and state 3 is SO. Suppose both travelers use the logit choice model to select paths on each day, in which the probability of choosing the top and bottom paths are $\frac{\exp(-t_1(x_1))}{\exp(-t_1(x_1))+\exp(-t_2(x_2))}$ and $\frac{\exp(-t_2(x_2))}{\exp(-t_2(x_2))+\exp(-t_2(x_2))}$, where $t_1(x_1)$ and $t_2(x_2)$ represents the travel times as a function of the previous day's flow. The stochastic process is Markovian and the steady state probabilities of observing states 1, 2, and 3 are 0.5654, 0.1414, and 0.2932 respectively. Thus, the expected TSTT is $16(0.5654) + 16(0.1414) + 12(0.2932) = 14.8272$. Now suppose we price the network using marginal tolls (4 units on the top link and no toll on the bottom one) and assume that both travelers now replace the travel time functions in the logit choice model with generalized costs (travel time + toll). The steady state distribution of the Markov chain for states 1, 2, and 3 is 0.467, 0.467, and 0.066 respectively and the expected TSTT is 15.736 which is higher than before.

On the other hand, suppose tolls were dynamic and a function of the system state. Specifically, let the toll on the top link be 0, 8, and 4, in states 1, 2, and 3 respectively. Then the steady state probabilities of finding the system in states 1, 2, and 3 are 0.25, 0.25, and 0.5 respectively and the expected TSTT is 14, which is less than the expected TSTT of the no-tolls scenario.

Remark. The tolling solutions discussed in this dissertation are state dependent. As the state of the network evolves within each day or across days, the tolls change and are hence dynamic in nature. Such tolling mechanisms are also referred to as adaptive or responsive congestion pricing (Boyles et al., 2010). Another type of dynamic pricing involves collecting different tolls at different times of the day. These time-of-day tolls are designed to ease recurring congestion, but they do not adapt to stochastic variations in traffic.

1.5 Contributions

This dissertation explores two kinds of dynamic pricing models while capturing within-day variation in supply-side and day-to-day variation in route choices. The within-day pricing framework studies the impact of non-recurring events such as incidents, weather, and bottlenecks that result in supply-side uncertainty. In such scenarios, travelers are incentivized to adaptively select links instead of following a fixed route. Assuming that links exist in multiple states with different delay functions, a state-dependent pricing mechanism is proposed that can bring the equilibrium and system optimum solutions into alignment. The state-dependent link tolls address externalities associated with non-recurring congestion just as static marginal tolls in regular traffic assignment reflect externalities related to recurring congestion. The major contributions of the within-day pricing model are:

1. Despite the prevalence of non-recurring congestion, tolling models under supply-side uncertainty have not received enough attention in pricing literature. Furthermore, existing research does not account for the effect of rerouting among travelers. In this dissertation, we formulate a system optimum model with recourse in which travelers do not simply select paths but follow adaptive routing policies that respond to *en route* information and minimize expected generalized costs.

2. Users arriving at a node observe the states of the downstream links and the associated tolls (that are different for different link-states) and select the next link to travel on. The optimal polices are computed assuming full-reset, i.e., the probability of observing a downstream link-state resets every time the traveler revisits its head node. However, the reset assumption is known to induce cycles in the optimal policies. We

address this modeling artifact by proposing a network transformation and a reformulation using symmetric delay functions. The Frank-Wolfe algorithm is adopted to find the equilibrium solution and the optimal tolls and the sub-optimality of static tolls is demonstrated on the Sioux Falls test network.

3. We formulate an alternate equilibrium with recourse model with split proportions as the variables. A solution method that is similar to the origin-based assignment is proposed. The results of this formulation are then used to construct a linear program which computes state-dependent optimal tolls that generate the least amount of expected revenue.

4. The within-day pricing mechanism would be useful especially in a network with connected, autonomous vehicles because the vehicles will have the compute power to calculate optimal policies and because the network manager can collect and vary link tolls without heavily investing in pricing infrastructure.

We also study day-to-day dynamic models, which unlike traditional equilibrium approaches can capture the fluctuations or stochasticity in traffic due to route choice uncertainty. The objective in these problems is to set dynamic tolls that minimize the expected total system travel time. Exact and approximate methods to solve the problem are discussed and their applications are demonstrated. The major contributions of the day-to-day pricing model are:

1. Existing dynamic pricing methods in day-to-day traffic literature focus on continuous time deterministic models. However, the solutions from these methods cannot be used to set tolls on different days in the network. To our knowledge, this research is the first to study dynamic tolling in a discrete time setting using an average cost Markov decision process (MDP) framework.

2. The problem is formulated as an infinite horizon average cost MDP and optimal stationary policies are computed that enable a system manager to decide the tolls based on the state of the system/flows on any given day. Several alternate pricing models with different objectives that may be of interest to a system manager were also formulated.

3. Most MDPs inherit the curse of dimensionality that makes it difficult to solve them using exact methods. Hence, we propose simple state space aggregation methods that were found to be computationally tractable. Using a test network, we demonstrate that the approximate optimal policies from these methods result in lower expected total system travel times when compared to the no-tolls case and also analyze the mixing times of associated Markov chains.

4. However, an exact computation of the optimal strategy requires closed form transition probabilities. We relax this assumption by exploring reinforcement learning methods in which state transitions are simulated or observed from the field. Small experiments on the Braess network are

performed to demonstrate the application of the proposed formulation.

1.6 Organization and Notes

This dissertation is organized as follows: In Chapters 2 and 3 we study the within-day pricing problem under supply-side uncertainty and in Chapters 4 and 5, we study the day-to-day pricing problem under route choice uncertainty. Chapter 2 explores the system optimum with recourse problem and provides a link-based method for computing the equilibrium flows and optimal state-dependent tolls. These models are extended in Chapter 3 by means of an alternate formulation in terms of the split proportions. The solution to this method is used to formulate a minimum revenue state-dependent pricing problem as a linear program. Chapter 4 introduces day-to-day dynamic models and formulates exact and approximate methods to find tolls as a function of the state of the system assuming that travelers follow a logit choice model. Chapter 5 attempts to find the optimal pricing policy by assuming that closed form expressions for route choice are unavailable and discusses models involving simulated route choices and online observations. Finally, in Chapter 6, the contributions of this dissertation are summarized and directions for future work are outlined.

Chapters 2 and 3 can be read independently of Chapters 4 and 5. Different notation has been used in these parts of the dissertation to avoid the use of uncommon symbols. Some of the work in this dissertation is adapted from

Rambha and Boyles (2016) and some of it is currently under review (Rambha et al., 2016).

For most part, this dissertation is self-contained. However, the reader might benefit from basic knowledge of traffic equilibrium, stochastic processes, and Markov decision processes. We recommend books by Sheffi (1985) and Patriksson (2015) for background on traffic assignment, a text by Kulkarni (2009) for an introduction to stochastic processes, and books by Puterman (2005) and Bertsekas (2007) for a comprehensive study of Markov decision processes.

Chapter 2

Within-Day Pricing: System Optimum with Recourse

2.1 Introduction

Static traffic assignment models discussed in Chapter 1 assume that travelers select routes *a priori*. However, in practice, uncertainty in network conditions encourages travelers to update their routes in an online manner. When the major source of uncertainty is in the "supply side", links in the network may be modeled using different states (perhaps representing accident conditions, vehicle breakdowns, special events, poor weather, rail-road crossings, temporary bottlenecks due to freight deliveries etc.) with different congestion functions (e.g., representing different capacity or free-flow speeds). However, such selfish routing of drivers is bound to be inefficient and the goal of this chapter is to extend Pigouvian pricing (Pigou, 1920) to minimize the expected system travel time in situations where users adaptively select links *en route*. When tolls change as a function of network states, drivers arriving at a node typically learn the adjacent link-states (and tolls) and choose which of those links to travel on to minimize their expected travel times. Although this assumption may appear far-fetched in the context of human drivers, it is possible for connected, autonomous vehicles to compute and rationally follow an optimal routing policy. Furthermore, connected autonomous vehicles would make it feasible for a network manager to collect and vary tolls on each link depending on the network conditions. Since generalized costs are a function of flows, different drivers will use different policies, which is likely to lead to an equilibrium at which point all used policies between an origin-destination (OD) pair have equal and minimal expected generalized costs. The objective is therefore to align this equilibrium flow solution, also dubbed as user equilibrium with recourse (UER), with the system optimal solution. The UER model was first formulated for acyclic networks (Unnikrishnan and Waller, 2009; Unnikrishnan, 2008; Ukkusuri, 2005) and later extended to cyclic networks in Boyles (2009) and Boyles and Waller (2010). Similar policy-based routing approaches were studied within the framework of dynamic traffic assignment (DTA) models (Hamdouch et al., 2004; Gao, 2012; Ma et al., 2016). However, solution algorithm correctness and properties such as equilibrium existence are difficult to show with simulation-based DTA models. Furthermore, it is also unclear if these models scale well with the problem size. The idea of policy-based routing and assignment can also be found in literature on transit networks (Hamdouch and Lawphongpanich, 2008, 2010; Trozzi et al., 2013; Hamdouch et al., 2014).

The probability that a link exists in a particular state is assumed known from historic data and the proposed traffic flow model is static in the sense that we ignore the time dimension and model a fluid version or the "steady state" flow. This assumption is reasonable if the types of disruptions being modeled are non-recurrent and short in duration relative to the modeling period. (e.g., if we are modeling a three-hour peak period and a minor accident reduces capacity for 15 minutes, it is reasonable to assume that 1/12 of the travelers will observe the accident state and 11/12 of the travelers will not.) Hence, we assume that the states observed by travelers arriving at a node are *independent* of the states observed by any other traveler arriving at that node. Without this assumption, it can be shown that even special cases of this problem are NP-hard (Provan, 2003). However, this assumption may encourage cycling, an unlikely phenomenon, as revisits to a node would reset the probabilities of link-states. We avoid this issue by imposing restrictions on the class of policies used in the proposed models.

The main contribution of this chapter lies in the formulation of a system optimal counterpart to the UER model, which we will henceforth refer to as system optimal with recourse (SOR) and the development of a marginal cost pricing rule (with different tolls for different states), very similar to that used in traditional static traffic assignment, which can bring the UER and SOR states into alignment. The state-dependent tolls in the SOR model address externalities associated with non-recurring congestion just as static marginal tolls (Pigou, 1920) reflect externalities related to recurring congestion. In addition, we also devise a convenient method to obtain solutions to these models when travelers' policies are disallowed from having cycles up to a certain length.

The SOR model should be distinguished from two other models which are superficially similar but in fact are substantially different. One other type of model defines stochastic states for the entire *network*, not individual links, and then solves a deterministic system optimal traffic assignment for each of these states. For instance, in the network in Figure 1.1 in Chapter 1, this model would involve solving two deterministic system optimal problems. This approach can quickly grow intractable for large networks (since the number of network states is exponential in network size), and reflects a different behavior assumption where all drivers are informed of the complete network state before departing, rather than receiving information incrementally.

Another type of model would solve for the system optimal assignment under expected conditions and have drivers begin following those paths, recalculating system optimal paths from their current location whenever information is received. That approach assumes that drivers do not anticipate receiving information and handle messages reactively, rather than proactively; an example in Waller and Ziliaskopoulos (2002) shows how that strategy can lead to suboptimal solutions.

The rest of the chapter is organized as follows. Section 2.2 introduces notation and describes the stochastic network model. In section 2.3, we formally define the UER and SOR problems, and in particular show that UER problem can be formulated as a Beckmann-like convex program. We also introduce a marginal cost pricing scheme that can internalize the congestion externalities in UER and result in a SOR state. In Section 2.4, we detail the algorithms that can be used to compute the SOR solution and the optimal tolls. We then propose a a more realistic SOR model that restricts cycling in the policies used by travelers and suggest a network transformation for finding the optimal tolls. Section 2.5 contains some numerical experiments on the Sioux Falls test network and in Section 2.6 we summarize the findings in this chapter.

2.2 Preliminaries

The approach to finding an equilibrium under supply-side uncertainty is similar to that used in deterministic TAPs. We begin by loading travelers on to the optimal policies, update the costs on all link states, recompute the optimal policies, and shift a fraction of the travelers to the new policy. These steps are repeated (see Figure2.1) until a certain convergence criteria is met. However, computing optimal policies (or even finding the cost of a given policy) and network loading are not as easy as with deterministic equilibrium problems. In deterministic TAPs, given a path, its cost can be simply computed by adding the costs of all the links that belong to the path. Further, loading travelers onto a path is also easy as one needs to just increase the corresponding flow on all the links belonging to the path. In the remainder of this section, we address the problem of estimating policy costs and network loading more formally.

Consider a strongly connected transportation network $G = (N, A)$ with sets of nodes N and arcs A. Let $Z \subseteq N$ represent the subset of nodes where trips begin and end. Let $\Gamma(i)$ and $\Gamma^{-1}(i)$ denote the downstream and upstream nodes of node i respectively. For any $(u, v) \in Z^2$, let d_{uv} be the demand

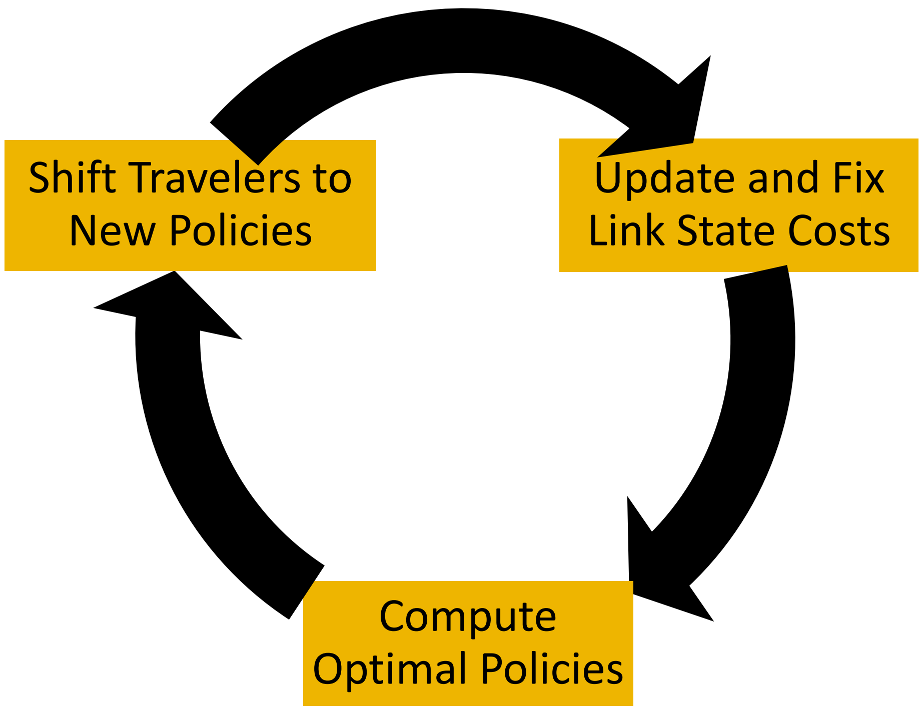


Figure 2.1: Equilibrium framework involving policies.

from origin u to destination v. Each arc $(i, j) \in A$ is associated with a set of states S_{ij} the arc can exist in; the link performance function for state $s \in S_{ij}$ is $t^s_{ij}(x^s_{ij})$, assumed positive and strictly increasing, where x^s_{ij} is the number of travelers using link (i, j) in state s (often called the link-state). Let $S = \cup_{(i,j)\in A} S_{ij}$ represent the set of all link-states in the network. Let $|N|$ and $|S|$ denote the number of nodes and the total number of link-states in the network respectively.

Upon arriving at any node i, a traveler observes a message vector $\theta \in \Theta_i = \times_{(i,j)\in A} S_{ij}$ informing him or her of the state of each link leaving node i, where Θ_i denotes the set of possible messages that can be received at node i. We will denote the state of link (i, j) corresponding to message θ using θ_{ij} or simply as s when it is clear from the context. Let q^θ be the probability of receiving message $\theta \in \Theta_i$ when arriving at node i. To simplify the notation, assume that the state of each link is independent of the state of other adjacent links; in this case, there exist $p^{\theta_{ij}}_{ij}$ such that $q^\theta_i = \prod_{(i,j)\in A} p^{\theta_{ij}}_{ij}$.

Define the set of node-states $\Phi = \{(i, \theta) : i \in N, \theta \in \Theta_i\}$; these correspond exactly to the decision points in the network, providing the location of a traveler and the message he or she just received. A policy $\pi : \Phi \to N$ is a function that maps each node-state to the node associated with the node-state (if we wish to terminate a trip) or a downstream node.

Associated with each policy π is a Markov chain, on the set of nodes N, with a transition matrix $\mathbf{R}_\pi \in \mathbb{R}^{|N|\times|N|}_+$ that represents the probabilities of moving from each node to any other;[1] its elements are $\mathbf{R}_\pi(i,j) = \sum_{\theta \in \Theta_i : \pi(i,\theta)=j} q^\theta$. A policy is said to be *cyclic* if the probability of revisiting any node is positive. A cyclic policy is said to have a cycle of length m if there exist a node that can be revisited with positive probability by traversing exactly m unique arcs. A policy that is not cyclic is referred to as an *acyclic* policy. An optimal policy, as we will see shortly, can be cyclic because of the full-reset assumption. However, we believe that the phenomena of cycling is unlikely to occur in practice, and we will address this modeling artifact in greater detail in Section 2.4.2.

A policy π *terminates at* i if the only eigenvector of $\mathbf{R}_\pi$ is the i-th standard basis $\mathbf{e}_i^T$, and is *non-waiting* if $\pi(i,\theta) = i$ only if π terminates at i. For any destination $v \in Z$, let Π_v denote the set of non-waiting policies terminating at v. From here, we restrict attention to non-waiting policies, that is, our model does not allow waiting at intermediate notes except the destination. Let $\Pi = \cup_{v \in Z} \Pi_v$.

Consider a policy $\pi \in \Pi_v$. Define $\rho_{ij}^{s\pi} = \sum_{\theta \in \Theta_i : \pi(i,\theta)=j, \theta_{ij}=s} q^\theta$ as the probability of leaving node i via link (i,j) in state $s \in S_{ij}$. Suppose the travel times for each link-state were fixed and denoted using t_{ij}^s. We will later consider the case where the travel times depend on flows and until then write it without reference to its flows. The expected travel time C_i^π from each node i to the destination

[1] The transition probabilities are not defined between pairs of node-states but instead represents the probabilities for moving between pairs of aggregated node-states which are collections of all node-states associated with a node (Boyles and Rambha, 2016).

v via routing policy π can be calculated using the following equations

$$C_v^\pi = 0 \tag{2.1}$$

$$C_i^\pi = \sum_{j \in \Gamma(i)} \sum_{s \in S_{ij}} \rho_{ij}^{s\pi}(t_{ij}^s + C_j^\pi) \qquad \forall\, i \in N \backslash \{v\} \tag{2.2}$$

Introducing $\mathcal{C}_{ij}^{s\pi}$ to be the expected travel time to the destination v for a traveler starting at the upstream end of on link (i, j) in state s and following routing policy π, we have

$$\mathcal{C}_{ij}^{s\pi} = t_{ij}^s + C_j^\pi \qquad \forall\, (i, j) \in A, s \in S_{ij} \tag{2.3}$$

or, upon eliminating the C variables, as the system

$$\mathcal{C}_{ij}^{s\pi} = t_{ij}^s + \sum_{k \in \Gamma(j)} \sum_{\bar{s} \in S_{jk}} \rho_{jk}^{\bar{s}\pi} \mathcal{C}_{jk}^{\bar{s}\pi} \qquad \forall\, (i, j) \in A, s \in S_{ij} \tag{2.4}$$

This equation can be expressed in matrix form as

$$\boldsymbol{\mathcal{C}}^\pi = \mathbf{t} + \mathbf{P}_\pi \boldsymbol{\mathcal{C}}^\pi \tag{2.5}$$

where $\boldsymbol{\mathcal{C}}^\pi \in \mathbb{R}_+^{|S| \times 1}$, $\mathbf{t} \in \mathbb{R}_+^{|S| \times 1}$, and $\mathbf{P}_\pi \in \mathbb{R}_+^{|S| \times |S|}$.

For example, consider the following network from Waller and Ziliaskopoulos (2002) in which a user is traveling from node 1 to 4. Suppose that the travel times on all arcs except $(3, 4)$ is deterministic. Arc (3,4) is assumed to exist in two states. For now, the travel times are assumed to be fixed and are not flow-dependent. Consider the policy in which the user takes arc (3,4) only if its cost is 1 and returns to node 3 via nodes 1 and 2 otherwise.

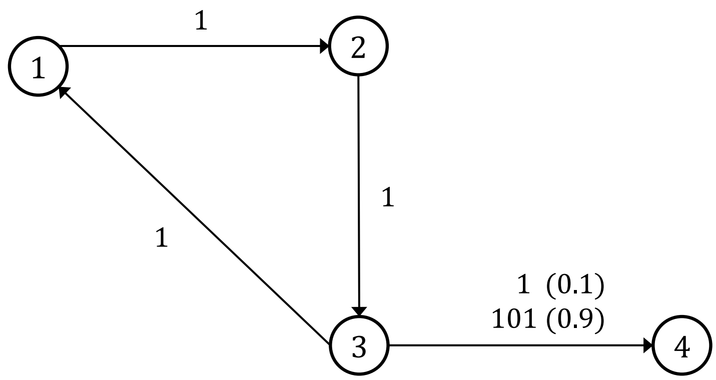


Figure 2.2: Computing the cost of a policy.

For the assumed policy, the $\mathbf{P}_\pi$ matrix may be populated as shown in equation (2.6).

$$\mathbf{P}_\pi = \begin{array}{c} \\ (1,2),[1] \\ (2,3),[1] \\ (3,1),[1] \\ (3,4),[1] \\ (3,4),[101] \end{array} \begin{array}{c} \begin{array}{ccccc} (1,2),[1] & (2,3),[1] & (3,1),[1] & (3,4),[1] & (3,4),[101] \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{2.6}$$

Since, $\mathbf{t} = (1\ 1\ 1\ 1\ 101)^T$, the cost of the policy is $\boldsymbol{\mathcal{C}}^\pi = (\mathbf{I} - \mathbf{P}_\pi)^{-1}\mathbf{t} = (30\ \ 29\ \ 31\ \ 1\ \ 101)^T$. Thus, the expected cost of reaching the destination from the origin is 30.

Let y_i^π denote the number of travelers originating at node i and choosing policy π. (We set y_i^π to 0 if $i \notin Z$ or if $i = v$.) Flow conservation requires $y_u^\pi \geq 0$

for all origins u and policies π, and $d_{uv} = \sum_{\pi \in \Pi_v} y_u^\pi$. Note that we employ a destination-based aggregation of policies; the origin of travelers is irrelevant for describing their choice at each node-state. A vector $\mathbf{y}^\pi \in \mathbb{R}_+^{|N|}$ is feasible if it satisfies flow conservation. Any feasible $\mathbf{y}^\pi$ defines a vector $\boldsymbol{\eta}^\pi \in \mathbb{R}_+^{|N|}$ of node-flows, with components η_i^π denoting the number of travelers arriving at node i using policy π, as well as the vector $\mathbf{x}^\pi \in \mathbb{R}_+^{|S| \times 1}$, whose components $x_{ij}^{s\pi}$ denote the number of travelers on policy π who experience link (i, j) in state s, through the linear system:

$$x_{ij}^{s\pi} = \rho_{ij}^{s\pi} \eta_i^\pi \qquad \forall (i, j) \in A, \pi \in \Pi \tag{2.7}$$

$$\eta_i^\pi = y_i^\pi + \sum_{h \in \Gamma^{-1}(i)} \sum_{\bar{s} \in S_{hi}} x_{hi}^{\bar{s}\pi} \qquad \forall i \in N, \pi \in \Pi \tag{2.8}$$

Then, eliminating the η variables yields a system of equations in the link state flows alone:

$$x_{ij}^{s\pi} = \rho_{ij}^{s\pi} y_i^\pi + \rho_{ij}^{s\pi} \sum_{h \in \Gamma^{-1}(i)} \sum_{\bar{s} \in S_{hi}} x_{hi}^{\bar{s}\pi} \qquad \forall\, (i, j) \in A, s \in S_{ij}, \pi \in \Pi \tag{2.9}$$

This equation can be expressed in matrix form as

$$\mathbf{x}^\pi = \mathbf{b}^\pi + \mathbf{P}_\pi^T \mathbf{x}^\pi \tag{2.10}$$

where $\mathbf{b}^\pi = \mathrm{vec}(\rho_{ij}^{s\pi} y_i^\pi) \in \mathbb{R}_+^{|S| \times 1}$. Thus, we may write $\mathbf{x}^\pi = \mathbf{A}_\pi^{-1} \mathbf{b}^\pi$ where $\mathbf{A}_\pi = (\mathbf{I} - \mathbf{P}_\pi^T)$. Note that the columns of $\mathbf{A}_\pi^{-1}$ denote the expected number of times each link-state is visited for a traveler starting at a specific link-state and following policy π. This can be seen by solving (2.10) for $\mathbf{x}^\pi$ and substituting standard basis vectors on the right-hand side. Given a policy π and feasible

$\mathbf{y}$, the corresponding $\boldsymbol{\eta}^{\pi}$ and $\mathbf{x}^{\pi}$ values can be identified by either solving the linear system directly (in transportation networks, this system is usually sparse) or by applying a network algorithm such as that in Boyles (2009).

For example, in the network introduced in Figure 2.2, suppose that the demand between nodes 1 and 4 is 1 and assume that all travelers follow the policy described earlier. Then, the link flows can be computed by first sending the 1 unit of demand along arcs (1,2) and (2,3). Upon reaching node 3, 10% of the travelers observe arc (3,4) in state s_1 and head to the destination. The remaining 90% take arc (3,1) as shown in the top panel of Figure 2.3. This process can be repeated for the 0.9 units of demand that cycles back to node 3 and for the 0.81 units of demand that cycles twice (see middle and bottom panels of Figure 2.3).

Since the policy followed by the travelers admits an infinite number of cycles, the flow on each link can be represented as a sum of a geometric series as shown in Figure 2.4.

Alternately, we can solve the flow conservation equations $\mathbf{x}^{\pi} = (\mathbf{I} - \mathbf{P}_{\pi}^{T})^{-1}\mathbf{b}^{\pi}$, described by equation (2.11), to obtain link-state flows.

$$\left[\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix} - \begin{pmatrix}0&0&1&0&0\\1&0&0&0&0\\0&0.9&0&0&0\\0&0.1&0&0&0\\0&0&0&0&0\end{pmatrix}\right]^{-1}\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} = \begin{pmatrix}10\\10\\9\\1\\0\end{pmatrix} \tag{2.11}$$

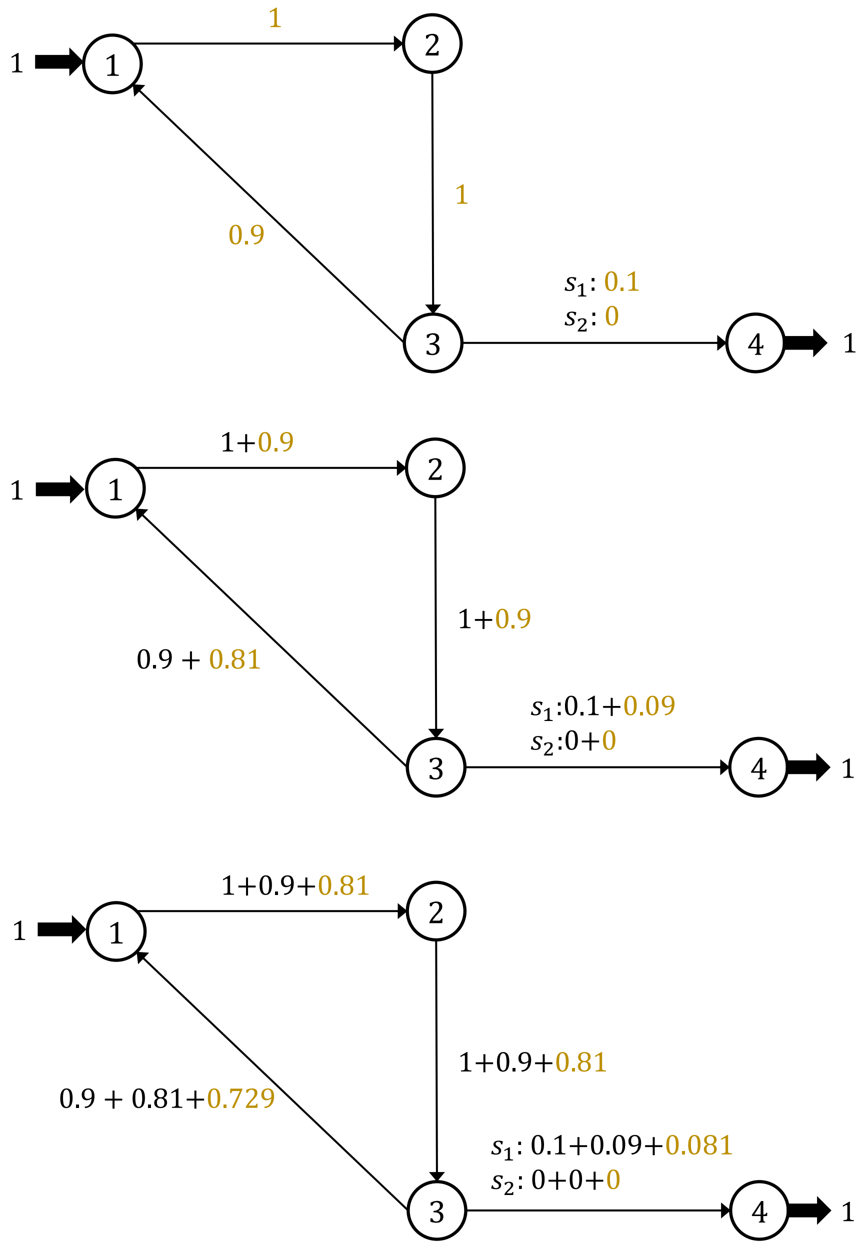


Figure 2.3: Network loading travelers iteratively.

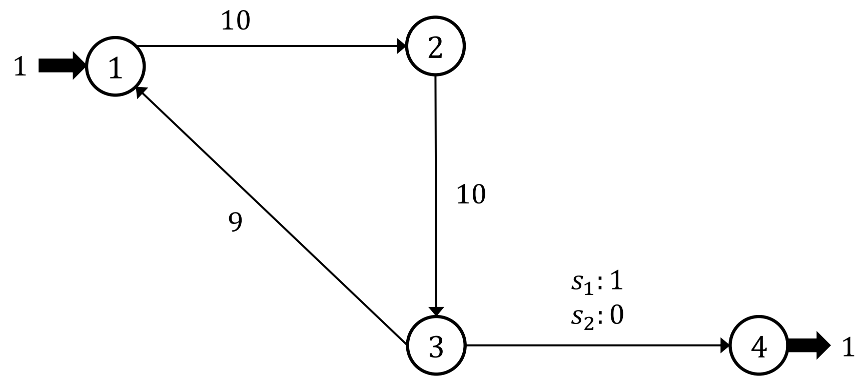


Figure 2.4: Link-state flows after network loading.

2.3 SOR and Marginal Cost Pricing

Let $\mathbf{y}$ denote the vector $(\mathbf{y}^\pi)_{\pi \in \Pi_v}$ and let $\mathbf{x} = (x^s_{ij})_{(i,j) \in S, s \in S_{ij}}$ denote link flows for each state aggregated by policies. As described above, every feasible policy flow vector $\mathbf{y}$ determines aggregate link flows by state $\mathbf{x}$, which in turn determine link travel times by state $\mathbf{t}$ through the link performance functions, which finally determine the policy costs $\boldsymbol{\mathcal{C}}^\pi$; thus we may write the policy costs as a function of the policy flows: $\boldsymbol{\mathcal{C}}^\pi(\mathbf{y})$. The system-optimal with recourse problem is to find $\mathbf{y}$ minimizing the TETT

$$\text{TETT} = \sum_{(u,v) \in Z^2} \sum_{\pi \in \Pi_v} y^\pi_u C^\pi_u(\mathbf{y}) \tag{2.12}$$

$$= \sum_{v \in Z} \sum_{\pi \in \Pi_v} \sum_{i \in N} y^\pi_i C^\pi_i(\mathbf{y}) \tag{2.13}$$

$$= \sum_{\pi \in \Pi} \sum_{i \in N} y^\pi_i \sum_{(i,j) \in \Gamma(i)} \sum_{s \in S_{ij}} \rho^{s\pi}_{ij} \mathcal{C}^{s\pi}_{ij}(\mathbf{y}) \text{ [using (2.2) and (2.3)]} \tag{2.14}$$

$$= \sum_{\pi \in \Pi} \sum_{(i,j) \in A} \sum_{s \in S_{ij}} y^\pi_i \rho^{s\pi}_{ij} \mathcal{C}^{s\pi}_{ij}(\mathbf{y}) \tag{2.15}$$

$$= \sum_{\pi \in \Pi} (\boldsymbol{\mathcal{C}}^{\pi}(\mathbf{y}))^T \mathbf{b}^{\pi} \tag{2.16}$$

$$= \sum_{\pi \in \Pi} (\boldsymbol{\mathcal{C}}^{\pi}(\mathbf{y}))^T \mathbf{A}_{\pi} \mathbf{A}_{\pi}^{-1} \mathbf{b}^{\pi} \tag{2.17}$$

$$= \sum_{\pi \in \Pi} (\mathbf{A}_{\pi}^T \boldsymbol{\mathcal{C}}^{\pi}(\mathbf{y}))^T \mathbf{x}^{\pi} \text{ [using (2.10)]} \tag{2.18}$$

$$= \sum_{\pi \in \Pi} \sum_{(i,j) \in A} \sum_{s \in S_{ij}} t_{ij}^s(x_{ij}^s) x_{ij}^{s\pi} \text{ [using (2.5)]} \tag{2.19}$$

$$= \sum_{(i,j) \in A} \sum_{s \in S_{ij}} t_{ij}^s(x_{ij}^s) x_{ij}^s \tag{2.20}$$

Since $\mathbf{x}$ is related to $\mathbf{y}$ by a linear system, and since each $t_{ij}^s(\cdot)$ is assumed strictly increasing, this latter reformulation shows that the system-optimal with recourse problem is a convex program with a strictly convex objective function with a unique optimal solution. Specifically, the SOR problem can be formulated as

$$
\begin{aligned}
\min_{\mathbf{y},\mathbf{x},\mathbf{x}^{\pi},\mathbf{b}^{\pi}} \quad & \sum_{(i,j) \in A} \sum_{s \in S_{ij}} x_{ij}^s t_{ij}^s(x_{ij}^s) && \text{(SOR)} && (2.21)\\
\text{s.t.} \quad & \sum_{\pi \in \Pi_v} y_u^{\pi} = d_{uv} && \forall (u,v) \in Z^2 && (2.22)\\
& \sum_{\pi \in \Pi} x_{ij}^{s\pi} = x_{ij}^s && \forall (i,j) \in A, s \in S_{ij} && (2.23)\\
& \mathbf{A}_{\pi} \mathbf{x}^{\pi} = \mathbf{b}^{\pi} && \forall \pi \in \Pi && (2.24)\\
& y_u^{\pi} \geq 0 && \forall \pi \in \Pi, u \in Z && (2.25)
\end{aligned}
$$

The SOR state is one in which all routing choices are made to minimize expected travel time for the entire system. This state is not likely to arise spontaneously, since drivers do not typically have enough information to determine which routing policy they should follow to minimize total expected travel time, and furthermore have no incentive to do so even if such information were available. The UER state corresponds to a decentralized, Nash equilibrium in which individual (nonatomic) drivers choose a policy which minimizes their own expected travel time to the destination. The UER state is based on a generalization of Wardrop's principle: all used policies between any origin and destination have equal and minimal expected travel time. That is, UER policy flows $\mathbf{y}$ are feasible and satisfy

$$y_u^\pi > 0 \Rightarrow C_u^\pi(\mathbf{y}) = \min_{\pi' \in \Pi_v} C_u^{\pi'}(\mathbf{y}) \qquad \forall v \in Z, \pi \in \Pi_v \,. \tag{2.26}$$

While intuitive, this definition is not particularly useful for finding UER policy flows. To this end, the following convex program is provided:

$$\begin{aligned}
\min_{\mathbf{y},\mathbf{x},\mathbf{x}^\pi,\mathbf{b}^\pi} \quad & \sum_{(i,j)\in A} \sum_{s \in S_{ij}} \int_0^{x_{ij}^s} t_{ij}^s(x)\, dx && \text{(UER)} && (2.27)\\
\text{s.t.} \quad & \sum_{\pi \in \Pi_v} y_u^\pi = d_{uv} && \forall (u,v) \in Z^2 && (2.28)\\
& \sum_{\pi \in \Pi} x_{ij}^{s\pi} = x_{ij}^s && \forall (i,j) \in A, s \in S_{ij} && (2.29)\\
& \mathbf{A}_\pi \mathbf{x}^\pi = \mathbf{b}^\pi && \forall \pi \in \Pi && (2.30)\\
& y_u^\pi \geq 0 && \forall \pi \in \Pi, u \in Z && (2.31)
\end{aligned}$$

Proposition 2.1. *The optimal solutions to the convex program* (2.27)–(2.31) *correspond exactly to policy flows satisfying the UER definition* (2.26)*.*

Proof. The proof of this proposition generalizes the proof by Unnikrishnan and Waller (2009) to cyclic networks and proceeds along similar lines as the proof that the Beckmann formulation yields user equilibrium solutions; however, the use of policies presents some additional technicalities. Specifically, in UER, link flows are obtained from policy flows by solving an implicit linear system, rather than obtaining link flows by directly summing flows on paths which use that link, as in Beckmann's formulation.

Begin by Lagrangianizing the flow conservation constraints (2.28) (with multipliers $\boldsymbol{\kappa}$), and substitute constraints (2.29) and (2.30) into the objective function, expressing it in terms of $\mathbf{y}$ alone (note that the $\mathbf{b}^\pi$ vector depends only on $\mathbf{y}$). This yields the Lagrangian

$$\mathcal{L}(\mathbf{y}, \boldsymbol{\kappa}) = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int_0^{\sum_{\pi\in\Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_\pi^{-1} \mathbf{b}^\pi} t_{ij}^s(x)\ dx + \sum_{(u,v)\in Z^2} \kappa_{uv}(d_{uv} - \sum_{\pi\in\Pi_v} y_u^\pi) \tag{2.32}$$

with only non-negativity constraints on each y^π, where $\mathbf{e}_{ij}^s \in \mathbb{R}_+^{|\mathcal{S}|\times 1}$ is a standard basis vector. Referring to $t_{ij}^s \left(\sum_{\pi\in\Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_\pi^{-1} \mathbf{b}^\pi\right)$ as t_{ij}^s for brevity, the resulting Karush-Kuhn-Tucker (KKT) conditions are

$$\sum_{(i,j)\in A} \sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi} \left(\sum_{\bar\pi\in\Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_{\bar\pi}^{-1} \mathbf{b}^{\bar\pi} \right) - \kappa_{uv} \geq 0 \qquad \forall \pi \in \Pi_v \tag{2.33}$$

$$y_u^\pi \left(\sum_{(i,j)\in A} \sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi} \left(\sum_{\bar\pi\in\Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_{\bar\pi}^{-1} \mathbf{b}^{\bar\pi} \right) - \kappa_{uv} \right) = 0 \qquad \forall \pi \in \Pi_v, u \in Z \tag{2.34}$$

$$\sum_{\pi\in\Pi_v} y^\pi = d_{uv} \qquad \forall (u,v) \in Z^2 \tag{2.35}$$

$$y_u^\pi \geq 0 \qquad \forall \pi \in \Pi, u \in Z \tag{2.36}$$

In order to establish an equivalence between the KKT and UER conditions, we proceed by showing that $\sum_{(i,j)\in A}\sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) = C_u^\pi(\mathbf{y})$. Therefore, equations (2.33) and (2.34) would translate to $C_u^\pi(\mathbf{y}) - \kappa_{uv} \geq 0$ and $y_u^\pi(C_u^\pi(\mathbf{y}) - \kappa_{uv}) = 0$, implying that κ_{uv} is the least expected time among all policies terminating at v and a policy terminating at v is used by travelers leaving u iff its expected travel time equals κ_{uv}.

Let $\mathbf{d} = \left(\frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right)\right)_{\substack{(i,j)\in A,\\ s\in S_{ij}}}$ be a vector in $\mathbb{R}^{|\mathcal{S}|\times 1}$. We may therefore write

$$\sum_{(i,j)\in A}\sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) = \mathbf{t}^T\mathbf{d} \tag{2.37}$$

$$= \left((\mathbf{I}-\mathbf{P}_\pi)\boldsymbol{\mathcal{C}}^\pi(\mathbf{y})\right)^T\mathbf{d} \text{ [using (2.5)]} \tag{2.38}$$

$$= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T(\mathbf{I}-\mathbf{P}_\pi)^T\mathbf{d} \tag{2.39}$$

$$= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T\mathbf{A}_\pi\mathbf{d} \tag{2.40}$$

Notice that elements of $\mathbf{d}$ can be simplified as

$$\frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) = \frac{\partial}{\partial y_u^\pi}\left((\mathbf{e}_{ij}^s)^T\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) \tag{2.41}$$

$$= (\mathbf{e}_{ij}^s)^T\frac{\partial}{\partial y_u^\pi}\left(\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) \tag{2.42}$$

which implies that $\mathbf{d} = \mathbf{I}\dfrac{\partial}{\partial y_u^\pi}\left(\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) = \mathbf{A}_\pi^{-1}\dfrac{\partial\mathbf{b}^\pi}{\partial y_u^\pi}$. Therefore, equation (2.40) can be rewritten as

$$\sum_{(i,j)\in A}\sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) = (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T\mathbf{A}_\pi\mathbf{A}_\pi^{-1}\frac{\partial\mathbf{b}^\pi}{\partial y_u^\pi} \tag{2.43}$$

$$= (\boldsymbol{\mathcal{C}}^{\pi}(\mathbf{y}))^T \frac{\partial \mathbf{b}^{\pi}}{\partial y_u^{\pi}} \tag{2.44}$$

$$= \sum_{(u,j)\in A} \sum_{s \in S_{uj}} \rho_{uj}^{s\pi} \mathcal{C}_{uj}^{s\pi}(\mathbf{y}) \tag{2.45}$$

$$= C_u^{\pi}(\mathbf{y}) \text{ [using (2.2) and (2.3)]} \tag{2.46}$$

Therefore, a solution to the convex program (2.27)–(2.31) satisfies the UER condition. ■

Notice that convex programs (UER) and (SOR) differ *only* in the objective functions; the constraints are exactly identical.[2] In the traditional traffic assignment problem, adding a "marginal cost" toll of $x_{ij}t'_{ij}(x_{ij})$ to each link brings the user equilibrium and system optimum states into alignment. Likewise, in the UER framework, adding a toll of $x_{ij}^s(t_{ij}^s)'(x_{ij}^s)$ to each *link-state* brings the UER and SOR states into alignment, as shown in the following result. In other words, to achieve the system optimum, the network manager may employ a responsive tolling scheme in which the state of each link is observed and the associated marginal toll is collected. Define the *tolled* link performance functions $\hat{t}_{ij}^s$ as $\hat{t}_{ij}^s(x_{ij}^s) = t_{ij}^s(x_{ij}^s) + x_{ij}^s(t_{ij}^s)'(x_{ij}^s)$.

Proposition 2.2. *An feasible solution to the convex program (UER) with respect to tolled link performance functions $\hat{t}_{ij}^s$ is an optimal solution to (UER) if*

[2] Since we assume strictly increasing and positive link delay functions, the UER and SOR objectives are strictly convex in link-state flows just like the Beckmann function in deterministic traffic assignment models (see Sheffi, 1985, chap. 3). Thus, the optimal link-state flows are unique but multiple policy flow solutions may exist.

and only if it is optimal to (SOR) with respect to the original link performance functions t_{ij}^s.

Proof. Consider a generic term $\int_0^{x_{ij}^s} \hat{t}_{ij}^s(x)\ dx$ in the objective function (2.27). Using the definition of $\hat{t}_{ij}^s$, this can be rewritten as

$$\int_0^{x_{ij}^s} t_{ij}^s(x)\ dx + \int_0^{x_{ij}^s} x(t_{ij}^s)'(x)\ dx\,. \tag{2.47}$$

Integrating the second term by parts, we have

$$\int_0^{x_{ij}^s} xt'_{ij}(x)\ dx = x_{ij}^s t_{ij}^s(x_{ij}^s) - \int_0^{x_{ij}^s} t_{ij}^s(x)\ dx \tag{2.48}$$

which, upon substitution into (2.47) shows that $\int_0^{x_{ij}^s} \hat{t}_{ij}^s(x)\ dx = x_{ij}^s t_{ij}^s(x_{ij}^s)$. That is, the objective functions for (UER) with respect to $\hat{t}$ and (SOR) with respect to t are identical. Since these programs have identical feasible regions, the set of optimal solutions are also identical. ■

2.4 Solution Methods

The UER and SOR models defined in Section 2.3 were formulated as convex optimization problems. This lets us use standard algorithms such as the method of successive averages (MSA) and the Frank-Wolfe (FW) algorithm (Sheffi and Powell, 1982; Frank and Wolfe, 1956) for finding the optimal solutions. Since these methods operate in the space of link-states, the memory requirements are very modest and the SOR problem can be solved without policy enumeration, much as the traditional system optimal problem can be solved without path enumeration.

In this section, we present the FW algorithm and the temporal dependence-online shortest path (TD-OSP) algorithm of Waller and Ziliaskopoulos (2002). The latter is used to find the optimal policies for an all-or-nothing assignment within each FW iteration. We then illustrate the issue of cycling using a small example and suggest a network transformation which eliminates cycles of certain lengths from all routing policies. Furthermore, we calculate the optimal state-dependent tolls for instances in which cycling is restricted by applying the FW algorithm, with minor changes, to the transformed network.

2.4.1 Frank-Wolfe and Online Shortest Paths

The Frank-Wolfe method is a gradient descent-type algorithm for solving non-linear convex optimization problems. We begin by initializing the travel times on all links for all link-states to their free flow travel times and use the TD-OSP algorithm (which we will explain shortly) for a given destination v to obtain a policy $\pi^* \in \Pi_v$ and cost vector $(C_i^{\pi^*})_{i \in N}$ which satisfies equations (2.1) and (2.2). After repeating this step for all destinations, the resulting policies and the OD-demand are used to construct the $\mathbf{A}_{\pi^*}$ and $\mathbf{b}^{\pi^*}$ matrices which help determine the link flows for each state for each policy via $\mathbf{x}^{\pi^*} = \mathbf{A}_{\pi^*}^{-1}\mathbf{b}^{\pi^*}$. The links flows are then aggregated to obtain $\mathbf{x}^*$ using which the generalized travel costs for different link-states are updated.

The TD-OSP algorithm is used again keeping these travel times fixed to obtain a new policy and a cost vector, which is in turn used to compute new state-dependent link flows. This flow solution is a descent direction and an optimal step size is used to compute a convex combination of the old and the new state-dependent link flows.

Algorithm 1 FRANK-WOLFE$(G, \mathbf{d})$

Step 1: Initialize
$\mathbf{x} \leftarrow \mathbf{0}$
$\mathbf{t} \leftarrow \hat{\mathbf{t}}(\mathbf{0})$
GAP $\leftarrow \infty$

Step 2:
while GAP $> \epsilon$ **do**
  $\mathbf{x}^* \leftarrow \mathbf{0}$
  for $v \in Z$ **do** ▷ All-or-Nothing Assignment
    $\pi_v^* \leftarrow$ TD-OSP$(G, \mathbf{t}, v)$
    $\mathbf{x}^* \leftarrow \mathbf{x}^* + \mathbf{A}_{\pi_v^*}^{-1} \mathbf{b}^{\pi_v^*}$
  end for
  GAP $\leftarrow (\mathbf{t} \cdot \mathbf{x}) \left(\sum_{u \in Z} \sum_{v \in Z} d_{uv} C_u^{\pi_v^*} \right)^{-1} - 1$
  $\varphi^* \leftarrow \arg\min_{\lambda \in [0,1]} \sum_{(i,j) \in A} \sum_{s \in S_{ij}} \int_0^{(1-\lambda) x_{ij}^s + \lambda x_{ij}^{s*}} \hat{t}_{ij}^s(x) dx$ ▷ Optimal Step Size
  $\mathbf{x} \leftarrow (1 - \varphi^*) \mathbf{x} + \varphi^* \mathbf{x}^*$
  $\mathbf{t} \leftarrow \hat{\mathbf{t}}(\mathbf{x})$
end while

The optimal step size φ^* is obtained by finding the zeros of the function $\sum_{(i,j) \in A} \sum_{s \in S_{ij}} \hat{t}_{ij}^s((1 - \varphi) x_{ij}^s + \varphi x_{ij}^{s*})(x_{ij}^{s*} - x_{ij}^s)$ using the bisection or Newton's method. The TD-OSP algorithm Waller and Ziliaskopoulos (2002) used to compute an optimal policy for the all-or-nothing assignment is essentially a value iteration approach, the pseudocode for which is reproduced in Algorithm 2.

A scan eligible list (SEL) containing a subset of nodes whose labels could change is maintained at each iteration and is first initialized with the upstream nodes of the destination. The algorithm also initializes the labels of all nodes except the destination to a sufficiently large number. We then proceed by picking an element of the SEL and computing the cost and probability of choosing its downstream link-states (Step 2.1). The expected label of the element is then updated in Step 2.2 if the optimality conditions are not met and its upstream nodes are added to the SEL. Step 2 is carried out until the SEL is empty, after which the optimal policy is constructed in Step 3 using the converged labels.

Two main features of the algorithm make it very efficient compared to a regular value iteration algorithm. (1) Instead of updating the values of all states in each iteration, the algorithm maintains a scan eligible list similar to those used in labeling methods for shortest paths. (2) The number of states at a node i is $\Pi_{j \in \Gamma(i)} |S_{ij}|$, which is exponential. However, to compute the expected label of node i it suffices to find the probabilities with which each downstream arc is chosen in different states. The TD-OSP algorithm exploits this observation in calculating the expected label of a node using a recursive procedure because of which its complexity is pseudo-polynomial.

The notation [ ] in Algorithm 2 denotes an empty vector and $\boldsymbol{\xi}' \leftarrow [\boldsymbol{\xi}' \;\; \xi_k p_{ij}^s]$ is used to update the vector $\boldsymbol{\xi}'$ by appending a new element $\xi_k p_{ij}^s$. Similar notation is used to denote updates to $\boldsymbol{\lambda}'$. The subroutine Reduce( ) removes duplicates from $\boldsymbol{\lambda}'$ and adds up the probabilities in $\boldsymbol{\xi}'$ to compute the probability of occurrence of elements in $\boldsymbol{\lambda}'$. For instance if $\boldsymbol{\lambda}' = [5\ \ 7\ \ 8\ \ 8\ \ 5\ \ 1\ \ 6\ \ 1]$ and all the elements of the associated probability vector $\boldsymbol{\xi}'$ are equal to 0.125, then function $\textsc{Reduce}(\boldsymbol{\xi}', \boldsymbol{\lambda}')$ returns vectors $\boldsymbol{\xi} = [0.25\ \ 0.125\ \ 0.25\ \ 0.25\ \ 0.125]$ and $\boldsymbol{\lambda} = [5\ \ 7\ \ 8\ \ 1\ \ 6]$.

2.4.2 Restricting Cycles

The TD-OSP algorithm described earlier assumes full-reset, i.e., upon revisiting a node, a traveler sees a realization of the downstream states drawn from their assumed probability distributions, irrespective of previously realized arc costs. In other words, if there was a disruption on a link and the traveler revisits its head node, he/she will find it in a disrupted state with the prior probability of a disruption irrespective of the time between revisits. This assumption can lead to cycling in the optimal policy since travelers may revisit nodes in anticipation of an uncongested downstream arc.[3]

For example, consider the network in Figure 2.5. Suppose there are a total of 500 users traveling from node 1 to node 5. Assume that all links except (3,5) exist in one state with a free flow travel time of 10. Suppose that the capacity of links $(1, 2), (1, 3)$, and $(2, 3)$ is 100 and that of links $(3, 2), (3, 4)$, and $(4, 5)$ is 50. Let the arc (3,5) exist in two states s_1 and s_2 with free flow travel time 10

[3]Such behavior is unrealistic except in situations in which travelers search for parking (Tang et al., 2014; Boyles et al., 2015).

Algorithm 2 TD-OSP $(G, \mathbf{t}, v)$

```
Step 1: Initialize Labels
C_v ← 0
C_i ← ∞ ∀ i ∈ N\{v}
SEL ← Γ^{-1}(v)

Step 2:
while SEL ≠ ∅ do
    Remove i from SEL
    ξ ← [1], λ ← [∞]
    for j ∈ Γ(i) do                                   ▷ Step 2.1
        ξ' ← [ ], λ' ← [ ]
        for s ∈ S_ij do
            for k = 1, ..., |ξ| do
                ξ' ← [ξ'  ξ_k p^s_ij]
                if t^s_ij + C_i < a then
                    λ' ← [λ'  t^s_ij + C_i]
                else
                    λ' ← [λ'  λ_k]
                end if
            end for
        end for
        (ξ, λ) ← REDUCE(ξ', λ')
    end for

    if ξ · λ < C_i then                               ▷ Step 2.2
        C_i ← ξ · λ
        SEL ← SEL ∪ Γ^{-1}(i)
    end if
end while

Step 3: Choose Optimal Policy
for i ∈ N, θ ∈ Θ_i do
    if i = v then
        π*_v(i, θ) = v
    else
        π*_v(i, θ) ∈ arg min_{j∈Γ(i)} {t^{θ_ij}_ij + C_j}
    end if
end for
return π*_v
```

and capacities 400 and 50 with equal probability.[4] Suppose that the delay on each link for each state is given by the Bureau of Public Roads (BPR) function $t_{ij}^s(x_{ij}^s) = \tau_{ij}^s(1+0.15(x_{ij}^s/\mu_{ij}^s)^4)$, where τ_{ij}^s is the free flow travel time and μ_{ij}^s is the capacity of link (i,j) in state s. The TETT of the UER solution and the SOR solution at a gap of 10^{-4} are 113365 and 113183 respectively. The values next to the links in the left panel indicate the SOR flows and the optimal tolls are shown in the right panel. Notice that travelers arriving at node 3 can either reach node 5 via node 4 or cycle between nodes 2 and 3 before choosing a downstream arc. The flow on link (3,2) indicates that a total of 59.83 units of flow cycles before reaching the destination. Since travelers are unlikely to cycle, two alternate reformulations of the SOR and UER problems can be defined (1) assuming no-reset or (2) by assuming that travelers choose only acyclic policies. The no-reset model is however not realistic since different travelers see different states that never changes over time whereas in practice, supply-side changes are temporary. Note that in the optimal policies of the no-reset version, travelers may cycle though the states of the arcs encountered earlier do not change. In comparison, the second behavioral assumption is more reasonable. However, solving the SOR problem with acyclic policies (we will refer to the routing problem involving acyclic policies as the acyclic OSP

[4]Since the states s_1 and s_2 are observed only half the time, the μ values for these states (in the BPR function) must be appropriately adjusted. Thus, the assumed capacities of 400 and 50 vehicles per hour correspond to $\mu_{35}^{s_1} = 200$ and $\mu_{35}^{s_2} = 25$ vehicles per 1/2 hour respectively. Notice that changing units this way also ensures that the solution to a UER model with identical link capacities in both states is consistent with that of the regular user equilibrium assignment.

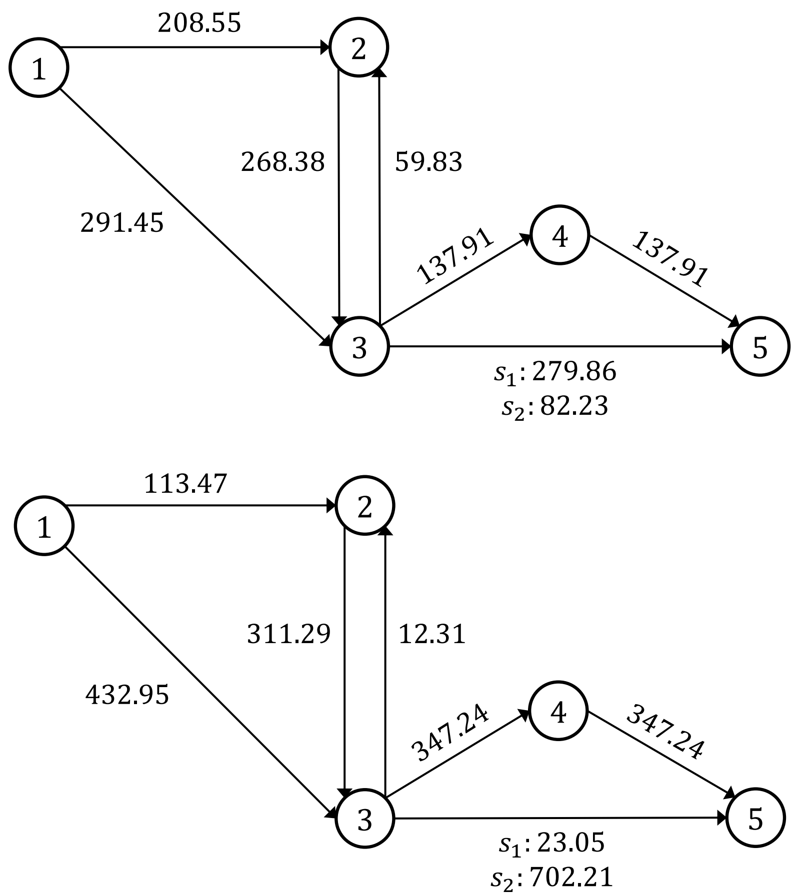


Figure 2.5: Optimal flows (top) and tolls (bottom) in a network that illustrates cycling.

problem) is difficult due to Proposition 2.4.2.

Proposition 2.3. *Acyclic OSP is NP-hard.*

Proof. The proof for this proposition is similar to that by Polychronopoulos and Tsitsiklis (1996) for the no-reset stochastic shortest path problem. We proceed by establishing a reduction from the directed Hamiltonian path problem. Consider a directed graph $\bar{G} = (\bar{N}, \bar{A})$ with node set $\bar{N}$ and arc set $\bar{A}$ with arcs of cost 0. Let $G' = (N', A')$ be an augmented graph with an additional node v' that can be reached directly from every node in $\bar{N}$ via an arc of cost 0 or 1 with equal probability. The optimal acyclic OSP from any node in G' would be to visit nodes in $\bar{N}$ until a node $i \in \bar{N}$ is found such that $(i, v') \in A'$ has a cost 0 or all the other nodes in $\bar{N}$ were visited and the arc cost to v' from those nodes was 1. Hence, one can construct a Hamiltonian path (if it exists) from the optimal policy of the acyclic OSP. ■

To address this issue, one option is to use a heuristic for the acyclic OSP problem by defining a *bush* using *reasonable links*. A reasonable link is one whose head node is closer to the destination than the tail. Closeness to the destination can be defined using distance or any other vector of node labels (such as the regular OSP labels). A bush is an acyclic subgraph in which the destination can be reached from all nodes. Solving the OSP problem on a bush will yield an acyclic policy which can then be used for an all-or-nothing assignment. However, there are two major problems with this approach. (1)

Fixing the bush and using the FW method will result in an equilibrium with respect to the subgraph and not the entire network. A similar issue can be found in literature on the logit based stochastic user equilibrium (Sheffi, 1985; Leurent, 1997). (2) Instead, if a different bush is used within each FW iteration (by defining reasonable links either using the OSP labels or expected link costs), a convergence criteria for equilibrium cannot be established since the routing policies are often suboptimal and thus do not yield all-or-nothing flow that is a descent direction. In fact, when we tested this method by defining reasonable links using the OSP labels for the original network, the relative gap was negative in several instances.[5]

Hence, solving the OSP subproblem to optimality is necessary to determine the system optimal flows and the optimal tolls. Since, the acyclic OSP problem is NP-hard, we instead compute policies which minimize expected generalized cost while permitting cycles above a certain length. This is achieved by modifying the state of the traveler in the online shortest path problem include a vector of m most recently visited nodes in addition to the node-message pair. Using this state definition, the action space at each state is modified by checking if one of the downstream nodes belongs to the set of m previously visited nodes. We will refer to this variant of the SOR and UER problem as the m-SOR and m-UER problems respectively. Thus, the used policies in the

[5] As an aside, notice that even if travelers used acyclic policies, a subnetwork with arcs belonging to all used policies can contain a cycle (unlike the regular traffic assignment). For this reason, faster equilibrium algorithms such as bush and origin based methods (Dial, 2006; Bar-Gera, 2002, 2010) cannot be easily modified to solve the UER and SOR problem.

m-SOR and m-UER solutions will not have cycles with at most $m+1$ arcs. For realistic problem instances, we suspect that one can completely avoid acyclic policies using small values of m since cycling among a large number of arcs is likely to result in increased expected travel time.

Solving the OSP problem with restrictions on the cycle length results in a larger transition matrix (since the states associated with the online routing problem also includes recently visited nodes) and for each policy and one could redefine the network loading process, with some difficulty, to obtain an equation similar to equation (2.10). Instead, in the remainder of this section, we propose a simpler network transformation that enables us to use the existing framework for the 0-SOR and 0-UER problems. The network transformation is broken into the following two phases.

Phase I: The first step in the network transformation is carried out to enumerate, for any node i, the set of all feasible vectors of the last m nodes in all paths that lead to node i. To this end, we add a dummy node X and connect it to all the nodes in the network including itself (see Figure 2.6). We then perform a breadth first search (BFS) for reaching node i and the distance labels (which represent the shortest number of arcs required to reach a node i) are used to enumerate the set of recently visited nodes M_i as proposed by Tang et al. (2014) (see Algorithm 3). Let $M_i(j)$ represent the set of nodes which can reach node i by traversing at most j arcs. For instance, in the network in Figure 2.6, when $m = 2$, $M_2(2) = \{1, 2, 3, X\}$. The dummy node X is useful

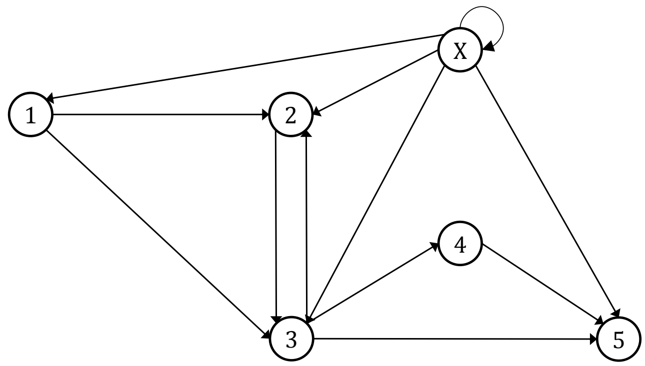


Figure 2.6: Network with dummy node to enumerate recently visited nodes.

in defining traveler states at the beginning sections of a trip when fewer than m nodes are visited. We will continue to refer to N and A as the nodes and arcs in the original network (before the addition of X).

Algorithm 3 ENUMERATE(G)

for $i \in N$ **do**
  Use BFS to find nodes that can reach i
  for $j = 0, 1, \ldots, m$ **do**
    Populate $M_i(j)$ using the BFS distance labels
  end for
  $M_i \leftarrow \times_{j=0}^{m} M_i(j)$
  Scan each element of M_i and discard infeasible paths
end for

Phase II: Define a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \cup_{i \in N} M_i \cup M$ and $M = N$. We use an alias M for the set N so that the latter can be used to refer to nodes in the original network. Throughout this section, we will use i and j to represents nodes in the original network and k and l for nodes in $\cup_{i \in N} M_i$. A regular arc in $\mathcal{G}$ is defined between node $k \in M_i$ and $l \in M_j$ if there is an arc

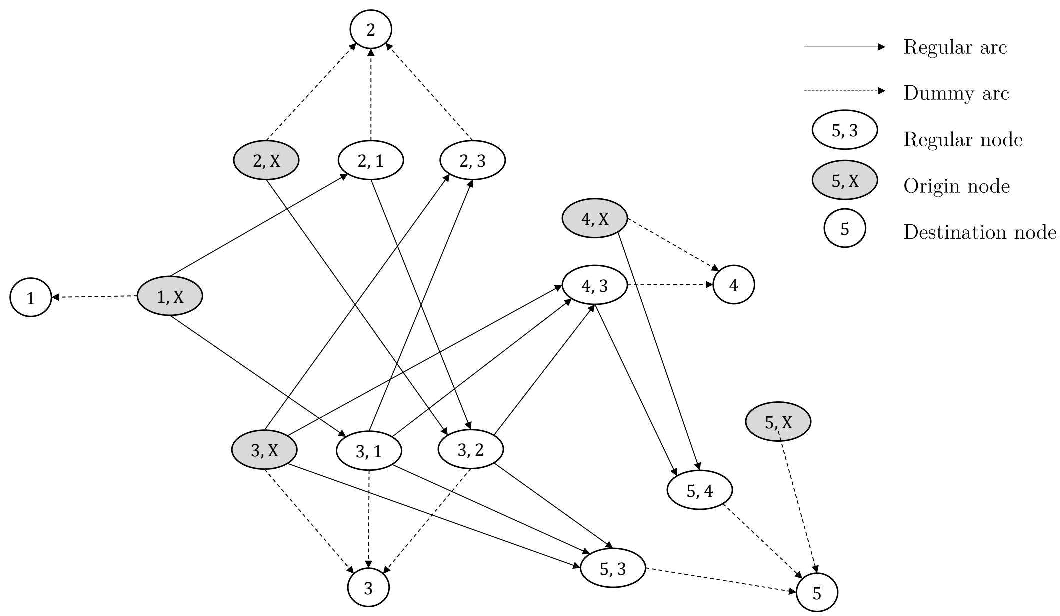


Figure 2.7: Network transformation to restrict cycling.

$(i, j) \in A$ (which we refer to as the parent arc) and if the first element of k equals the last element of l. For example, for the network in Figure 2.6, when $m = 1$, the network $\mathcal{G}$ (see Figure 2.7) contains arcs between nodes $(2, 1)$ and $(3, 2)$ since the node 3 in the original network can be reached from node 2 (in the original network) and the first element of (2,1) is the last element of (3,2). Let $\mathcal{A}_{ij} \subset \mathcal{A}$ represent the set of arcs in $\mathcal{A}$ which share the same parent arc $(i, j) \in A$.

Finally, dummy arcs are created in $\mathcal{G}$ to connect nodes in M_i and $i \in M$. These arcs are assumed to exist in a single state with zero free flow travel time and infinite capacity. The subset of nodes $M \in \mathcal{N}$ play the role of destinations and the nodes $\{(i, X, \ldots, X) : i \in N\}$ serve as origins. For instance, in Figure 2.6, if d_{15} users travel from node 1 to node 5, then the demand between $(1, X)$ and 5 in Figure 2.7 is set to d_{15}.

We suppose that the regular arcs exist in the same number of states as their parent arcs. However, the travel time on a regular arc is not solely a function of its flow but also depends on the flow on other arcs which share the same parent arc. More precisely, if (i, j) is the parent arc of (k, l), then $t^s_{kl}(\mathbf{x}_{\mathcal{A}}) = t^s_{ij}(\sum_{(\hat{k},\hat{l}) \in \mathcal{A}_{ij}} x^s_{\hat{k}\hat{l}})$, where $\mathbf{x}_{\mathcal{A}}$ represents the vector of state-dependent link flows in $\mathcal{G}$. For instance, the travel time on arc between $(2, X)$ and $(3, 2)$ in Figure 2.6 is a function of all travelers using arc $(2, 3)$ in the original network, which is obtained by adding the flow on arc between $(2, X)$ and $(3, 2)$ (which represents travelers starting at 2 and headed to 3) and $(2, 1)$ and $(3, 2)$ (which represents users traveling to 3 after reaching node 2 via node 1). While this construct violates the separability assumption of the arc costs, the link delay functions for arcs in $\mathcal{G}$ are symmetric, i.e., $\partial t^s_{k_1 l_1} / \partial x^s_{k_2 l_2} = \partial t^s_{k_2 l_2} / \partial x^s_{k_1 l_1} \, \forall \, (k_1, l_1), (k_2, l_2) \in \mathcal{A}$ and hence the FW method can be applied to compute the equilibrium solution and the optimal tolls (Vliet, 1987).

For the network in Figure 2.5, the 1-SOR problem can be used to eliminate cycling between nodes 2 and 3. The system optimal flows and tolls are shown in Figure 2.8. As expected, the flow and toll on link (3,2) is zero. However, note that there is a wide variation in the tolls when compared to the 0-SOR model.

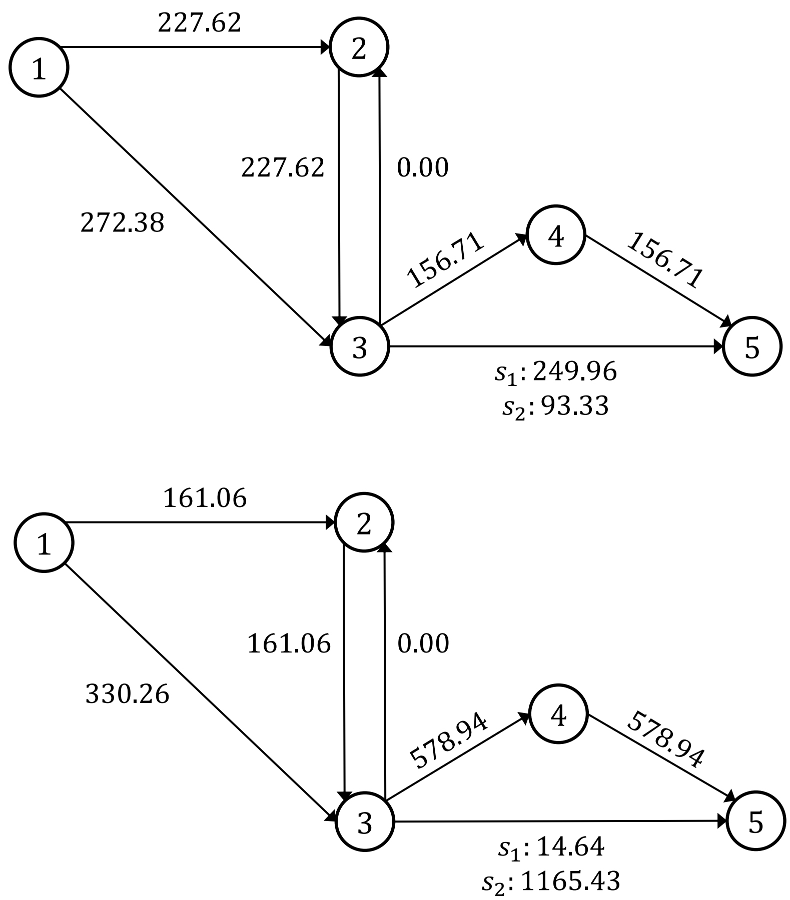


Figure 2.8: Optimal flows (top) and tolls (bottom) for the 1-SOR problem.

Table 2.1: Total expected travel time of UER and SOR solutions.

m	*No. of nodes*	*No. of arcs*	*UER*	*SOR*
0	24	76	8.6256E+06	8.3526E+06
1	125	378	8.7206E+06	8.4502E+06
2	379	1224	8.7211E+06	8.4502E+06
3	1237	3864	8.7213E+06	8.4502E+06

2.5 Demonstration

The FW algorithm for the SOR and UER problems was tested on the Sioux Falls network consisting of 24 nodes and 76 links (see Figure 2.9). Each link in the network was assumed to exist in two states: one corresponding to normal operating condition and another representing disrupted condition due to supply side uncertainty (which was modeled using a 50% reduction in capacity). The probabilities for these two link-states were set to 0.9 and 0.1 respectively for all the links in the network. The arc data for the normal conditions was obtained from `https://github.com/bstabler/TransportationNetworks` and the BPR function was used to estimate the state-dependent travel times using the state-dependent flows. The value of ϵ in the FW algorithm was set to 10^{-4}. Table 2.1 shows the TETT values for the SOR and UER variants. The TETT values for $m = 0$ and $m = 1$ are significantly different but the difference between the TETT of variants with larger m is minimal. Since the Sioux Falls network does not have many cycles of length exactly 3, the TETT values for $m = 2$ are close to that for $m = 1$ as expected. However, the results for $m = 3$ indicate that many of the cyclic policies used by travelers in the 0-UER and 0-SOR assignment have cycles only of length 2.

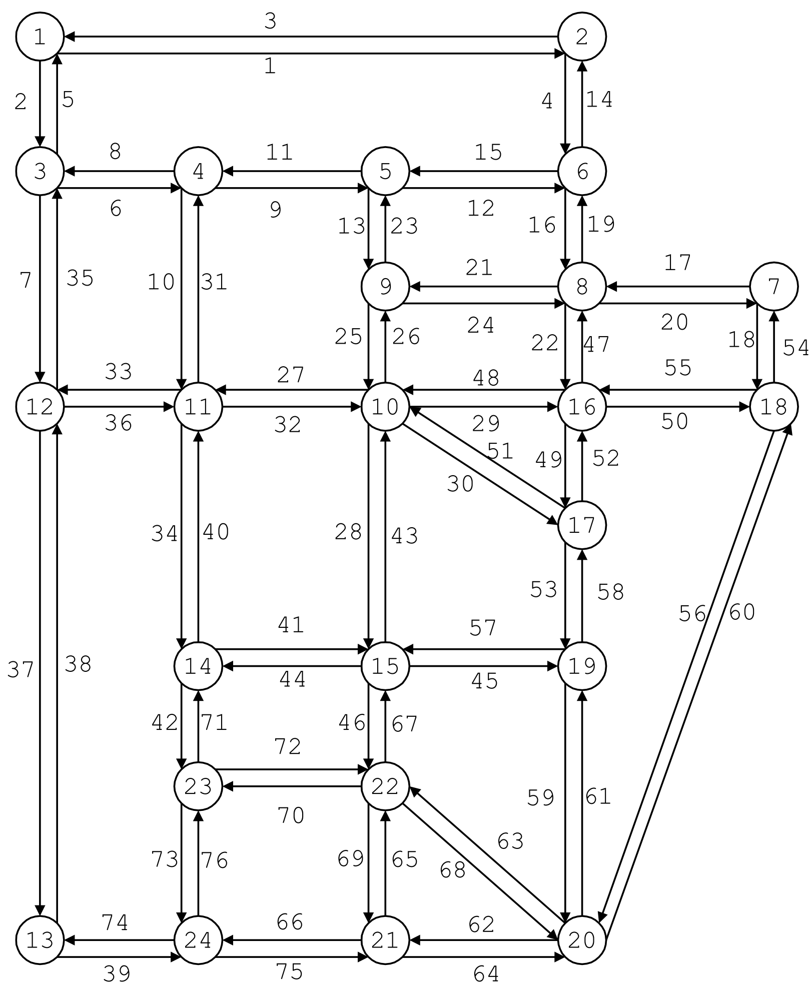


Figure 2.9: Sioux Falls network.

A comparison of the marginal tolls for different values of m is presented in Table 2.2. The results reinforce the previous observation that restricting cycles of length 2 has a noticeable effect on the equilibrium solution and optimal tolls. However, optimal tolls for instances which disallows cycles of length less than or equal to 3 or 4 are nearly the same as those for instances in which cycles of length 2 are prohibited.

Table 2.2: Comparison of marginal tolls.

	0-SOR vs. 1-SOR	1-SOR vs. 2-SOR	2-SOR vs. 3-SOR
RMS error	9.066	0.068	0.066
Maximum error	2.805	0.395	0.389
Minimum error	-48.500	-0.199	-0.244

Figure 2.10 depicts the run-times in seconds for the four SOR problems. The FW algorithm was implemented in C++ on a Linux machine with an Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz processor, 16 GB RAM, and 6 MB cache. A deque data structure for the TD-OSP scan eligible list in which nodes are removed from the front and added to its back was found improve the run-times. Matrix inversion in the all-or-nothing assignment was performed using the Eigen library. As the 1-SOR, 2-SOR, and 3-SOR problems are defined on a transformed network with more number of arcs they take a longer time to converge.

When roadway capacities are stochastic, one may estimate the expected capacity and calculate a static marginal toll using traditional traffic assignment models. However, when travelers select links adaptively, charging static marginal

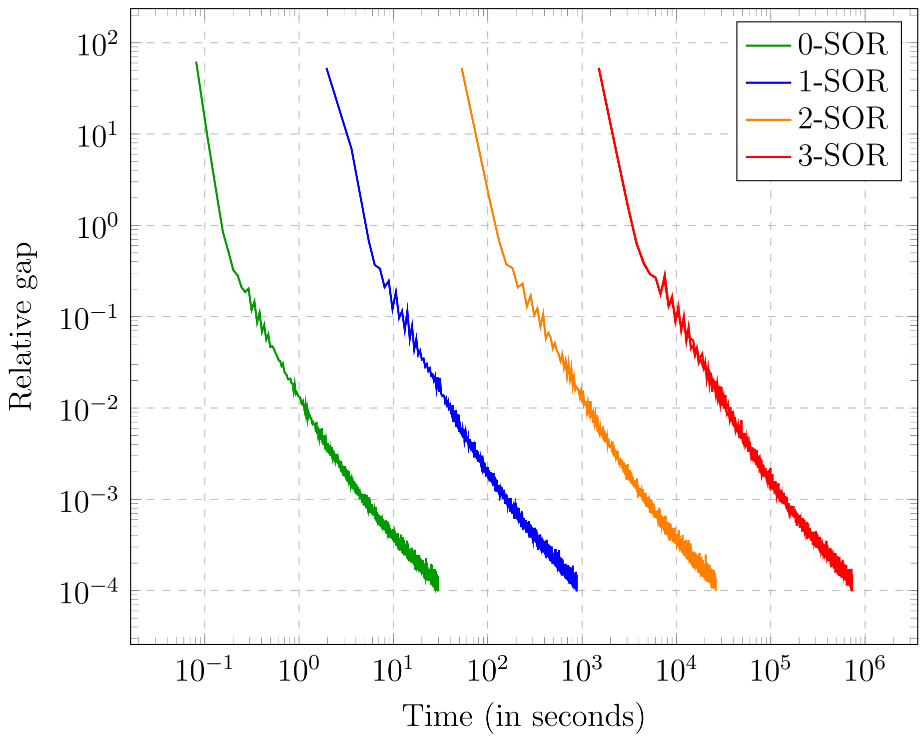


Figure 2.10: Computational performance of the FW method for different SOR variants.

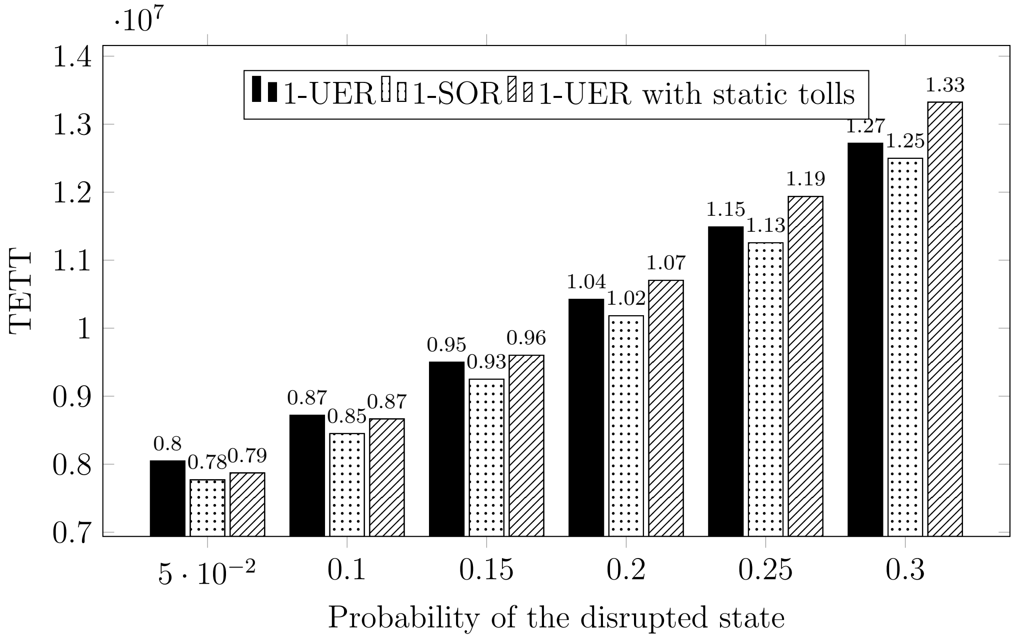


Figure 2.11: Impact of static and state-dependent marginal tolls for different disruption probabilities.

tolls will lead to suboptimal system performance. In fact, in some cases, it may also lead to an increase in the TETT when compared to the no-tolls (UER) or the do-nothing scenario. To highlight the advantage of state-dependent marginal tolls over static tolls we compared the TETT values of 1-UER, 1-SOR, and 1-UER with static marginal toll models in Figures 2.11 and 2.12. In addition, sensitivity to input parameters such as the probability and severity of disruption were also studied. The 1-UER model with static marginal tolls assumes that users adapt to *en route* link state information but the toll is calculated using a traditional system optimum model with expected link capacities.

Figure 2.11 shows the TETT values for the three models for different link disruption probabilities. The probabilities of disruption, which represents the fraction of time links are disrupted, are plotted on the x-axis and the TETT values (scaled down by a factor of 10^7) are represented on the y-axis. As before, the link capacities in the disrupted state were reduced by 50%. The results indicate that when disruption probabilities are low (i.e., when disruptions occur for small durations), the static marginal tolls result in a state with lower TETT than the no-tolls case but are still suboptimal when compared to the SOR tolls. However, as the disruption probabilities increased, static marginal tolls led to more congestion than the UER state. For instance, when the links are disrupted for 30% of the time, using static tolls results in a TETT of 13.3 million vehicle-minutes, which is nearly 5% higher than the TETT of the no-tolls case (12.7 million vehicle-minutes).

Sensitivity analysis with respect to severity of disruptions revealed a similar trend in the TETT values (see Figure 2.11). The SOR and UER models were tested on three instances in which the disrupted link capacities was set to 25% (Low), 50% (Med), and 75% (High) of the normal operating capacity. The probability of finding a link in a disrupted state was fixed at 0.1. As the severity of disruption increased, the performance of the UER model with static marginal tolls worsened when compared with the UER state.

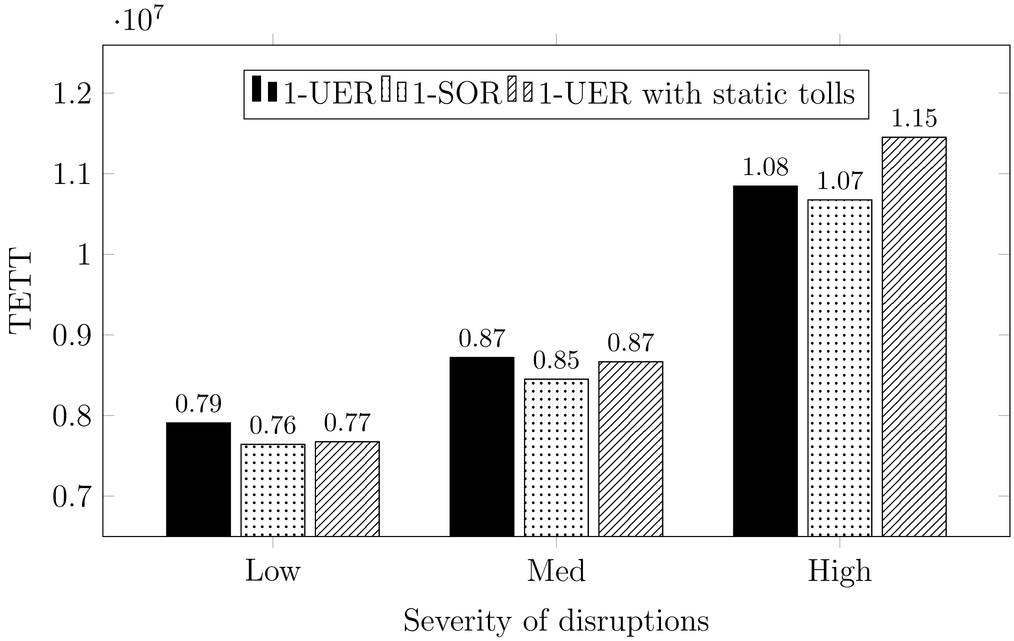


Figure 2.12: Impact of static and state-dependent marginal tolls for different levels of disruption severity.

2.6 Summary

In this chapter, a congestion pricing model was proposed for networks in which links exist in different states (representing non-recurring events such as poor weather, incidents, and temporary bottlenecks) with probabilities that are *exogenous* and independent of the flow variables. Traveler do not simply choose paths but select policies which respond to *en route* information. Both the user equilibrium and system optimum versions of this problem were defined and it was shown that marginal cost pricing (with a different price for each link-state) can align the UER and SOR solutions.

The optimal policies used by travelers at equilibrium are known to contain cycles due to the reset assumption of link-states. In order to correct this modeling artifact, a network transformation is proposed that restricts cycling in the optimal policies. The intuition behind this comes from the fact that the SOR and UER problems on acyclic graphs do not involve cyclic policies. Our network transformation eliminates cycles of certain lengths and thus makes the underlying graph relatively "acyclic". The proposed methods were demonstrated using the Sioux Falls network and the results indicate that problem of determining the optimal marginal tolls is computationally tractable even when cycles of certain lengths are avoided. In the next chapter, we formulate the minimum revenue congestion pricing problem for system optimum with recourse which may make the state-dependent pricing model more appealing by potentially reducing or eliminating tolls, especially on disrupted links.

Chapter 3

Within-Day Pricing: Minimum Expected Revenue Tolls

3.1 Introduction

In the static traffic assignment models discussed in Chapter 1, marginal tolls were shown to lead to a SO state. However, the set of tolls that improve system efficiency are not necessarily unique. The same is true for state-dependent tolls in the SOR model. For instance, consider the network that was used in the previous chapter to illustrate techniques for restricting cycling (see Figure 3.1). The top panel shows the optimal marginal tolls which generate a revenue of 393906.40 units whereas the bottom panel shows another set of state-dependent tolls that also lead to an SOR state while collecting only a total of 8266.93 units from the travelers in the network.

Minimum revenue congestion pricing models for deterministic TAPs (Bergendorff et al., 1997; Dial, 1999a, 2000; Penchina, 2004) were developed to make the concept of tolling more acceptable to the public. In a few cases, these minimum revenue tolls were shown to be equitable because they ensure that at least one of the paths between each OD pair remains untolled. As noted in Chapter 1, a majority of governments and the public are averse to the idea of tolling. This is because tolls and fuel taxes are perceivable out-of-pocket costs unlike the societal benefits of reduced network congestion. Although tolls are considered to be transfer payments, in which case the revenue generated does not matter, having a tolling scheme that achieves a social optimum state while collecting less money from travelers makes it an appealing network improvement strategy.

In deterministic TAPs, the KKT conditions for the Beckmann formulation are both necessary and sufficient. Using the SO link flow solution, these conditions may be reduced to a set of linear inequalities involving the tolls and the Lagrangian multipliers associated with the flow conservation constraints. Thus, one can formulate a linear program with the revenue (i.e., flow on each link times the toll) as the objective subject to linear equilibrium constraints. In the same spirit, this chapter explores the problem of determining state-dependent minimum expected revenue tolls under supply-side uncertainty. Throughout this chapter, for the purpose of brevity, we will refer to the objective simply as revenue instead of expected revenue.

The only caveat is that the user equilibrium and system optimum model with recourse discussed in the previous chapter involve link-state and policy flow variables. While this formulation uses the least amount of memory, link flows are aggregated over all OD pairs and hence cannot be directly used in constructing the constraint set of the minimum revenue pricing problem. Motivated by this fact, we propose a reformulation of the UER/SOR problem

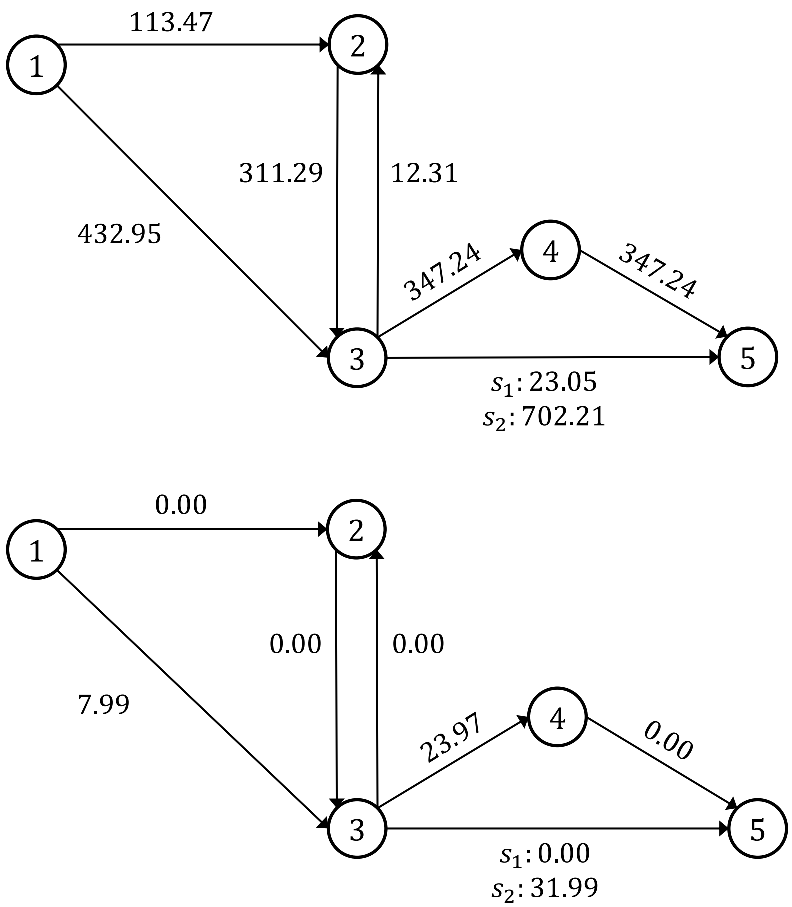


Figure 3.1: Marginal tolls (top) and minimum revenue tolls (bottom) for the 0-SOR problem.

in this chapter that will help derive accurate solutions without using policy flow variables. This formulation also has the potential to improve convergence for large problems as they can eliminate residual flows unlike the link-based methods.

Throughout this chapter, we will discuss methods to solve the 0-SOR and 0-UER models. Extending the proposed methods to other recourse variants remains beyond the scope of this dissertation. The rest of this chapter is organized as follows. In Section 3.2, we propose an alternate UER formulation, derive the first order optimality conditions, and establish equivalence between the first order conditions and the UER principle. We then propose a solution method that tracks the split proportions at each node-state in Section 3.3. Section 3.4 discusses two linear programming formulations for computing the minimum revenue tolls that are based on the UER reformulation and solution methods discussed in Sections 3.2 and 3.3. Algorithms for the UER reformulation and linear programs for minimum revenue are tested on the Sioux Falls network in Section 3.5. Finally in Section 3.6, we summarize the findings of this chapter.

3.2 Multiple Origin, Single Destination Problem

Consider a network $G = (N, A)$ with a single destination v but multiple origins belonging to the set Z. The methods discussed in this section can be repeated for different destinations iteratively to solve the multiple OD pair problem. Let the demand from origin $u \in Z$ to v be d_{uv}. Let, as before, the set of messages at a node i be denoted by Θ_i and q_i^θ be the probability of observing message θ. Assume that z_{ij}^θ is the number of travelers who observe message $\theta \in \Theta_i$ at node i and take arc (i, j). The link-state of arc (i, j) is denoted using θ_{ij}. Notice that $x_{ij}^s = \sum_{\theta:\theta_{ij}=s} z_{ij}^\theta$. Suppose $\mathbf{z} = (z_{ij}^\theta)_{(i,j)\in A,\theta\in\Theta_i}$ and $\mathbf{x} = (x_{ij}^s)_{(i,j)\in A,s\in S_{ij}}$. Also suppose that $\sum_{(i,j)\in A} |S_{ij}|$ and $\sum_{(i,j)\in A} |\Theta_i|$ are denoted as $|S|$ and $|M|$, respectively. Then, the UER problem can also be formulated as

$$\begin{aligned}
\min \quad & f(\mathbf{z}) = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int_0^{\sum_{\theta:\theta_{ij}=s} z_{ij}^\theta} t_{ij}^s(z)dz && (3.1)\\
\text{s.t.} \quad & q_i^\theta \left(\eta_i + \sum_{(h,i)\in A} \sum_{\bar{\theta}\in\Theta_h} z_{hi}^{\bar{\theta}} \right) = \sum_{(i,j)\in A} z_{ij}^\theta \qquad i \in N, \theta \in \Theta_i && (3.2)\\
& z_{ij}^\theta \geq 0 \qquad \forall (i,j) \in A, \theta \in \Theta_i && (3.3)
\end{aligned}$$

where η_i is the demand entering/leaving the node i and is defined as

$$\eta_i = \begin{cases} d_{uv} & \text{if } i = u \\ -\sum_{u\in Z} d_{uv} & \text{if } i = v \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

In order to formally establish an equivalence between the above formulation and the UER conditions, we first derive the KKT conditions of the above non-linear program. Lagrangianizing the flow conservation constraints,

$$L(\mathbf{z}) = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int\limits_0^{\sum_{\theta:\theta_{ij}=s} z_{ij}^\theta} t_{ij}^s(z)dz + \lambda_i^\theta \left(\sum_{(i,j)\in A} z_{ij}^\theta - \sum_{(h,i)\in A} \sum_{\bar{\theta}\in\Theta_h} q_i^\theta z_{hi}^{\bar{\theta}} - q_i^\theta \eta_i \right) \tag{3.5}$$

Denoting $t_{ij}^{\theta_{ij}}\left(\sum_{\theta:\theta_{ij}=s} z_{ij}^{\theta}\right)$ as t_{ij}^{θ}, the KKT conditions for all $(i,j) \in A, \theta \in \Theta_i$ are

$$\frac{\partial L(\mathbf{z})}{\partial z_{ij}^{\theta}} \geq 0 \Rightarrow t_{ij}^{\theta} + \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta}} \lambda_j^{\bar{\theta}} - \lambda_i^{\theta} \geq 0 \tag{3.6}$$

$$z_{ij}^{\theta}\left(\frac{\partial L(\mathbf{z})}{\partial z_{ij}^{\theta}}\right) = 0 \Rightarrow z_{ij}^{\theta}\left(t_{ij}^{\theta} + \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta}} \lambda_j^{\bar{\theta}} - \lambda_i^{\theta}\right) = 0 \tag{3.7}$$

Notice that the above conditions hold even if a fixed real number is added to some of the λ values. Hence, we set

$$\lambda_v^{\theta} = 0 \qquad \forall\, \theta \in \Theta_v \tag{3.8}$$

Equations (3.6)–(3.8) are first order necessary conditions, i.e., for any optimal solution to the non-linear program defined by equations (3.1)–(3.3), there exists a vector of unrestricted λs that satisfy equations (3.6)–(3.8).

The following proposition proves that the optimal solution to this formulation can be used to construct policies that satisfy the UER conditions (i.e., at equilibrium, all used policies have equal and minimal expected costs). Suppose that a policy π carries flow from origin u to destination v. If a node i is accessible from u, i.e., if the probability of reaching i from node u is positive, which we denote using $u \to i$, then $\pi(i,\theta) = j \Rightarrow \bar{z}_{ij}^{\theta} > 0$, where $\bar{z}_{ij}^{\theta}$ is a solution to the non-linear program (3.1)–(3.3). In such cases, we say that the policy π is *used* by travelers from origin u. Any policy that does not carry demand from origin u is said to be *unused* by travelers starting at u. Throughout this chapter we assume that all policies terminate at the destination with probability 1.

Proposition 3.1. *Every used policy constructed from $\bar{z}_{ij}^{\theta}$ has the same expected cost from all origins. All unused policies are at least as expensive as the used policies.*

Proof. Let u be an arbitrary origin. Suppose a policy π is used by travelers starting at u. The value function (distance labels) associated with states (i, θ), denoted by $\mu_i^{\theta}(\pi)$, can be estimated uniquely by solving the following system of equations.

$$\mu_i^{\theta}(\pi) = t_{i,\pi(i,\theta)}^{\theta} + \sum_{\bar{\theta} \in \Theta_{\pi(i,\theta)}} q_{\pi(i,\theta)}^{\bar{\theta}} \mu_{\pi(i,\theta)}^{\bar{\theta}}(\pi) \ \forall \ i \in N \backslash \{v\}, \theta \in \Theta_i \tag{3.9}$$

$$\mu_v^{\theta}(\pi) = 0 \qquad \forall \ \theta \in \Theta_v \tag{3.10}$$

In order to compute the labels associated with the origin $\mu_u^{\theta}(\pi)$, it suffices to solve a subset of the above equations. More specifically, we can ignore the nodes that cannot be reached from u with positive probability, while using policy π, as shown in the following equations.

$$\mu_i^{\theta}(\pi) = t_{i,\pi(i,\theta)}^{\theta} + \sum_{\bar{\theta} \in \Theta_{\pi(i,\theta)}} q_{\pi(i,\theta)}^{\bar{\theta}} \mu_{\pi(i,\theta)}^{\bar{\theta}}(\pi) \ \forall \ i \in N \backslash \{v\} : u \to i, \theta \in \Theta_i \tag{3.11}$$

$$\mu_v^{\theta}(\pi) = 0 \qquad \forall \ \theta \in \Theta_v \tag{3.12}$$

Since policy π is used by travelers from u, $\bar{z}_{i\pi(i,\theta)}^{\theta} > 0 \, \forall \, i \in N \backslash \{v\} : u \to i, \theta \in \Theta_i$. Notice that equations (3.11) and (3.12) are a subset of the KKT conditions (3.6)–(3.8). Hence, for any vector of λs satisfying the KKT conditions, $\lambda_u^{\theta} = \mu_u^{\theta}(\pi) \, \forall \, \theta \in \Theta_u$. Therefore, by defining $\lambda_u = \sum_{\theta \in \Theta_u} q^{\theta} \lambda_u^{\theta}$, we conclude that all used policies have the same expected cost from u.

Now consider a policy π that is unused by travelers from u. The expected cost to reach v from u can be obtained by solving equations (3.11) and (3.12) as before. However, we cannot conclude that $z^{\theta}_{i\pi(\theta)} = 0$ for any $i \in N\backslash\{v\}, \theta \in \Theta_i$ and simplify the KKT conditions since travelers using other policies may choose $\pi(i,\theta)$ when at state (i,θ). From equation (3.7),

$$t^{\theta}_{i,\pi(i,\theta)} + \sum_{\bar{\theta}\in\Theta_{\pi(i,\theta)}} q^{\bar{\theta}}_{j}\lambda^{\bar{\theta}}_{\pi(i,\theta)} - \lambda^{\theta}_{\pi(i,\theta)} \geq 0 \,\forall\, i \in N\backslash\{v\} : u \to i, \theta \in \Theta_i \tag{3.13}$$

Using equations (3.11) and (3.13), for all $i \in N\backslash\{v\} : u \to i, \theta \in \Theta_i$, we may write

$$\lambda^{\theta}_{i} - \mu^{\theta}_{i}(\pi) \leq \sum_{\bar{\theta}\in\Theta_{\pi(i,\theta)}} q^{\bar{\theta}}_{\pi(i,\theta)}\left(\lambda^{\bar{\theta}}_{\pi(i,\theta)} - \mu^{\bar{\theta}}_{\pi(i,\theta)}(\pi)\right) \tag{3.14}$$

which in matrix form can be represented as $(\mathbf{I} - \mathbf{Q}_{\pi})(\boldsymbol{\lambda} - \boldsymbol{\mu}) \preceq \mathbf{0}$, where $\mathbf{Q}_{\pi}$ represents a transition probability matrix and $\preceq$ denotes a component-wise inequality. We proceed by left-multiplying both sides with $(\mathbf{I} - \mathbf{Q}_{\pi})^{-1}$. However, one must make sure that this preserves the component-wise inequality. Since $\mathbf{Q}_{\pi}$ is a transition probability matrix, all of its eigenvalues have a magnitude less than 1 which implies that

$$(\mathbf{I} - \mathbf{Q}_{\pi})^{-1} = \mathbf{I} + \mathbf{Q}_{\pi} + \mathbf{Q}^{2}_{\pi} + \ldots \tag{3.15}$$

Since each of the elements of $\mathbf{Q}_{\pi}$ is non-negative, it follows from the above equation that the all the elements of $(\mathbf{I} - \mathbf{Q}_{\pi})^{-1}$ are non-negative. Hence,

$$(\mathbf{I} - \mathbf{Q}_{\pi})(\boldsymbol{\lambda} - \boldsymbol{\mu}) \preceq \mathbf{0} \tag{3.16}$$

$$\Rightarrow (\mathbf{I}-\mathbf{Q}_\pi)^{-1}(\mathbf{I}-\mathbf{Q}_\pi)(\boldsymbol{\lambda}-\boldsymbol{\mu}) \preceq (\mathbf{I}-\mathbf{Q}_\pi)^{-1}\mathbf{0} \tag{3.17}$$

$$\Rightarrow \boldsymbol{\lambda}-\boldsymbol{\mu} \preceq \mathbf{0} \tag{3.18}$$

$$\Rightarrow \boldsymbol{\lambda} \preceq \boldsymbol{\mu} \tag{3.19}$$

Thus, $\mu_u^\theta(\pi) \geq \lambda_u^\theta \, \forall \, \theta \in \Theta_u \Rightarrow \mu_u(\pi) \geq \lambda_u$. Hence, the cost of all unused policies are at least λ_u. ■

3.3 A Solution Method using Split Proportions

The previous section established that a solution to the new formulation satisfies the UER principle. In this section, we describe a method similar to origin-based assignment (OBA) (Bar-Gera, 2002) to compute the optimal z values. Let α_{ij}^θ represent the proportion of travelers arriving at node i in node-state (i,θ) and choosing link (i,j). Denote $\boldsymbol{\alpha} = (\alpha_{ij}^\theta)_{(i,j)\in A, \theta\in\Theta_i}$. Thus, the following equations hold:

$$q_i^\theta \alpha_{ij}^\theta \left(\eta_i + \sum_{(h,i)\in A} \sum_{\bar{\theta}\in\Theta_h} z_{hi}^{\bar{\theta}} \right) = z_{ij}^\theta \qquad \forall (i,j) \in A, \theta \in \Theta_i \tag{3.20}$$

which in matrix form can be expressed as $(\mathbf{I}-\mathbf{Q}_{\boldsymbol{\alpha}}^T)\mathbf{z} = \mathbf{a}^{\boldsymbol{\alpha}}$, where $\mathbf{Q}_{\boldsymbol{\alpha}}^T \in \mathbb{R}_+^{|M|\times|M|}$ and $\mathbf{a}^{\boldsymbol{\alpha}} \in \mathbb{R}_+^{|M|\times 1}$. The flow conservation equations can also be written in terms of the split proportions $\boldsymbol{\alpha}$ as follows:

$$\sum_{(i,j)\in A} \alpha_{ij}^\theta = 1 \qquad \forall i \in N \backslash \{v\}, \theta \in \Theta_i \tag{3.21}$$

$$\alpha_{vj}^\theta = 0 \qquad \forall j \in \Gamma(v), \theta \in \Theta_v \tag{3.22}$$

$$\alpha_{ij}^{\theta} \geq 0 \qquad \forall (i,j) \in A, \theta \in \Theta_i \tag{3.23}$$

Further, since $x_{ij}^s = \sum_{\theta \in \Theta_i : \theta_{ij} = s} z_{ij}^{\theta}$, the following equations are satisfied.

$$x_{ij}^s = \sum_{\substack{\theta \in \Theta_i: \\ \theta_{ij}=s}} q_i^{\theta} \alpha_{ij}^{\theta} \left(\eta_i + \sum_{(h,i) \in A} \sum_{\bar{s} \in S_{hi}} x_{hi}^{\bar{s}} \right) \forall\, (i,j) \in A, s \in S_{ij} \tag{3.24}$$

$$\Rightarrow x_{ij}^s - \sum_{\substack{\theta \in \Theta_i: \\ \theta_{ij}=s}} \sum_{(h,i) \in A} \sum_{\bar{s} \in S_{hi}} q_i^{\theta} \alpha_{ij}^{\theta} x_{hi}^{\bar{s}} = \sum_{\substack{\theta \in \Theta_i: \\ \theta_{ij}=s}} \sum_{(h,i) \in A} \sum_{\bar{s} \in S_{hi}} \eta_i q_i^{\theta} \alpha_{ij}^{\theta} \,\forall\, (i,j) \in A, s \in S_{ij} \tag{3.25}$$

These equations can be compactly written in matrix form as $\left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^T\right) \mathbf{x} = \mathbf{b}^{\boldsymbol{\alpha}}$, where $\mathbf{P}_{\boldsymbol{\alpha}}^T \in \mathbb{R}_+^{|S| \times |S|}$ and $\mathbf{b}^{\boldsymbol{\alpha}} \in \mathbb{R}_+^{|S| \times 1}$. Thus, given $\boldsymbol{\alpha}$, one can use equations (3.20) and (3.24) to compute $\mathbf{z}$ and $\mathbf{x}$ respectively. (Since $\mathbf{x}$ and $\mathbf{z}$ are functions of $\boldsymbol{\alpha}$, we could denote them as $\mathbf{x}_{\boldsymbol{\alpha}}$ and $\mathbf{z}_{\alpha}$, but we ignore the subscripts to keep the notation simple.)

The solution to equations (3.20) and (3.24) is unique as one can interpret $\boldsymbol{\alpha}$ to represent a stochastic policy, i.e., at every node state, each of the downstream arcs is chosen with a certain probability. Thus, $\mathbf{Q}_{\boldsymbol{\alpha}}$ and $\mathbf{P}_{\boldsymbol{\alpha}}$ can be viewed as transition probability matrices of an absorbing Markov chain. Unlike the UER formulation in the earlier chapter, in which different number of travelers used different policies, every traveler in the network is now assumed to follow the same stochastic policy. Since travelers are non-atomic, the probabilities of choosing the downstream links at each node state coincide with the split proportions.

Treating the split proportions $\boldsymbol{\alpha}$ as a vector of decision variables, we now focus on obtaining a descent direction to the optimization model described earlier. This would let us use a gradient projection-like algorithm to solve the problem. The elements of the Jacobian $\nabla_{\boldsymbol{\alpha}} f$ can be computed as follows

$$\frac{\partial f}{\partial \alpha_{ij}^{\theta}} = \sum_{(k,l)\in A} \sum_{s\in S_{kl}} \frac{\partial f}{\partial x_{kl}^{s}} \frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} \tag{3.26}$$

$$= \sum_{(k,l)\in A} \sum_{s\in S_{kl}} t_{kl}^{s} \frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} \tag{3.27}$$

To estimate $\partial x_{kl}^{s}/\partial \alpha_{ij}^{\theta}$, we partially differentiate equations (3.24) with respect to α_{ij}^{θ}. The following cases arise:

Case 1:$(k,l) \neq (i,j)$ or $(k,l) = (i,j), s \neq \theta_{ij}$

$$\frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} = \sum_{\bar{\theta}:\bar{\theta}_{kl}=s} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} q_{k}^{\bar{\theta}} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} \tag{3.28}$$

Case 2:$(k,l) = (i,j), s = \theta_{ij}$

$$\frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} = q_{k}^{\theta}\eta_{k} + \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} \frac{\partial}{\partial \alpha_{ij}^{\theta}} \left(q_{k}^{\theta} \alpha_{kl}^{\theta} x_{hk}^{\bar{s}}\right) + \sum_{\substack{\bar{\theta}:\bar{\theta}\neq\theta \\ \bar{\theta}_{kl}=s}} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} q_{k}^{\bar{\theta}} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} \tag{3.29}$$

$$= q_{k}^{\theta}\eta_{k} + \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} \left(q_{k}^{\theta} x_{hk}^{\bar{s}} + q_{k}^{\theta} \alpha_{kl}^{\theta} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}}\right) + \sum_{\substack{\bar{\theta}:\bar{\theta}\neq\theta \\ \bar{\theta}_{kl}=s}} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} \tag{3.30}$$

$$= q_{k}^{\theta} \left(\eta_{k} + \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} x_{hk}^{\bar{s}}\right) + \sum_{\bar{\theta}:\bar{\theta}_{kl}=s} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} q_{k}^{\theta} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} \tag{3.31}$$

Thus, equations (3.28) and (3.31) form a system of equations in $\partial x_{kl}^{s}/\partial \alpha_{ij}^{\theta}$ as shown below.

$$\frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} - \sum_{\bar{\theta}:\bar{\theta}_{kl}=s} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} q_k^{\bar{\theta}} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} = 0 \,\forall\, (k,l) \neq (i,j) or (k,l) = (i,j), s \neq \theta_{ij} \tag{3.32}$$

$$\frac{\partial x_{kl}^{s}}{\partial \alpha_{ij}^{\theta}} - \sum_{\bar{\theta}:\bar{\theta}_{kl}=s} \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} q_k^{\theta} \alpha_{kl}^{\bar{\theta}} \frac{\partial x_{hk}^{\bar{s}}}{\partial \alpha_{ij}^{\theta}} = q_k^{\theta} \left(\eta_k + \sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} x_{hk}^{\bar{s}} \right) \forall\, (k,l) = (i,j), s = \theta_{ij} \tag{3.33}$$

From the above equations and equation (3.25), one may notice that the coefficient matrix of the unknown partials is the same, i.e., $\left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^{T}\right)$. In fact, the coefficient matrix is the same for every arc (i,j) and message θ. The only thing that changes for a different α_{ij}^{θ} is the right hand side. Thus, the partial derivatives of the link-state flows with respect to α_{ij}^{θ} can be written as $\left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^{T}\right)^{-1} \mathbf{b}^{\boldsymbol{\alpha}}\left((i,j),\theta\right)$, where $\mathbf{b}^{\boldsymbol{\alpha}}\left((i,j),\theta\right) \in \mathbb{R}_{+}^{|S|\times 1}$ and is defined as

$$b_{kl}^{s\boldsymbol{\alpha}}((i,j),\theta) = \begin{cases} q_k^{\theta} \left(\eta_k + \displaystyle\sum_{(h,k)\in A} \sum_{\bar{s}\in S_{hk}} x_{hk}^{\bar{s}} \right) & \text{if } (k,l) = (i,j), s = \theta_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{3.34}$$

Let $\mathcal{C}_{ij}^{s\boldsymbol{\alpha}}$ be the expected travel time to reach the destination for a traveler following the stochastic policy $\boldsymbol{\alpha}$ and starting at the upstream end of link (i,j) in state s. Then, the policy costs are related to the link costs by the equation $\boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}} = \mathbf{t}^{\boldsymbol{\alpha}} + \mathbf{P}_{\boldsymbol{\alpha}} \boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}}$, where $\boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}} \in \mathbb{R}_{+}^{|S|\times 1}$ and $\mathbf{t} \in \mathbb{R}_{+}^{|S|\times 1}$.

Thus, we can rewrite equation (3.27) as

$$\frac{\partial f}{\partial \alpha_{ij}^{\theta}} = \left(\mathbf{t}^{\boldsymbol{\alpha}}\right)^{T} \left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^{T}\right)^{-1} \mathbf{b}^{\boldsymbol{\alpha}}\left((i,j),\theta\right) \tag{3.35}$$

$$= \left(\left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}} \right) \boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}} \right)^T \left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^T \right)^{-1} \mathbf{b}^{\boldsymbol{\alpha}} \left((i,j), \theta \right) \tag{3.36}$$

$$= \left(\boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}} \right)^T \left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}} \right)^T \left(\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^T \right)^{-1} \mathbf{b}^{\boldsymbol{\alpha}} \left((i,j), \theta \right) \tag{3.37}$$

$$= \left(\boldsymbol{\mathcal{C}}^{\boldsymbol{\alpha}} \right)^T \mathbf{b}^{\boldsymbol{\alpha}} \left((i,j), \theta \right) \tag{3.38}$$

$$= \mathcal{C}_{ij}^{\theta_{ij}\boldsymbol{\alpha}} q_i^{\theta} \left(\eta_i + \sum_{(h,i) \in A} \sum_{\bar{s} \in S_{hi}} x_{hi}^{\bar{s}} \right) \tag{3.39}$$

We will henceforth refer to $\mathcal{C}_{ij}^{\theta_{ij}\boldsymbol{\alpha}}$ simply as $\mathcal{C}_{ij}^{\theta\boldsymbol{\alpha}}$ as long as it is clear from the context. Once the descent direction is known, we can choose an appropriate step size and project the new $\boldsymbol{\alpha}$ onto the feasible region described using equations (3.21) and (3.23). To simplify the projection part we can define basic and non-basic split proportions analogous to those in standard gradient projection algorithms. Consider a node-message pair (i, θ), where $i \neq v$. Let β_i^{θ} represent the head node of an arc adjacent to node i. We will refer to (i, β_i^{θ}) as the *basic arc* at (i, θ). All the remaining arcs will be referred to as *non-basic arcs* at state (i, θ). Let the vector of all non-basic arc-message proportions be referred to as $\boldsymbol{\alpha}(\text{NB})$. Then, the flow conservation constraints (3.21) and (3.23) can be expressed as

$$\alpha_{i\beta_i^{\theta}}^{\theta} = 1 - \sum_{(i,j) \in A, j \neq \beta_i^{\theta}} \alpha_{ij}^{\theta}(\text{NB}) \qquad \forall i \in N, \theta \in \Theta_i \tag{3.40}$$

$$\alpha_{vj}^{\theta} = 0 \qquad \forall j \in \Gamma(v), \theta \in \Theta_v \tag{3.41}$$

$$\alpha_{ij}^{\theta}(\text{NB}) \geq 0 \qquad \forall (i,j) \in A, \theta \in \Theta_i \tag{3.42}$$

For $j \neq \beta_i^{\theta}$ and $i \neq v$, we can compute the derivatives of the Beckmann objective function with respected to the non-basic alphas using the following equations:

$$\frac{\partial f}{\partial \alpha_{ij}^{\theta}(\mathrm{NB})} = \frac{\partial f}{\partial \alpha_{ij}^{\theta}} - \frac{\partial f}{\partial \alpha_{i\beta_i^{\theta}}^{\theta}} \tag{3.43}$$

$$= (\mathcal{C}_{ij}^{\theta\boldsymbol{\alpha}} - \mathcal{C}_{i\beta_i^{\theta}}^{\theta\boldsymbol{\alpha}}) q_i^{\theta} \left(\eta_i + \sum_{(h,i)\in A} \sum_{\bar{s}\in S_{hi}} x_{hi}^{\bar{s}} \right) \tag{3.44}$$

Remark. The expression for the gradient of the objective with respect to the non-basic split proportions is reminiscent of mixed strategy equilibria in game theory. In mixed strategy equilibria, all strategies with positive weight have equal expected utility. The same is true with split proportions in that all downstream arcs that are chosen with positive probabilities must have equal $\mathcal{C}$ values (assuming that the basic arc at each node-state belongs to the least cost policy).

Algorithm 4 outlines the pseudocode for a destination-based assignment using the split proportions for a single destination v. In the first step, we initialize the travel times for all link-states to their free flow travel times and find the online shortest path. The optimal policy is then used to initialize basic arcs at each node-state. Subsequently, the alpha values of the basic arc at each node-state are set to 1. Finally the link-state flows are computed and the link costs are updated.

The algorithm then equilibrates flow by computing new online shortest paths, updating the basic arcs, and shifting the split proportions from non-basic arcs to basic arcs at each node-state. The split proportions are adjusted using the gradient of the objective as calculated in equation (3.44). However, the exact step size that reduces the objective is slightly difficult to obtain. We employ a method similar to the OBA algorithm proposed by Bar-Gera (2002) and compute the social pressure associated with a step size φ. The social pressure associated with updating $\boldsymbol{\alpha}$ to $\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$ is given by $-\Delta\mathbf{x} \cdot \mathbf{t}(\mathbf{x} + \Delta\mathbf{x})$, where $\mathbf{x}$ and $\mathbf{x} + \Delta\mathbf{x}$ are the link-state flows that correspond to the split proportions $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}$ respectively. We set φ to 1 and reduce it by half until the social pressure becomes positive. The details of updating the split proportions are shown in Algorithm 5.

The reason for computing the social pressure to update the split proportion is that the Beckmann objective is not convex in $\boldsymbol{\alpha}$. Therefore, the first order optimality conditions are necessary but not sufficient. Hence, we find the link-state flows after each split proportions update and verify if the Beckmann function has improved. Bar-Gera (2002) showed that, at every sub-optimal solution of the OBA, there exists a $\varphi > 0$ that results in positive social pressure, and we hypothesize that the same is true for the equilibrium with recourse problem.

For a given φ, if the inflow to a node in the network is positive, the non-basic split proportions are first reduced by an amount $\varphi \partial f / \partial \alpha_{ij}^{\theta}(\text{NB})$. However, if the reduced split proportion is negative, we project it back to the feasible region by setting it to zero. When the inflow is zero, the non-basic split proportions are set to zero. In this particular case, other update rules that depend on the

Algorithm 4 DBA $(G, \mathbf{t}, v)$

Step 1: Initialize Flows
$\boldsymbol{\alpha} \leftarrow \mathbf{0}$
GAP $\leftarrow \infty$
$\pi^* \leftarrow$ TD-OSP$(G, \mathbf{t}(\mathbf{0}), v)$
for $i \in N \backslash \{v\}, \theta \in \Theta_i$ **do**
  $\beta_i^\theta \leftarrow \pi^*(i, \theta)$
  $\alpha_{i\beta_i^\theta}^\theta \leftarrow 1$
end for
$\mathbf{x} \leftarrow (\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^T)^{-1} \mathbf{b}^{\boldsymbol{\alpha}}$

Step 2: Equilibrate Flows
while GAP $> \epsilon$ **do**
  $\pi^* \leftarrow$ TD-OSP$(G, \mathbf{t}(\mathbf{x}), v)$
  GAP $\leftarrow (\mathbf{t} \cdot \mathbf{x}) \left(\sum_{u \in Z} \sum_{v \in Z} d_{uv} C_u^{\pi^*}\right)^{-1} - 1$
  for $i \in N \backslash \{v\}, \theta \in \Theta_i$ **do**
    $\beta_i^\theta \leftarrow \pi^*(i, \theta)$
  end for
  Update the set of non-basic arcs-messages NB
  Compute $\nabla_{\boldsymbol{\alpha}} f$ and $\nabla_{\boldsymbol{\alpha}(\mathrm{NB})} f$ using equations (3.39) and (3.44)
  $\boldsymbol{\alpha} \leftarrow \textsc{UpdateAlpha}\left(G, v, \boldsymbol{\alpha}, \nabla_{\boldsymbol{\alpha}(\mathrm{NB})} f\right)$
end while

expected costs $\mathcal{C}$ such as those defined in Bar-Gera (2002) (see equation (36)) may also be used. For the numerical experiments discussed later, in order to speed up convergence, we sometimes update the split proportions at a single node-state instead of updating them at all node-states using a single value of φ, estimate the new link-state costs, and recompute the online shortest paths.

Algorithm 4 provides a solution to the multiple origin, single destination problem. To solve the UER and SOR problems for multiple destinations, one can simply keep track of the destination dependent split proportions and link-state

Algorithm 5 $\textsc{UpdateAlpha}\big(G, v, \boldsymbol{\alpha}, \nabla_{\boldsymbol{\alpha}(\mathrm{NB})} f\big)$

$\varphi \leftarrow 1$
$\hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}$
$\hat{\mathbf{x}} \leftarrow \mathbf{x}$
Social Pressure $\leftarrow -\infty$
while *Social Pressure* < 0 **do**
  $\boldsymbol{\alpha} \leftarrow \hat{\boldsymbol{\alpha}}$
  for $(i, j) \in A, \theta \in \Theta_i : i \neq v, j \neq \beta_i^\theta$ **do** ▷ Update non-basic variables
    if $(\eta_i + \sum_{(h,i) \in A} \sum_{\bar{s} \in S_{hi}} x_{hi}^{\bar{s}}) > 0$ **then**
      $\alpha_{ij}^\theta \leftarrow \max\big\{\alpha_{ij}^\theta - \varphi \partial f / \partial \alpha_{ij}^\theta(\mathrm{NB}), 0\big\}$
    else
      $\alpha_{ij}^\theta \leftarrow 0$
    end if
  end for

  for $i \in N \backslash \{v\}, \theta \in \Theta_i$ **do** ▷ Update basic variables
    $\alpha_{i\beta_i^\theta}^\theta \leftarrow 1 - \sum_{(i,j) \in A, j \neq \beta_i^\theta} \alpha_{ij}^\theta(\mathrm{NB})$
  end for
  $\mathbf{x} \leftarrow (\mathbf{I} - \mathbf{P}_{\boldsymbol{\alpha}}^T)^{-1} \mathbf{b}^{\boldsymbol{\alpha}}$
  Social Pressure $\leftarrow -(\mathbf{x} - \hat{\mathbf{x}}) \cdot \mathbf{t}(\mathbf{x})$
  $\varphi \leftarrow \varphi / 2$
end while
return $\boldsymbol{\alpha}$

flows, and equilibrate the flows for each destination iteratively while treating the flows associated with other destinations as background traffic. Thus, a detailed exposition of the multiple destination case will not presented.

3.4 Minimum Revenue Tolling

As discussed at the beginning of this chapter, multiple toll vectors may result in the same SOR state. In order to find the toll pattern that minimizes the total revenue using a linear program, it is first necessary to establish that the KKT conditions to the UER reformulation (3.1)–(3.3) are not just necessary but are also sufficient. The following proposition helps prove this fact.

Proposition 3.2. *The objective of the non-linear program* (3.1)–(3.3) *is convex. Hence, the KKT conditions* (3.6)–(3.8) *are both necessary and sufficient.*

Proof. Let, as before,

$$g(\mathbf{x}) = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int_0^{x_{ij}^s} t_{ij}^s(x)\ dx$$

$$f(\mathbf{z}) = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int_0^{\sum_{\theta:\theta_{ij}=s} z_{ij}^\theta} t_{ij}^s(z)dz$$

Since $x_{ij}^s = \sum_{\theta:\theta_{ij}=s} z_{ij}^\theta$, $g(\mathbf{x}) = f(\mathbf{z})$, and the vector $\mathbf{x}$ can be written as linear function of $\mathbf{z}$, $\mathbf{x} = \mathbf{M}\mathbf{z}$, where $\mathbf{M}$ is a state-message incidence matrix of zeros and ones.

Consider two distinct points $\mathbf{z}_1$ and $\mathbf{z}_2$. Let $\mathbf{z} = \varphi\mathbf{z}_1 + (1-\varphi)\mathbf{z}_2$ for some $\varphi \in (0,1)$. Multiplying both sides of this equation by $\mathbf{M}$, we get $\mathbf{M}\mathbf{z} = \varphi\mathbf{M}\mathbf{z}_1 + (1-\varphi)\mathbf{M}\mathbf{z}_2$, which implies $\mathbf{x} = \varphi\mathbf{x}_1 + (1-\varphi)\mathbf{x}_2$. However, note that $\mathbf{x}_1$ and $\mathbf{x}_2$ are not necessarily distinct as multiple link-message flows may produce the same link-state flows much like the relationship between path flows and link flows in deterministic TAPs. Since $g(\cdot)$ is a strictly convex function and $g(\mathbf{x}) = f(\mathbf{z})$,

$$g(\mathbf{x}) \leq \varphi g(\mathbf{x}_1) + (1-\varphi)g(\mathbf{x}_2) \tag{3.45}$$

$$\Rightarrow f(\mathbf{z}) \leq \varphi f(\mathbf{z}_1) + (1-\varphi) f(\mathbf{z}_2) \tag{3.46}$$

Therefore, f is convex and the KKT conditions (3.6)–(3.8) are both necessary and sufficient. ■

Note that since f is not strictly convex the optimal solution is not necessarily unique in link-message flows.

Equations (3.6)–(3.8) are the KKT conditions for the multiple origin, single destination UER model. We can extend this to the multiple OD case in the following way

$$t_{ij}^{\theta v} + \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta}} \lambda_j^{\bar{\theta} v} - \lambda_i^{\theta v} \geq 0 \qquad \forall\, v \in Z, (i,j) \in A, \theta \in \Theta_i \tag{3.47}$$

$$z_{ij}^{\theta v} \left(t_{ij}^{\theta v} + \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta}} \lambda_j^{\bar{\theta} v} - \lambda_i^{\theta v} \right) = 0 \qquad \forall\, v \in Z, (i,j) \in A, \theta \in \Theta_i \tag{3.48}$$

$$\lambda_v^{\theta v} = 0 \qquad \forall\, v \in Z, \theta \in \Theta_v \tag{3.49}$$

Suppose the vector $\bar{\boldsymbol{\alpha}}$ represents the multi-commodity SOR solution. Let that the resulting flow variables be represented as $\bar{\mathbf{x}}$ and $\bar{\mathbf{z}}$. Using the split proportions and link-state flows, we can determine if $\bar{z}_{ij}^{\theta v}$ is strictly positive or zero. From equation (3.20), notice that $\bar{z}_{ij}^{\theta v}$ is strictly positive only if both $\bar{\alpha}_{ij}^{\theta v}$ and the inflow to node i for destination v are positive. This information helps us simplify the KKT conditions and formulate the minimum revenue problem as a linear program (see equations (3.50)–(3.54)).

$$\min_{\mathbf{c},\boldsymbol{\lambda}} \sum_{v\in Z}\sum_{(i,j)\in A}\sum_{\theta\in\Theta_i} \bar{z}_{ij}^{\theta v} c_{ij}^{\theta v} \qquad \text{(MINREV-1)} \quad (3.50)$$

$$\text{s.t. } \lambda_i^{\theta v} - c_{ij}^{\theta v} - \sum_{\bar{\theta}\in\Theta_j} q_j^{\bar{\theta} v}\lambda_j^{\bar{\theta} v} = \bar{t}_{ij}^{\theta} \quad \forall v \in Z, (i,j)\in A, \theta\in\Theta_i : \bar{z}_{ij}^{\theta v} > 0 \quad (3.51)$$

$$\lambda_i^{\theta v} - c_{ij}^{\theta v} - \sum_{\bar{\theta}\in\Theta_j} q_j^{\bar{\theta} v}\lambda_j^{\bar{\theta} v} \leq \bar{t}_{ij}^{\theta} \quad \forall v \in Z, (i,j)\in A, \theta\in\Theta_i : \bar{z}_{ij}^{\theta v} = 0 \quad (3.52)$$

$$\lambda_v^{\theta v} = 0 \qquad \forall v\in Z, \theta\in\Theta_v \quad (3.53)$$

$$c_{ij}^{\theta v} \geq 0 \qquad \forall v\in Z, (i,j)\in A, \theta\in\Theta_i \quad (3.54)$$

Equations (3.51) and (3.52) are identical to equations (3.47) and (3.48) except for an additional term $c_{ij}^{\theta v}$, which represents the toll paid by travelers heading to v when they choose to travel on arc (i,j) after receiving a message θ at node i. The objective represents sum total of the tolls paid by the travelers in the network. Clearly, the marginal tolls are feasible for the MINREV-1 formulation.

The tolls on links in this formulation are a function of the destination and the message received at the tail node. However, from a practical standpoint, to implement such a tolling mechanism, the system manager would require a lot of information from travelers, some of which may be difficult to obtain for privacy reasons. Alternately, by imposing additional constraints, it is possible to estimate minimum revenue tolls for each link-state as shown below.

$$\min_{\mathbf{c},\boldsymbol{\varsigma},\boldsymbol{\lambda}} \sum_{(i,j)\in A}\sum_{s\in S_{ij}} \bar{x}_{ij}^{s}\varsigma_{ij}^{s} \qquad \text{(MINREV-2)} \quad (3.55)$$

$$\text{s.t. } \lambda_i^{\theta v} - c_{ij}^{\theta v} - \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta} v} \lambda_j^{\bar{\theta} v} = \bar{t}_{ij}^{\theta} \quad \forall v \in Z, (i,j) \in A, \theta \in \Theta_i : \bar{z}_{ij}^{\theta v} > 0 \tag{3.56}$$

$$\lambda_i^{\theta v} - c_{ij}^{\theta v} - \sum_{\bar{\theta} \in \Theta_j} q_j^{\bar{\theta} v} \lambda_j^{\bar{\theta} v} \leq \bar{t}_{ij}^{\theta} \quad \forall v \in Z, (i,j) \in A, \theta \in \Theta_i : \bar{z}_{ij}^{\theta v} = 0 \tag{3.57}$$

$$\varsigma_{ij}^{s} = c_{ij}^{\theta v} \quad \forall v \in Z, (i,j) \in A, \theta \in \Theta_i : \theta_{ij} = s \tag{3.58}$$

$$\lambda_v^{\theta v} = 0 \quad \forall v \in Z, \theta \in \Theta_v \tag{3.59}$$

$$c_{ij}^{\theta v} \geq 0 \quad \forall v \in Z, (i,j) \in A, \theta \in \Theta_i \tag{3.60}$$

Equation (3.58) in the formulation MINREV-2 is an additional constraint that ensures that the toll paid by travelers on each link is purely a function of the link-state. Note also that in both MINREV-1 and MINREV-2, we disallow tolls from being negative. Negative tolls have been used in minimum revenue pricing models as they act as an instrument for providing incentives to travelers (Hearn and Ramana, 1998). In deterministic TAPs, one would just need to design incentives such that there are no negative cycles because a negative cycle would incentivize travelers to infinitely cycle and exploit the system. However, designing negative tolls or incentives is more complicated in the presence of supply-side uncertainty because detecting an unbounded instance in which travelers cycle indefinitely is not trivial (Provan, 2003; Boyles and Rambha, 2016). Hence, we stick to the minimum revenue problem with non-negativity constraints in this dissertation.

Remark. For formulating MINREV-1 and MINREV-2, knowledge of the signs of $\bar{z}_{ij}^{\theta}$s would suffice. In this chapter, we described a solution method using split proportions to estimate $\bar{\mathbf{z}}$. Alternately, one can also construct a feasible

$\bar{\mathbf{z}}$ from a vector of optimal link-state flows $\bar{\mathbf{x}}$ that solves the SOR formulation discussed in the previous chapter, and use it to populate the constraints of the minimum revenue pricing problem. Additionally, a maximum entropy like objective can be used to find a vector of $\bar{\mathbf{z}}$ from the optimal link-state flows $\bar{\mathbf{x}}$. This would maximize the number of non-zero elements in $\bar{\mathbf{z}}$ and in turn result in a tighter constraint set when solving MINREV-1 and MINREV-2. However, since the link-state flows from the methods discussed in the previous chapter are aggregated across all destinations, in order to compute a disaggregated solution, one would need to reformulate the equilibrium model in a manner similar to that proposed by Nie (2012).

3.5 Results

In this section, we illustrate the results of minimum revenue formulations MINREV-1 and MINREV-2 on the Sioux Falls test network that was introduced in the previous chapter. Algorithm 4 was used to solve the SOR problem to a gap of 10^{-6} and the optimal split proportions were used to construct the constraints of the LPs. Since the optimality gap is not exactly zero, we relax the equality constraints (3.52) and (3.57) as suggested by Dial (2000) and allow the left hand sides to lie in an ϵ-neighborhood of 0. The linear programs were solved using CPLEX's dual simplex algorithm (v12.6.2). The optimal objective values, problem sizes, and computation times are shown in Table 3.1.

Table 3.1: Summary of Minimum Revenue Results.

	Marginal Tolls	MINREV-1	MINREV-2
Revenue	1.88e+07	77547.50	5.93e+06
No. of Variables	-	27264	27416
No. of Constraints	-	21312	42624
Computation Time (sec)	-	0.23	4.25

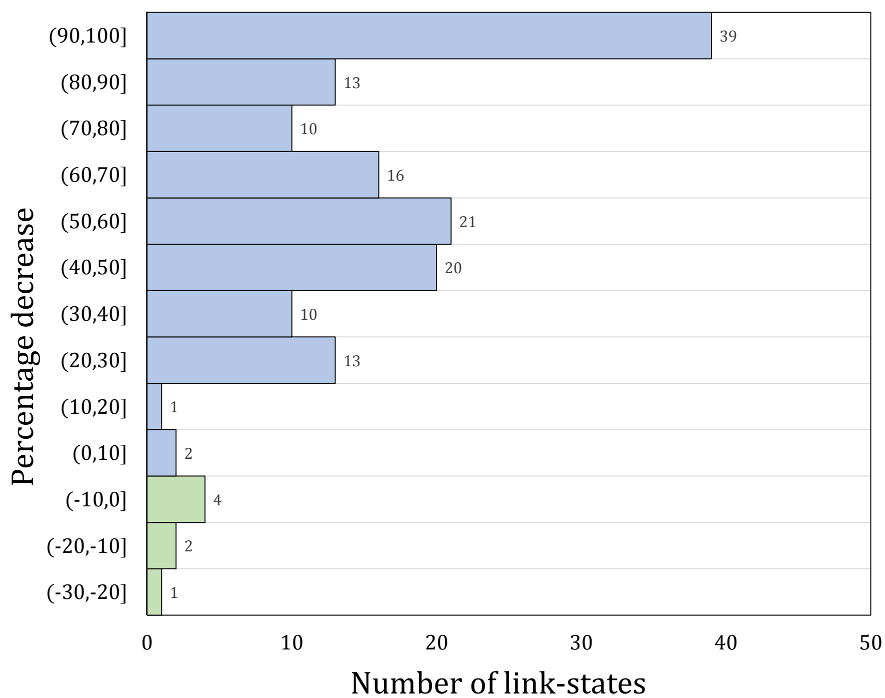


Figure 3.2: Histogram of the percentage decrease in tolls.

As expected the solutions to the linear program outperform the revenue from the marginal tolls. It was also found that the extent of improvement due differential tolls from MINREV-1 is orders of magnitude greater than that due to MINREV-2 tolls. Tables 3.2 and 3.3 show the SOR flows, marginal, and minimum revenue tolls from MINREV-2 for the two possible states on each link. A histogram of the percentage decrease in tolls is demonstrated in Figure 3.2. For most link-states the tolls are reduced by a significant fraction. The minimum revenue tolls for 7 link-states (shown in green) were however higher than that of the marginal tolls.

3.6 Summary

In this chapter, the SOR model was extended by computing tolls that optimize flow and minimize revenue, using two linear programming formulations. Such socially optimal, minimum revenue tolls can improve the acceptability of congestion pricing since they are not as burdensome as marginal tolls. In order to construct the constraints of the LPs, the flow on each arc, stratified by destinations and the message vector at the tail node was required. Since this information is not readily available from the output of a link-state based SOR formulation, a new convex optimization model was suggested.

We first established that any set of policy flows constructed from the output of this new optimization model satisfies the UER principle. This was followed by the development of a solution method that involved estimating the optimal

Table 3.2: Comparison of marginal and minimum revenue tolls.

Arc	*Normal State*			*Disrupted State*		
	Flow	*Marg. Toll*	*Min Toll*	*Flow*	*Marg. Toll*	*Min Toll*
(1, 2)	7266.20	0.034	0.000	807.36	0.544	0.000
(1, 3)	10572.20	0.152	0.000	1174.68	2.437	0.000
(2, 1)	7453.55	0.038	0.000	794.57	0.510	0.000
(2, 6)	6479.31	13.335	6.056	417.54	24.140	16.711
(3, 1)	10665.60	0.158	0.000	906.65	0.865	0.000
(3, 4)	15826.60	2.678	0.703	1233.95	10.386	7.782
(3, 12)	13032.30	0.352	0.000	1419.20	5.193	0.656
(4, 3)	15496.60	2.461	0.000	1289.92	12.403	7.490
(4, 5)	18494.90	2.140	2.763	1179.70	3.719	4.064
(4, 11)	6261.72	14.528	10.594	396.31	24.470	18.062
(5, 4)	17721.10	1.804	0.000	1793.11	19.849	7.780
(5, 6)	6870.10	13.595	7.463	387.64	14.466	7.884
(5, 9)	15948.90	29.585	5.093	941.35	37.692	11.832
(6, 2)	6663.79	14.919	5.986	407.62	21.927	12.386
(6, 5)	6703.30	12.322	8.439	447.24	25.632	16.977
(6, 8)	11740.10	60.341	22.240	733.17	96.344	47.055
(7, 8)	12559.30	18.051	0.000	741.46	23.019	3.546
(7, 18)	16364.80	0.437	0.000	1706.28	5.425	3.541
(8, 6)	11862.50	62.897	0.000	678.18	70.532	6.634
(8, 7)	12512.50	17.783	11.187	782.80	28.598	16.601
(8, 9)	7187.85	37.527	0.000	476.60	76.148	14.608
(8, 16)	7636.67	23.991	14.314	424.25	23.989	14.314
(9, 5)	15761.50	28.219	0.000	1075.57	64.239	21.032
(9, 8)	7275.35	39.388	19.015	452.20	61.712	31.896
(9, 10)	20638.00	13.272	13.987	1566.59	46.258	35.788
(10, 9)	20978.60	14.170	0.000	1336.01	24.468	7.399
(10, 11)	16569.00	34.462	0.000	1101.40	70.635	21.120
(10, 15)	21929.80	38.071	11.399	1376.24	61.990	25.345
(10, 16)	10092.00	68.300	33.383	576.36	76.276	37.342
(10, 17)	7904.38	45.932	19.408	473.21	61.937	29.377
(11, 4)	6240.72	14.334	2.094	403.61	26.323	7.537
(11, 10)	16464.00	33.596	13.159	1012.25	50.396	23.544
(11, 12)	7173.61	25.025	9.298	398.53	25.024	9.298
(11, 14)	8869.80	40.037	12.525	579.67	76.670	34.911
(12, 3)	13405.00	0.394	0.000	1145.97	2.208	0.425
(12, 11)	7055.94	23.423	17.584	400.75	25.587	18.996
(12, 13)	14410.30	0.263	0.000	1601.14	4.206	0.000
(13, 12)	14565.50	0.274	0.000	1529.96	3.507	2.670

Table 3.3: *Comparison of marginal and minimum revenue tolls (continued).*

Arc	Normal State *Flow*	Normal State *Marg. Toll*	Normal State *Min Toll*	Disrupted State *Flow*	Disrupted State *Marg. Toll*	Disrupted State *Min Toll*
(13,24)	9930.22	52.939	21.269	567.57	59.307	26.591
(14,11)	8894.13	40.478	2.478	562.93	68.188	19.749
(14,15)	8518.09	34.825	14.079	498.26	42.799	23.458
(14,23)	7276.71	17.435	10.430	472.28	32.476	35.785
(15,10)	21833.50	37.407	0.000	1345.74	56.675	10.045
(15,14)	8495.05	34.450	0.000	527.66	53.830	10.096
(15,19)	17191.60	5.325	0.000	1230.59	14.677	9.558
(15,22)	15400.60	18.177	5.550	1013.98	35.857	14.756
(16,8)	7635.49	23.976	2.746	424.24	23.986	2.743
(16,10)	10184.40	70.836	19.094	586.50	81.784	24.846
(16,17)	10006.60	24.512	17.468	609.74	35.474	23.238
(16,18)	18611.00	2.194	0.000	1570.77	11.688	2.414
(17,10)	8134.11	51.510	2.034	471.85	61.231	7.647
(17,16)	9892.43	23.412	11.805	646.13	44.731	25.089
(17,19)	7766.72	12.290	0.000	567.85	36.865	19.923
(18,7)	16465.70	0.448	0.000	1610.83	4.309	0.192
(18,16)	18957.20	2.362	0.000	1403.78	7.456	3.927
(18,20)	20274.90	2.060	0.000	1537.55	7.154	3.938
(19,15)	16885.80	4.957	2.946	1453.28	28.548	16.651
(19,17)	7940.36	13.426	13.665	544.75	31.222	23.333
(19,20)	8276.59	27.407	19.952	499.67	38.219	25.416
(20,18)	20112.60	1.995	0.135	1784.51	12.980	7.124
(20,19)	8369.29	28.656	13.340	474.51	31.083	14.799
(20,21)	6284.35	13.056	0.000	411.68	25.239	7.666
(20,22)	7113.59	17.641	0.000	443.91	28.084	7.701
(21,20)	6201.49	12.381	12.989	432.03	30.612	23.137
(21,22)	8024.87	10.139	3.814	607.86	35.039	18.887
(21,24)	9128.31	33.441	0.000	586.96	60.014	16.481
(22,15)	15210.60	17.296	16.712	1066.51	43.885	32.843
(22,20)	7232.12	18.847	17.912	440.06	27.122	24.207
(22,21)	8098.50	10.516	7.822	506.54	16.896	7.382
(22,23)	8759.53	34.458	7.531	554.32	58.007	22.173
(23,14)	7264.19	17.316	9.057	486.03	36.429	20.211
(23,22)	8721.51	33.863	6.525	541.88	52.976	17.679
(23,24)	7677.91	9.555	0.000	555.92	27.568	11.449
(24,13)	9873.47	51.739	7.314	608.35	78.280	24.137
(24,21)	9116.70	33.271	13.920	563.75	51.067	24.078
(24,23)	7631.40	9.326	10.802	553.21	27.035	18.846

split proportions, i.e., the fraction of travelers choosing different downstream arcs at each node-state. The proposed models were tested on the Sioux Falls test network and the results indicate a 68% decrease in the total toll collected.

Chapter 4

Day-to-Day Pricing: Closed Form Route Choice Dynamics

4.1 Introduction

In the equilibrium models discussed so far, it has been assumed that travelers are rational and have a perfect knowledge of the network topology and its response to congestion. However, when a large number of humans interact, the extent of reasoning required to arrive at an equilibrium solution is beyond one's ability. Two alternate concepts which do not rely on these assumptions exist in literature – *stochastic user equilibrium* (SUE) and *day-to-day dynamic models or Markovian traffic assignment models*. Both these approaches infuse uncertainty into travelers' choices and the uncertainty is assumed to result from randomness in users' perceived travel times. However, they differ from each other in a vital way. Stochastic user equilibrium models (Dial, 1971; Daganzo and Sheffi, 1977; Sheffi, 1985), which are formulated as fixed point problems, define equilibrium as a state in which users' perceived travel times are minimized.

On the other hand, day-to-day models (Cascetta, 1989; Friesz et al., 1994; Cantarella and Cascetta, 1995) are deterministic or stochastic dynamic processes in which states/feasible flows evolve over time. In discrete time models with stochastic dynamics, travelers select paths each day based on historical information of network travel times and a probabilistic route choice mechanism which induces transitions from one state to another. Under certain mild conditions, the stochastic process can be modeled as a Markov chain with a unique steady state distribution. Thus, although the system is never at rest, it attains an 'equilibrium' in the probability distribution of flow patterns.

Since paths are selected randomly on each day, the total system travel time is no longer deterministic but is a random variable. Using the steady state distribution of the stochastic process, one can compute the expected total system travel time (TSTT), which can be used as a metric for studying the extent of congestion in the network. An immediate question of interest is the following. Just as congestion pricing is used to achieve SO flows in traffic assignment, can a system manager do the same to reduce the expected TSTT?

Selecting the right tolls in a day-to-day setting would thus require us to estimate the steady state distribution for each admissible toll pattern and select one that minimizes the expected TSTT. However, one can do better than such static tolling schemes by dynamically varying tolls. While dynamic tolling has received some attention in literature, most existing research focuses primarily on continuous time models. These studies use control theory to determine the optimal time varying toll as the system state evolves with time according to some deterministic dynamic (Friesz et al., 2004; Xiao et al., 2014). However, continuous time formulations are not really 'day-to-day' models and their solutions cannot be used to dynamically price a network over different days. A major contribution of this research is in addressing this gap by proposing a dynamic day-to-day pricing mechanism in a discrete time setting that computes the optimal link tolls to reduce the expected TSTT. We formulate this problem as an infinite horizon average cost Markov decision process (MDP) and seek stationary policies that inform us the tolls as a function of the state of the system. In other words, the system manager observes the state or flow pattern and sets tolls, which are then revealed to the travelers. Travelers pick paths the next day in a probabilistic manner depending on the current state and revealed tolls.

Tolls in real world transportation networks are largely levied on freeways and hence the path choice set for travelers may be assumed to be small. However, even in sparse networks, presence of a large number of travelers results in an exponential number of states. Therefore, as with most MDPs, we are faced with the *curse of dimensionality* that prevents us from using this model on practical networks. To address this problem, we also propose simple approximation techniques using state space aggregation to handle instances with a large number of travelers and demonstrate its performance on a small test network. For the most part, we will restrict our attention to a single OD pair and the logit choice model for route selection. Extensions to more general settings are conceptually straightforward. In the next chapter, we will relax the assumption of the knowledge of closed form expressions for route choice and explore learning based approaches.

The rest of this chapter is organized as follows. In Section 4.2, we describe the two approaches (discrete and continuous) to model the evolution of traffic as a stochastic process. We also discuss existing literature on dynamic pricing. Section 4.3 describes an average cost MDP model for finding a dynamic pricing policy that minimizes the expected TSTT. In Section 4.4, we propose an approximate dynamic programming method using state space aggregation and test its performance on a simple network in Section 4.5. In Section 4.6 we formulate other variant MDPs such as those that optimize the probability of convergence to a flow solution and those that involve incentives and summarize our findings.

4.2 Literature Review

Day-to-day traffic models can be classified into two categories − *discrete* and *continuous*. Both these categories of models appear in literature in two flavors − *deterministic* and *stochastic*. The nomenclature is sometimes misleading as continuous time route switching processes are also referred to as 'day-to-day' models. In this section we review existing literature on these models and dynamic pricing. The reader may refer to Watling and Hazelton (2003) and Watling and Cantarella (2013) for a more comprehensive summary of day-to-day dynamics and for their connections with UE and SUE.

4.2.1 Discrete Time Day-to-Day Models

Cascetta (1989) formulated the evolution of traffic as a discrete time Markov chain. The number of travelers was assumed to be finite. Extensions to model an infinite number of travelers also exist (see Watling and Cantarella, 2013). Under the following assumptions on the path choice probabilities, it was shown that the stochastic process has a unique steady state distribution: (1) time invariant path choice probabilities, (2) the probability of selecting any path between an OD pair is non-negative, and (3) the probability of choosing a path depends on the states (flow patterns) of the system on at most m previous days (which ensures that the process is m-dependent Markovian). Commonly used path choice models in literature include the logit and probit choice models. In logit choice models, the probability of selecting a path is additionally assumed to depend on a parameter θ which defines the extent of making a mistake or the extent of irrationality.

This model was extended by Cascetta and Cantarella (1991) to account for within day fluctuations in traffic. Travelers were assumed to have access to travel time information in periods prior to their departure and condition their choices based on historic day-to-day information and also within-day information. Watling (1996) studied day-to-day models for asymmetric traffic assignment problems (i.e., for ones in which the Jacobian of the cost functions is not symmetric and multiple equilibria may exist). The stationary distributions in such problems were found to have multiple modes at the stable equilibria or a unimodal shape if one of the equilibrium dominated the others.

Several efforts have been made to estimate the expected route flows and the correlations among flow patterns in day-to-day models (Davis and Nihan, 1993; Hazelton and Watling, 2004) as the computation of steady state distributions of Markov chains for cases with a large number of travelers can be intensive even when using Monte Carlo simulations. For networks with a large number of travelers, the expected flows may be approximated to an SUE solution (Davis and Nihan, 1993). Discrete time day-to-day models (see Cantarella and Cascetta, 1995) with deterministic dynamics have also been studied in literature. These models employ a deterministic mapping, e.g., best response mechanism (see Brown, 1951; Robinson, 1951), that provides the state of the system on the next day as a function of the flows observed on previous days.

4.2.2 Continuous Time Day-to-Day Models

Continuous time day-to-day dynamics may also be modeled as continuous time Markov chains in a manner similar to discrete time day-to-day models. This approach is relatively more common in behavioral economics (see Sandholm, 2010, Chapters 10-12). Travelers are assumed to be atomic (i.e., flows are integral) and their choices are characterized by − inertia, myopic behavior, and mutations. Inertia implies that travelers do not frequently change paths but do so only when they are given a strategy revision opportunity, which presents itself at random times. The sojourn times for each traveler (time between two successive revision opportunities) are assumed to be exponentially distributed. Myopic behavior implies that travelers choose actions to optimize their present travel times rather than infinite-horizon discounted travel times. Mutations reflect the assumption that travelers may "tremble" or make mistakes while choosing a path. Depending on the probabilities that are assigned to the strategies that are not best responses, different learning algorithms can be constructed (Young, 1993, 2004; Kandori et al., 1993; Kandori and Rob, 1995; Blume, 1996; Rambha and Boyles, 2013). Under certain assumptions, existence of a unique steady state/limiting distribution can be ensured. Blume (1996) showed that for logit choice models, as θ tends to zero, the set of states with positive limiting probabilities (called a *stochastically stable set*) coincides with the set of NE. Further, for cases with a large number of players, it may be shown that deterministic and stochastic approaches are equivalent to each other when observed for a finite period of time (Sandholm, 2010).

The deterministic version of continuous day-to-day models assumes that the state of the system evolves with time as an ordinary differential equation and has been widely studied in transportation literature. Travelers are usually assumed to be infinitely divisible (non-atomic). One of the most commonly used dynamic is the Smith's dynamic (**?**) in which users shift between routes at a rate proportional to the difference between their current travel times. Other deterministic dynamics that have appeared in literature in transportation and behavioral economics include replicator dynamics (Taylor and Jonker, 1978; Smith and Price, 1973), projection dynamics (Nagurney and Zhang, 1997), and Brown-von Neumann-Nash dynamic (Brown and Von Neumann, 1950). The common objective in studying these models is to verify if the rest points of the dynamic are unique and coincide with the UE solution. Most deterministic dynamical systems are formulated using path flows. However, from a practical standpoint, as the number of paths may increase exponentially with the network size, researchers have recently developed more practical link based dynamic models (Zhang et al., 2001; He et al., 2010; Han and Du, 2012; Guo et al., 2015). In this context, Yang and Zhang (2009)and Guo et al. (2013) proposed a class of dynamic route choice models called rational adjustment processes (whose stationary states are at UE) in continuous and discrete time settings respectively.

4.2.3 Dynamic Pricing

In the context of day-to-day models, existing methods usually focus on continuous time versions and are formulated as optimal control problems (Wie and Tobin, 1998; Friesz et al., 2004; Yang, 2008). The system is assumed to be tolled for a finite period of time and, using boundary conditions, it is ensured that the network remains in a SO state at the end of the finite time horizon. Friesz et al. (2004) developed such a pricing model in which the objective was to maximize net social benefits while ensuring that a minimum revenue target is met. Several other alternate objectives may be modeled using this framework such as minimizing travel costs and minimizing the time required to guide the system to an SO state (Xiao et al., 2014). Other related pricing literature includes studies to achieve SO flows using static tolls when users route according to SUE (Smith et al., 1995; Yang, 1999); piecewise constant pricing mechanisms that ensure the convergence of the multiplicative update rule and replicator dynamics to an SO state (Farokhi and Johansson, 2015); self-learning and feedback learning controller for tolling managed lanes (Yin and Lou, 2009); and time-varying tolls in dynamic traffic assignment or within-day dynamic models (Joksimovic et al., 2005).

4.2.4 Summary

As noted in the previous subsections, there is a huge body of literature on different versions of day-to-day dynamic models. The congestion pricing methods developed in this chapter optimize network performance in the presence of daily fluctuation in traffic flows. Hence, we use a discrete time stochastic day-to-day dynamic model along the lines of those developed by Cascetta (1989).

4.3 Dynamic Pricing – Average Cost MDP Formulation

In this section, we introduce the four components of the MDP − the state space, the action space, transition probabilities and the costs and discuss a commonly used method for solving it.

4.3.1 Preliminaries

We make the following assumptions for the day-to-day traffic model with tolls:

1. The network has a single origin-destination (OD) pair with r routes. The formulation may be extended to include multiple OD pairs but has been avoided to simplify the notation.

2. There are a total of n travelers (assumed atomic and finite). Throughout this chapter and the next, the number of travelers will be assumed to be integral and fixed. Although finiteness is limiting because of demand uncertainty in networks, the treatment of models with elastic demand is a topic in itself and can be justly studied only if the problem with a fixed number of travelers is fully understood.

3. Tolls can be collected in discrete amounts and along routes in the network. The discretization is mainly to assist the computation of optimal tolling policies and is realistic since currencies have a lowest denomination. The assumption that tolls are collected at a route level is not restrictive because the problem may be easily reformulated using link level tolls (in which case the action space is a vector of link tolls).

4. Users make route choice decisions on a day-to-day basis based on a given route choice model in which travel times are replaced by generalized costs. Decisions are conditioned on the previous day's travel times. All travelers are also assumed to be homogenous with the same value of travel time.

5. The objective of the system manager is to minimize the expected TSTT. Other objectives that may be of interest are discussed in Section 4.6.1.

State space

Let $R = \{1, \ldots, r\}$ denote the set of routes. We define the state space S as the set of all feasible route flow solutions, i.e., $\left\{(x_1, x_2, \ldots, x_r) \in \mathbb{Z}_+^r : \sum_{i \in R} x_i = n\right\}$. The vector $\mathbf{x} = (x_1, x_2, \ldots, x_r) \in S$ contains the flows on each of the paths between the OD pair. Since we are dealing with a network with r routes and n travelers, there are a total of $\binom{n+r-1}{n}$ feasible flow solutions/states. We use $\mathbf{x}_k$ to denote the state of the system at time step/day k.

Action space

We will represent an action using a toll vector $\mathbf{u} = (u_1, u_2, \ldots, u_r)$, which denotes the tolls on the paths in the network. Assume that the action space at state $\mathbf{x}$ is $U(\mathbf{x})$. Also suppose that the action space for each state $\mathbf{x}$ is the Cartesian product $\{\tau_1, \tau_2, \ldots, \tau_l\}^r$ where $\tau_1, \ldots, \tau_l$ are some allowable prices.

Transition probabilities

Let $t_i : S \to \mathbb{R}$ be the travel time on path $i \in R$ as a function of the state. We assume that the travel time functions are bounded. No further assumptions such as separability or monotonicity are needed. We suppose that the path choice probability $q_r(\mathbf{x}, \mathbf{u})$ for each traveler is a function of $\mathbf{x}$ and $\mathbf{u}$ and is positive for all routes for all state-action pairs. We further suppose that each traveler independently chooses a path using this distribution. Thus, the probability of moving from state $\mathbf{x}$ to $\mathbf{y}$ when action $\mathbf{u} \in U(\mathbf{x})$ is taken in state $\mathbf{x}$

is given by the following multinomial probability distribution

$$\Pr\left[\mathbf{y} = (y_1, y_2, \ldots, y_r) | \mathbf{x}, \mathbf{u}\right] = p_{\mathbf{xy}}(\mathbf{u}) = \frac{n!}{y_1! y_2! \ldots y_r!} q_1(\mathbf{x}, \mathbf{u})^{y_1} \ldots q_r(\mathbf{x}, \mathbf{u})^{y_r} \tag{4.1}$$

If travelers use the logit choice model with parameter θ, we may write $q_r(\mathbf{x}, \mathbf{u})$ as follows:

$$q_r(\mathbf{x}, \mathbf{u}) = \frac{e^{-\theta[t_r(\mathbf{x}) + u_r]}}{\sum_{i=1}^{r} e^{-\theta[t_i(\mathbf{x}) + u_i]}} \tag{4.2}$$

where $t_i(\mathbf{x}) + u_i$ is the generalized cost on route i. Then, the transition probabilities take the form

$$p_{\mathbf{xy}}(\mathbf{u}) = \frac{n!}{y_1! y_2! \ldots y_r!} \left(\frac{e^{-\theta[t_1(\mathbf{x}) + u_1]}}{\sum_{i=1}^{r} e^{-\theta[t_i(\mathbf{x}) + u_i]}} \right)^{y_1} \ldots \left(\frac{e^{-\theta[t_r(\mathbf{x}) + u_r]}}{\sum_{i=1}^{r} e^{-\theta[t_i(\mathbf{x}) + u_i]}} \right)^{y_r} \tag{4.3}$$

$$= \frac{n!}{y_1! y_2! \ldots y_r!} \prod_{j=1}^{r} \left(\frac{e^{-\theta[t_j(\mathbf{x}) + u_j]}}{\sum_{i=1}^{r} e^{-\theta[t_i(\mathbf{x}) + u_i]}} \right)^{y_j} \tag{4.4}$$

Remark. Route choice processes in day-to-day models can be made more general than what has been described above. The system state on day k usually includes historical information and is defined as a vector of flows on previous m days $\left(\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_{k-(m-1)}\right)$. Travelers are assumed to compute perceived travel times $\tilde{t}_i(.)$ for each path i between their OD pair as the sum of a weighted average of travel times on route i on previous m days and a random term that accounts for perception errors or unobserved factors.

$$\tilde{t}_i\left(\left(\mathbf{x}_k, \ldots, \mathbf{x}_{k-(m-1)}\right)\right) = \sum_{j=0}^{m-1} w_j t_i\left(\mathbf{x}_{k-j}\right) + \tilde{\epsilon} \tag{4.5}$$

The terms w_j represent the weights associated with the observed travel times on previous days. The probability of choosing path i is thus given by

$$\Pr\left[\tilde{t}_i\left((\mathbf{x}_k,\ldots,\mathbf{x}_{k-(m-1)})\right) < \tilde{t}_{i'}\left((\mathbf{x}_k,\ldots,\mathbf{x}_{k-(m-1)})\right)\ \forall\, i \neq i', i' \in R\right] \quad (4.6)$$

Depending on the assumed distributions of the error terms, different route choice models such as logit and probit may be obtained. Logit choice models are relatively widely used as the path choice probabilities have a closed form expression.

Costs

Let $g(\mathbf{x},\mathbf{u})$ denote the expected cost incurred every time decision $\mathbf{u}$ is taken in state $\mathbf{x}$. In order to minimize the expected TSTT, we define the cost as $g(\mathbf{x},\mathbf{u}) = \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u})\mathrm{TSTT}(\mathbf{y})$, where $\mathrm{TSTT}(\mathbf{y})$ is the total system travel time of state $\mathbf{y}$. Note that $g(\mathbf{x},\mathbf{u})$ is bounded because the travel time functions t_i are assumed to be bounded.

4.3.2 Objective and Algorithms

The system manager observes the state on a particular day and chooses the tolls based on some policy $\left(\boldsymbol{\mu}(\mathbf{x})\right)_{\mathbf{x}\in S}$, which specifies the action $\boldsymbol{\mu}(\mathbf{x}) \in U(\mathbf{x})$ to be taken when in state $\mathbf{x}$ and reveals them to the travelers before the next day. Travelers make decisions based on the previous day's state and the revealed tolls as shown in Figure 4.1.

Decisions are made over an infinite horizon but at discrete intervals of time $0, 1, 2, \ldots, k, \ldots$. Let $J_{\boldsymbol{\mu}}(\mathbf{x})$ be the average cost per stage or the expected

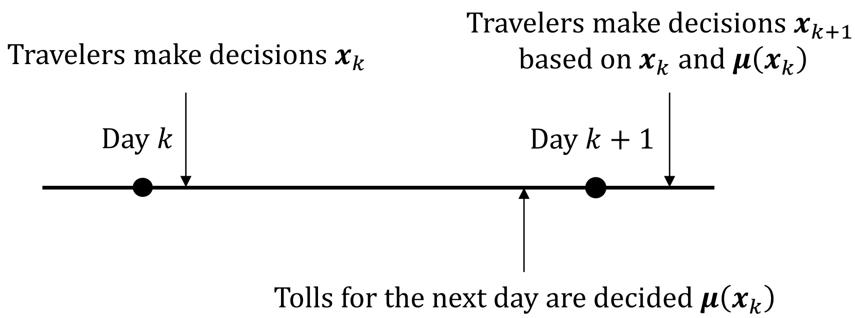


Figure 4.1: Timeline for the pricing mechanism

TSTT for policy $\boldsymbol{\mu}$ assuming that the system starts at state $\mathbf{x}$, i.e., $\mathbf{x}_0 = \mathbf{x}$. Thus, we may write

$$J_{\boldsymbol{\mu}}(\mathbf{x}) = \lim_{K \to \infty} \frac{1}{K} \mathbb{E} \left\{ \sum_{k=0}^{K-1} g\left(\mathbf{x}_k, \boldsymbol{\mu}(\mathbf{x}_k)\right) \,\middle|\, \mathbf{x}_0 = \mathbf{x} \right\} \tag{4.7}$$

Remark. A majority of infinite horizon MDPs are formulated as discounted cost problems. In this chapter, assuming that the system manager minimizes the expected TSTT, we use the average cost version instead for a couple of reasons. First, average cost MDPs are mathematically attractive as their objective values, as we will see shortly, do not depend on the initial state of the system. On the other hand, discounted cost problems are often extremely sensitive to both initial conditions and the discount factor.

Second, and more importantly, discounted cost models are appropriate when the costs associated with state-action pairs have an economic value. In such cases, the discount factor can simply be set to the interest rate. Although estimating the monetary value of system wide travel time savings in transportation networks appears difficult, one may still use the discounted cost framework to place more weight on near-term savings in TSTT, especially when the Markov chains associated with optimal average cost policies visits states with high TSTT initially and converges to states with low TSTT only after a long time. However, for the problem instances we tested (see Section 4.5), the Markov chains associated with the optimal policies were found to mix quickly and the time averages of TSTT over a finite number of initial days for different sample paths were fairly close to the optimal expected TSTT, and thereby did not motivate the need for discounting.

We restrict our attention to time-invariant or stationary policies (since we are only dealing with stationary policies, the above limit always exists). The advantages of stationary policies are two-fold. First, an optimal stationary policy is relatively easy to compute. Second, since the policies do not directly depend on the day k, implementing a stationary policy is much easier. Note that stationarity of policies does not imply that the tolls are static. It implies that the tolls are purely a function of the state and as the states of the network vary over time, so do the tolls. We seek an optimal policy $\boldsymbol{\mu}^*$ such that

$$J^*(\mathbf{x}) \equiv J_{\boldsymbol{\mu}^*}(\mathbf{x}) = \min_{\boldsymbol{\mu} \in \Pi} J_{\boldsymbol{\mu}}(\mathbf{x}) \tag{4.8}$$

where Π denotes the set of all admissible policies. We now state some standard results concerning average cost per stage MDPs that are relevant to the current chapter and the next. A more detailed account of these can be found in Puterman (2005) and Bertsekas (2007).

Definition 1. *For a given stationary policy $\boldsymbol{\mu}$, state $\mathbf{y}$ is said to be accessible from $\mathbf{x}$, and is denoted by $\mathbf{x} \to \mathbf{y}$, if for some $k > 0$, $\mathbf{y}$ can be reached from $\mathbf{x}$ with positive probability in k days, i.e.,* $\Pr[\mathbf{x}_k = \mathbf{y} | \mathbf{x}_0 = \mathbf{x}, \boldsymbol{\mu}] > 0$. *Further, if $\mathbf{x} \to \mathbf{y}$ and $\mathbf{y} \to \mathbf{x}$, we say that $\mathbf{x}$ communicates with $\mathbf{y}$. If $\mathbf{y}$ is not accessible from $\mathbf{x}$, we denote it by $\mathbf{x} \nrightarrow \mathbf{y}$.*

Definition 2. *For a given stationary policy $\boldsymbol{\mu}$, a subset of states $S' \subseteq S$ is a recurrent class or a closed communicating class if*

(a) All states in S' communicate with each other.

(b) $\mathbf{x} \in S'$ and $\mathbf{y} \notin S' \Rightarrow \mathbf{x} \nrightarrow \mathbf{y}$.

Definition 3. *An MDP is said to be ergodic if the Markov chain induced by every deterministic stationary policy is irreducible, i.e., has a single recurrent class.*

For the logit choice model described in this chapter, the path choice probabilities and the transition probabilities between every pair of states, defined using (4.2) and (4.4) respectively, are positive for all policies. Thus, using Definitions 1 and 2, we conclude that all states communicate with each other and belong to a single recurrent class. Therefore, by Definition 3, the MDP is ergodic.

Proposition 4.1 (Equal costs)**.** *If an MDP is ergodic then the average cost problem has equal costs, i.e,*

$$J^*(\mathbf{x}) = J^*(\mathbf{y}), \; \forall \, \mathbf{x}, \mathbf{y} \in S \tag{4.9}$$

Proof. Consider a stationary policy $\boldsymbol{\mu}$. Clearly, the cost incurred up to a finite number of stages do not matter when computing the expected TSTT $J_{\boldsymbol{\mu}}(\mathbf{x})$ under the policy $\boldsymbol{\mu}$ assuming that we start at state $\mathbf{x}$, i.e., suppose $K' < \infty$, then

$$\lim_{K\to\infty} \frac{1}{K}\mathbb{E}\left\{\sum_{k=0}^{K'-1} g\left(\mathbf{x}_k, \boldsymbol{\mu}(\mathbf{x}_k)\right) \,\Big|\, \mathbf{x}_0 = \mathbf{x}\right\} = 0 \tag{4.10}$$

Suppose the random variable $\widetilde{K}$ represents the time taken for the Markov chain to move from $\mathbf{x}$ to $\mathbf{y}$ for the first time under policy $\boldsymbol{\mu}$. Since the state $\mathbf{y}$ is accessible from $\mathbf{x}$ under the policy $\boldsymbol{\mu}$, it follows that $\mathbb{E}[\widetilde{K}] < \infty$. Therefore, using (4.7),

$$\begin{aligned} J_{\boldsymbol{\mu}}(\mathbf{x}) &= \lim_{K\to\infty} \frac{1}{K}\mathbb{E}\left\{\sum_{k=0}^{\widetilde{K}-1} g\left(\mathbf{x}_k, \boldsymbol{\mu}(\mathbf{x}_k)\right) \,\Big|\, \mathbf{x}_0 = \mathbf{x}\right\} \\ &\quad + \lim_{K\to\infty} \frac{1}{K}\mathbb{E}\left\{\sum_{k=\widetilde{K}}^{K-1} g\left(\mathbf{x}_k, \boldsymbol{\mu}(\mathbf{x}_k)\right) \,\Big|\, \mathbf{x}_{\widetilde{K}} = \mathbf{y}\right\} && (4.11) \\ &= 0 + \lim_{K\to\infty} \frac{1}{K}\mathbb{E}\left\{\sum_{k=\widetilde{K}}^{K-1} g\left(\mathbf{x}_k, \boldsymbol{\mu}(\mathbf{x}_k)\right) \,\Big|\, \mathbf{x}_{\widetilde{K}} = \mathbf{y}\right\} && (4.12) \\ &= J_{\boldsymbol{\mu}}(\mathbf{y}) && (4.13) \end{aligned}$$

Since the expected TSTT is independent of the initial state for every stationary policy, the same is true for the optimal policy. Hence, $J^*(\mathbf{x}) = J^*(\mathbf{y})$, $\forall\, \mathbf{x}, \mathbf{y} \in S$. ■

Thus, Proposition 1 implies that the optimal expected TSTT is independent of the initial conditions, i.e., state of the system on day 0.

Proposition 4.2 (Bellman's equation)**.** *Suppose* $\lambda^* = J^*(\mathbf{x})$*. Then exists* $h^*(\mathbf{x})\,\forall\,\mathbf{x} \in S$ *(not necessarily unique) that satisfies the following Bellman's equation*

$$\lambda^* + h^*(\mathbf{x}) = \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h^*(\mathbf{y}) \right\} \forall\,\mathbf{x} \in S \tag{4.14}$$

Also, if some λ *and a vector of* h*'s satisfy* (4.14)*, then* λ *is the optimal average cost per stage. Further, a policy* $\boldsymbol{\mu}^*(\mathbf{x})$ *defined as follows is optimal*

$$\boldsymbol{\mu}^*(\mathbf{x}) \in \operatorname*{argmin}_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h^*(\mathbf{y}) \right\} \forall\,\mathbf{x} \in S \tag{4.15}$$

Proof. See Bertsekas (2007). ■

Since the problem has a finite state space and a finite action space, the optimal J values and policies can be computed using value iteration, policy iteration or linear programming (LP). The value iteration method updates J's in the following manner and $\lambda^* = J^*(\mathbf{x})$ is obtained by evaluating $\lim_{k\to\infty} \frac{J_k(\mathbf{x})}{k}$ (k denotes the iteration number).

$$J_{k+1}(\mathbf{x}) = \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) J_k(\mathbf{y}) \right\} \forall\,\mathbf{x} \in S \tag{4.16}$$

where $J_0(\mathbf{x})$ can be initialized to any arbitrary value for all $\mathbf{x} \in S$. However, such an iterative procedure can lead to numerical instability as $J_k(\mathbf{x}) \to \infty$. This issue is typically avoided using relative value iteration in which we define a *differential cost vector* h_k as $h_k(\mathbf{x}) = J_k(\mathbf{x}) - J_k(\mathbf{s})\,\forall\,\mathbf{x} \in S$, where $\mathbf{s}$ is an arbitrary state in S. Hence for all $\mathbf{x} \in S$,

$$h_{k+1}(\mathbf{x}) = J_{k+1}(\mathbf{x}) - J_{k+1}(\mathbf{s}) \tag{4.17}$$

$$= \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) J_k(\mathbf{y}) \right\} - \min_{\mathbf{u}\in U(\mathbf{s})} \left\{ g(\mathbf{s},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{sy}}(\mathbf{u}) J_k(\mathbf{y}) \right\} \tag{4.18}$$

$$= \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h_k(\mathbf{y}) \right\} - \min_{\mathbf{u}\in U(\mathbf{s})} \left\{ g(\mathbf{s},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{sy}}(\mathbf{u}) h_k(\mathbf{y}) \right\} \tag{4.19}$$

The iterates generated by (4.19) and λ_{k+1} defined according to (4.20) converge and satisfy the Bellman's equation as defined in Proposition 4.2 (Puterman, 2005; Bertsekas, 2007).

$$\lambda_{k+1} = \min_{\mathbf{u}\in U(\mathbf{s})} \left\{ g(\mathbf{s},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{sy}}(\mathbf{u}) h_{k+1}(\mathbf{y}) \right\} \tag{4.20}$$

The pseudocode for relative value iteration is summarized in Algorithm 6. Let $\epsilon > 0$ denote the required level of convergence. Suppose that $M > \epsilon$ and $\mathrm{sp}(\cdot)$ represents the span semi-norm which is defined as $\mathrm{sp}(h) = \max_{\mathbf{x}\in S} h(\mathbf{x}) - \min_{\mathbf{x}\in S} h(\mathbf{x})$. The span semi-norm is used to compute the difference between the upper and lower bounds of the optimal expected TSTT λ^*.

4.4 Approximate Dynamic Programming – State Space Aggregation

The model formulated in the previous section, while being theoretically appealing, may not be suited for practical implementation especially when there are a large number of travelers or if there are many routes to choose from. For instance if 1000 travelers make route choices each day in a network with 10 routes, the size of the state space is equal to $\binom{1000+10-1}{1000} \approx 10^{21}$. The problem

Algorithm 6 *Pseudocode for relative value iteration*

Step 1:
Initialize $h_0(\mathbf{x})\,\forall\,\mathbf{x} \in S$ to any arbitrary values
$error \leftarrow M$
$k \leftarrow 0$

Step 2:
while $error > \epsilon$ **do**
  for each $\mathbf{x} \in S$ **do**
$$(Th_k)(\mathbf{x}) \leftarrow \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h_k(\mathbf{y}) \right\}$$
$$h_{k+1}(\mathbf{x}) \leftarrow (Th_k)(\mathbf{x}) - (Th_k)(\mathbf{s})$$
  end for
  if $k \geq 1$ **then** $error \leftarrow \mathrm{sp}(Th_k - Th_{k-1})$
  $k \leftarrow k + 1$
end while

Step 3:

Choose $\boldsymbol{\mu}^*(\mathbf{x}) \in \arg\min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u})(Th_{k-1})(\mathbf{y}) \right\} \forall\,\mathbf{x} \in S$

further gets compounded when we extend the model to multiple OD pairs. In this section, we address this issue by developing approximation methods that involve state space aggregation.[1]

When dealing with networks with a large number of travelers, several states may not be significantly different from each other. For instance, if there are a 1000 travelers in a network with two parallel links, the states (1000,0) and (999,1) are likely to be indistinguishable both in terms of the associated travel times and the optimal policies. This motivates us to develop approximate dynamic programming methods by aggregating states in order to reduce the computational times. Thus, we need to address the following questions: (1) how should states be aggregated? and (2) what are the transition probabilities between states in the aggregated system?

A simple attempt to aggregate states may be made using intervals of some chosen width. For instance, in the network with two parallel links, we may group states for which the flow on one of the links (say the top link) is between 0 and 10, 11 and 20 and so on. For any such aggregation/partition of the set S, transition probabilities between aggregated states may be computed by adding the transition probabilities between every pair of original states within two aggregated states. Although we save on the time required to compute

[1]State space aggregation was preferred over other approximate dynamic programming methods such as rollout algorithms (Bertsekas et al., 1997) and approximate linear programming (de Farias and Van Roy, 2006) because it avoids the enumeration of states and the computation of multinomially distributed transition probabilities.

the optimal policy, we would still have to enumerate all states and calculate the transition probabilities associated with states in the original state space as given by the expressions derived in (4.4).

Alternately, we can exploit the fact that for large n, a multinomial distributed random variable may be approximated to have a multivariate normal distribution (Sheffi, 1985). In order to do so, we first assume that the state space is continuous by supposing that travelers are infinitesimally divisible (non-atomic). Let $\mathbf{y}$ represent a vector of random variables (path flows) when action $\mathbf{u}$ is taken in state $\mathbf{x}$. In (4.1) we saw that $\mathbf{y}|\mathbf{x},\mathbf{u}$ is multinomially distributed. When n is large, we can approximate it with the multivariate normal $\mathbf{y}|\mathbf{x},\mathbf{u} \sim \mathcal{N}\left(\boldsymbol{\alpha}(\mathbf{x},\mathbf{u}), \boldsymbol{\Sigma}(\mathbf{x},\mathbf{u})\right)$, where $\boldsymbol{\alpha}(\mathbf{x},\mathbf{u}) = \Big(nq_1(\mathbf{x},\mathbf{u}), nq_2(\mathbf{x},\mathbf{u}), \ldots, nq_r(\mathbf{x},\mathbf{u})\Big)$ and $\boldsymbol{\Sigma}(\mathbf{x},\mathbf{u}) = [\Sigma_{ij}(\mathbf{x},\mathbf{u})]$ is the covariance matrix with elements given by

$$\Sigma_{ij}(\mathbf{x},\mathbf{u}) = \begin{cases} -nq_i(\mathbf{x},\mathbf{u})q_j(\mathbf{x},\mathbf{u}) & \text{if } i \neq j \\ nq_i(\mathbf{x},\mathbf{u})(1 - q_i(\mathbf{x},\mathbf{u})) & \text{otherwise} \end{cases} \tag{4.21}$$

The density function of $\mathbf{y}$, $f(\mathbf{y}|\mathbf{x},\mathbf{u})$, is given by

$$\frac{1}{\sqrt{(2\pi)^r \det \boldsymbol{\Sigma}(\mathbf{x},\mathbf{u})}} \exp\left(-\frac{1}{2}\left(\mathbf{y} - \boldsymbol{\alpha}(\mathbf{x},\mathbf{u})\right)^T \boldsymbol{\Sigma}(\mathbf{x},\mathbf{u})^{-1}\left(\mathbf{y} - \boldsymbol{\alpha}(\mathbf{x},\mathbf{u})\right)\right)$$

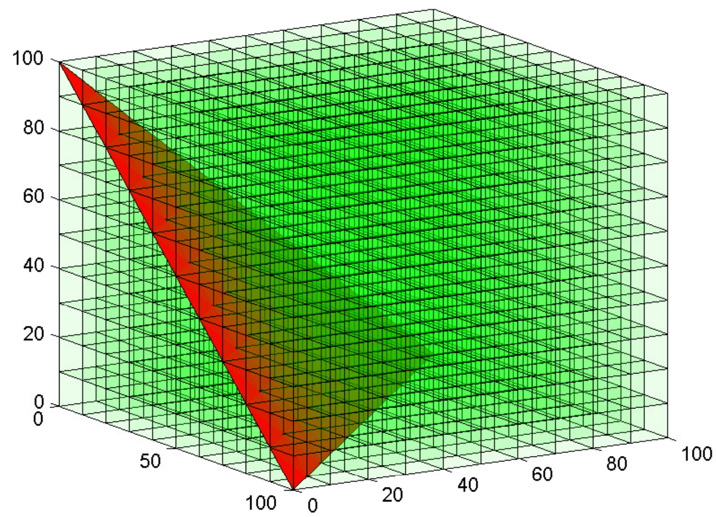
State space

The theory of infinite horizon MDPs is well established for problems with finite state and action spaces. In order to take advantage of existing methods to solve them, we construct a finite number of states from a continuous state space by generalizing the idea of aggregating states using intervals. Let us first define the set $\mathcal{I} = \left\{[0, \frac{n}{\delta}], [\frac{n}{\delta}, \frac{2n}{\delta}], \ldots, [\frac{(\delta-1)n}{\delta}, n]\right\}$. Notice that $\mathcal{I}^r$ is the set of all hypercubes formed by dividing the flow on each route into δ intervals. We then consider the space $\mathcal{S} = \{\mathcal{X} \in \mathcal{I}^r : |\mathcal{X} \cap \{\mathbf{x} \in [0, n]^r : \sum_{i=1}^r x_i = n\}| > 1\}$, where $|\cdot|$ represents the cardinality of a set. Figure 4.2 helps visualize this construct. Suppose there are 100 travelers and three routes, and the flows on each route are represented on the three axes. Assume that we divide the each axis into 10 intervals. This divides the space $[0, 100]^3$ into 1000 hypercubes as shown in Figure 4.2a. We then pick only those hypercubes which intersect the set of feasible flows (i.e, the simplex $x_1 + x_2 + x_3 = 100$) at more than one point, which gives the 100 hypercubes in Figure 4.2b. We exclude the hypercubes that intersect the simplex at exactly one point as we can always find another hypercube belonging to $\mathcal{S}$ that contains the point.

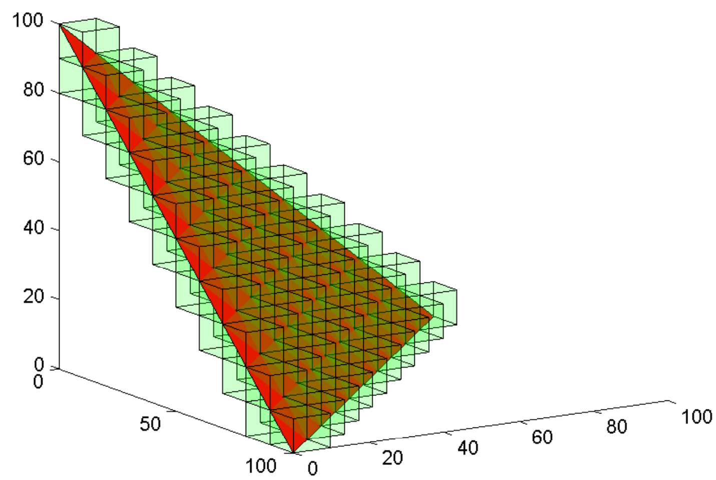
Let the state space for the approximate method be $\mathcal{S}$. For any state $\mathcal{X} \in \mathcal{S}$, let $\mathcal{X}_c \in \mathbb{R}^r$ be the center of the hypercube $\mathcal{X}$. We evaluate the properties of the state $\mathcal{X}$ such as the TSTT at this point. Notice that the point $\mathcal{X}_c$ may or may not satisfy the flow conservation constraint depending on the choice of r and δ. However, when we consider a sufficiently large number of intervals (δ), $\mathcal{X}_c$ may be assumed to be close enough to the simplex so that the errors in approximating the TSTT are small. We now define the remaining components of the MDP.

Action space

The action space for the approximate MDP at each state is same as before,



(a) Set of all hypercubes $\mathcal{I}^r$



(b) Hypercubes that intersect the simplex

Figure 4.2: State space for the approximate methods

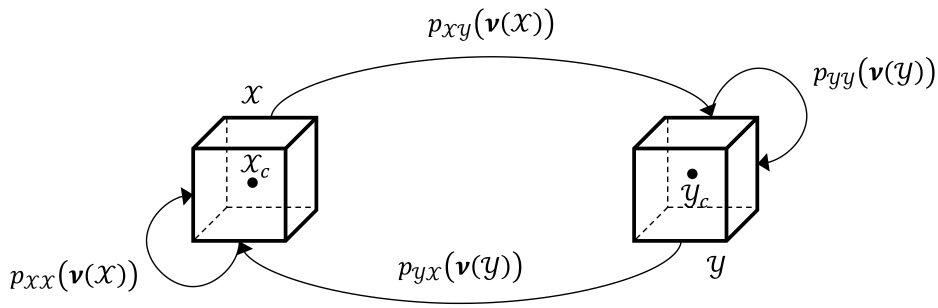


Figure 4.3: Transitions between aggregated states

i.e., $U(\mathcal{X}) = \{\tau_1, \tau_2, \ldots, \tau_l\}^r$.

Transition probabilities

Let the transition probabilities of moving from state $\mathcal{X}$ to $\mathcal{Y}$ under action $\mathbf{u}$ be denoted as $p_{\mathcal{X}\mathcal{Y}}(\mathbf{u})$. The transition probabilities may be approximated using the cumulative densities of the multivariate normal but this may lead to values that do not add up to 1. Hence, we first define $p'_{\mathcal{X}\mathcal{Y}}(\mathbf{u})$ as

$$p'_{\mathcal{X}\mathcal{Y}}(\mathbf{u}) = \int_{\mathbf{y}\in\mathcal{Y}} \frac{\exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\alpha}(\mathcal{X}_c, \mathbf{u}))^T \boldsymbol{\Sigma}(\mathcal{X}_c, \mathbf{u})^{-1}(\mathbf{y} - \boldsymbol{\alpha}(\mathcal{X}_c, \mathbf{u}))\right)}{\sqrt{(2\pi)^r \det \boldsymbol{\Sigma}(\mathcal{X}_c, \mathbf{u})}} d\mathbf{y} \quad (4.22)$$

where $\alpha(\mathcal{X}_c, \mathbf{u})$ and $\Sigma(\mathcal{X}_c, \mathbf{u})$ are defined as mentioned earlier. Next, we normalize these values by setting $p_{\mathcal{X}\mathcal{Y}}(\mathbf{u}) = p'_{\mathcal{X}\mathcal{Y}}(\mathbf{u}) / \sum_{\mathcal{Y}'\in\mathcal{S}} p'_{\mathcal{X}\mathcal{Y}'}(\mathbf{u})$.

Costs

The cost incurred in choosing $\mathbf{u}$ in state $\mathcal{X}$, $g(\mathcal{X}, \mathbf{u})$, is defined as $g(\mathcal{X}, \mathbf{u}) = \sum_{\mathcal{Y}\in\mathcal{S}} p_{\mathcal{X}\mathcal{Y}}(\mathbf{u}) \mathrm{TSTT}(\mathcal{Y}_c)$.

The objective for the approximate MDP is defined similarly as in Section 4.3.2. Let $J_{\boldsymbol{\nu}}(\mathcal{X})$ be the average cost per stage for policy $\boldsymbol{\nu}$ when the system starts at state $\mathcal{X}$. Assuming that $\mathcal{X}_k$ represents a state on the k^{th} day in the aggregated system,

$$J_{\boldsymbol{\nu}}(\mathcal{X}) = \lim_{K\to\infty} \frac{1}{K}\mathbb{E}\left\{\sum_{k=0}^{K-1} g\left(\mathcal{X}_k, \boldsymbol{\nu}(\mathcal{X}_k)\right) \,\middle|\, \mathcal{X}_0 = \mathcal{X}\right\} \tag{4.23}$$

where $\big(\boldsymbol{\nu}(\mathcal{X})\big)_{\mathcal{X}\in\mathcal{S}}$ specifies the action $\boldsymbol{\nu}(\mathcal{X}) \in U(\mathcal{X})$ to be taken when the system is in state $\mathcal{X}$. Let the optimal policy be denoted by $\boldsymbol{\nu}^*$, i.e., $J^*(\mathcal{X}) \equiv J_{\boldsymbol{\nu}^*}(\mathcal{X}) = \min_{\boldsymbol{\nu}\in\Phi} J_{\boldsymbol{\nu}}(\mathcal{X})$, where Φ is the set of all admissible policies. Since the state and action spaces are finite, Algorithm 6 can be applied to solve the approximate MDP.

Let $\Gamma : S \to \mathcal{S}$ be a mapping that gives the aggregated state to which a state in the original state space belongs (ties are broken arbitrarily). Then an approximate optimal policy for the original MDP can be defined as $\big(\boldsymbol{\mu}(\mathbf{x})\big)_{\mathbf{x}\in S}$, where $\boldsymbol{\mu}(\mathbf{x}) = \boldsymbol{\nu}^*\big(\Gamma(\mathbf{x})\big)$.

4.5 Demonstration

The method described in the previous section provides a policy that is optimal for the approximate MDP and an immediate question of interest is if it is close to the optimal policy for the original MDP. One possible way to answer this question is by tracking the errors involved. However, this is extremely difficult as we are making several approximations to the transition probabilities and in aggregating states. Instead, we resort to numerical experimentation to make claims about the approximate policy. While computing the optimal expected

TSTT for large n is difficult, for small values of n, we can use relative value iteration or other methods to exactly compute the optimal expected TSTT which can thus be used to ascertain how far we are from the optimal.

For any n, clearly the expected TSTT of the no-tolls case (or the do-nothing option) gives an upper bound to the optimal TSTT. Calculating the expected TSTT of a given policy is relatively easy and can be done by estimating the steady state distribution of the Markov chain under that policy or by simulation. Thus, we can estimate the expected TSTT under the approximate policy $\boldsymbol{\nu}^*(\Gamma(\mathbf{x}))$ and check if is an improvement over the no-tolls option, i.e., if it provides a better upper bound.

As n increases, the quality of approximations made to the state space and transition probabilities improves and depending on the available computing power, one can pick larger δ to develop finer partition schemes. Hence, we claim that this empirical line of analysis is proof enough that this method may be applied to problems with large state spaces.

For the numerical results presented in this chapter, we consider the network in Figure 4.4. Each traveler has three routes 1-2-4, 1-2-3-4, and 1-3-4. The link level travel times are assumed to be a function of the link flows and are shown in the figure. We assume that the set of possible tolls on each route can be enumerated as $\{0, 2, 4, \ldots, 8\}$, i.e., the action space for each state is $\{0, 2, 4, \ldots, 8\}^3$. The methods described were implemented in C++ (using the g++ compiler with -O2 optimization flags) on a Linux machine with 24

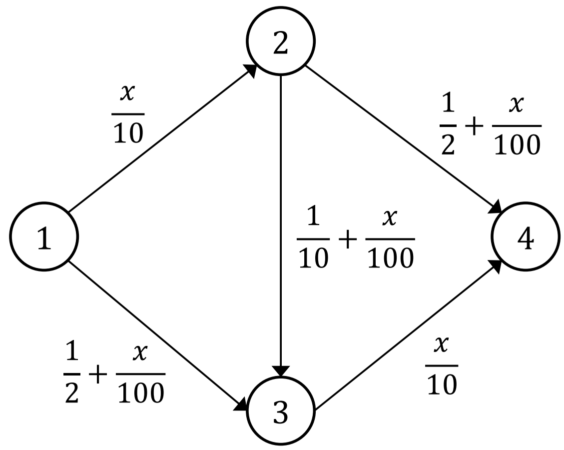


Figure 4.4: Network used to test the approximations

core Intel Xeon processors (3.33 GHz) and 12 MB cache. The cumulative densities for the multivariate normal distribution were obtained using a Fortran function available at `http://www.math.wsu.edu/faculty/genz/software/software.html`. The termination criterion for relative value iteration was set to 1E-07. The value of the dispersion parameter θ was fixed at 0.1. Most of the code was implemented in a parallel environment using OpenMP except for function calls to the Fortran code.

Solution quality

Table 4.1 compares the expected TSTT of the optimal and approximate policies for different number of travelers. It can be observed that the approximate policies perform better than the no-tolls option and the quality of approximation gets better with increase in δ. In all four cases, the approximate policies were found to be optimal or nearly optimal for large δ.

Table 4.1: Comparison on expected TSTT of policies

n	*Optimal exp. TSTT*	*Exp. TSTT of approx. policy* $\delta = 5$	$\delta = 10$	$\delta = 20$	*No-tolls exp. TSTT*
50	200.012	200.012	200.012	200.012	233.966
100	720.506	746.378	720.520	720.510	830.267
150	1532.220	1658.040	1532.930	1532.530	1730.410
200	2618.180	2881.280	2619.450	2618.750	2932.050

Computational performance

Table 4.2 indicates the wall-clock time in seconds for the steps involved in solving the exact and approximate MDPs. As mentioned earlier, the algorithms were implemented in a parallel environment with 24 cores except for the computation of transition probabilities of the approximate MDPs. Since these probabilities can be computed independently of each other, one can expect near linear speedup if implemented in a parallel manner. Notice that for $n = 100$, the value iteration for the exact method takes nearly 50 minutes and on the other hand the approximate methods provide near optimal solutions within a few seconds. For the exact MDP, when $n = 150$ and $n = 200$, the memory requirements for storing the transition probabilities exceeded available computing resources and hence they were not stored but were recomputed within each value iteration step. The run times for these instances were around 3 to 5 hours and have been left out of Table 4.2 in order to provide a fair comparison.

The results appear promising and for problems with much larger n, we may choose a value of δ according to the available computational resources. As the quality of the approximations get better with increase in n, the approximate

Table 4.2: Wall-clock times (in seconds) for exact and approximate methods

Number of Travelers →		50	100	150	200
Exact MDP	*No. of states*	1326	5151	11476	20301
	State space	0.023	0.193	0.561	1.316
	Trans prob	5.370	229.614	-	-
	Value itn	2.597	3056.490		
Approx. MDP ($\delta = 5$)	*No. of states*	25	25	25	25
	State space	5.80E-05	7.40E-05	5.70E-05	6.00E-05
	Trans prob	293.713	198.249	145.295	103.442
	Value itn	0.210	1.558	0.013	0.040
Approx. MDP ($\delta = 10$)	*No. of states*	100	100	100	100
	State space	5.13E-04	3.69E-04	3.36E-04	3.04E-04
	Trans prob	1351.90	1785.57	1461.96	1163.03
	Value itn	0.189	0.346	0.388	0.016
Approx. MDP ($\delta = 20$)	*No. of states*	400	400	400	400
	State space	3.79E-03	3.77E-03	2.60E-03	2.53E-03
	Trans prob	9538.21	9505.30	9606.38	9733.46
	Value itn	0.377	0.451	0.243	0.425

policies can be expected to perform better than the no-toll policy.

Mixing times

The Markov chain associated with any given policy is irreducible and aperiodic and hence has a steady state distribution. However, it takes a certain amount of time for the Markov chain to get "close" to its the stationary distribution. While, this is not a major concern for the average cost MDP, from a theoretical perspective, since we let $k \to \infty$, it would be useful to know how long it takes the Markov chain to reach its stationary distribution from a practical standpoint.

This question can be answered by analyzing the mixing time of the Markov chain associated with the optimal policy. In order to do so, a few definitions are in order. Let $\|\cdot\|_{TV}$ represent the total variation distance, which is a measure of the distance between two probability distributions. For any two probability density functions π and π', the total variation distance is defined as $\|\pi - \pi'\|_{TV} = \max_{A \subset S} |\pi(A) - \pi'(A)|$. Further, if S is discrete, it can be shown that $\|\pi - \pi'\|_{TV} = \frac{1}{2} \max_{\mathbf{x} \in S} |\pi(\mathbf{x}) - \pi'(\mathbf{x})|$ (Levin et al., 2009).

Now let P represent the transition probability matrix associated with the optimal policy $\boldsymbol{\mu}^*$, i.e., $P(\mathbf{x}, \mathbf{y}) = p_{\mathbf{xy}}(\boldsymbol{\mu}^*(\mathbf{x}))$ and let $\pi(\mathbf{x})$ denote the steady state probability of observing state $\mathbf{x}$. We define the maximum variation distance $d(k)$ between the rows of P and π after k steps as follows:

$$d(k) = \max_{\mathbf{x} \in S} \|P^k(\mathbf{x}, \cdot) - \pi(\cdot)\|_{TV} \tag{4.24}$$

Since a steady state distribution exists, $d(k) \to 0$ as $k \to \infty$. Further, for irreducible and aperiodic Markov chains, the rate at which $d(k)$ shrinks to zero is exponential and is bounded below by $\frac{1}{2}(1-\gamma)^k$, where γ is the absolute spectral gap which equals one minus the second largest magnitude eigenvalue (Montenegro and Tetali, 2006). Thus, the higher the value of gamma, the faster the rate at which the Markov chain converges to its steady state distribution. Another way to analyze the mixing times is by observing the least number of time steps before $d(k)$ falls below an arbitrary threshold ϵ.

$$t_{mix}(\epsilon) = \min\{k : d(k) \leq \epsilon\} \tag{4.25}$$

Table 4.3: Spectral gap and mixing times for different problem instances

Number of travelers (n)	*Spectral gap* (γ)	*Mixing time* ($t_{mix}(0.01)$)
50	0.767	3
100	0.530	6
150	0.491	7
200	0.560	6

Table 4.3 shows the spectral gap and mixing times for the Markov chains associated with the optimal policy for the four cases tested earlier. The spectral gap is not close to 0 and hence the Markov chains mix fairly quickly.

Finite Markov chains are often known to abruptly convergence to their stationary distributions. This feature, also known as the cutoff phenomena (Diaconis, 1996; Chen, 2006), results in a sudden drop in the $d(k)$ values. Figure 4.5 shows the maximum variation distance across consecutive days for different problem instances. While no evidence of the cutoff phenomenon was found, it may be interesting to see if such phenomena occur in problems with larger state spaces. We, however, observed that the mixing times of Markov chains increase with increase in the size of the state space.

4.6 Discussion

In this section, we first present some other objectives that may be of interest to a system manager or a private tolling agency. We then conclude by summarizing the methods proposed in this chapter.

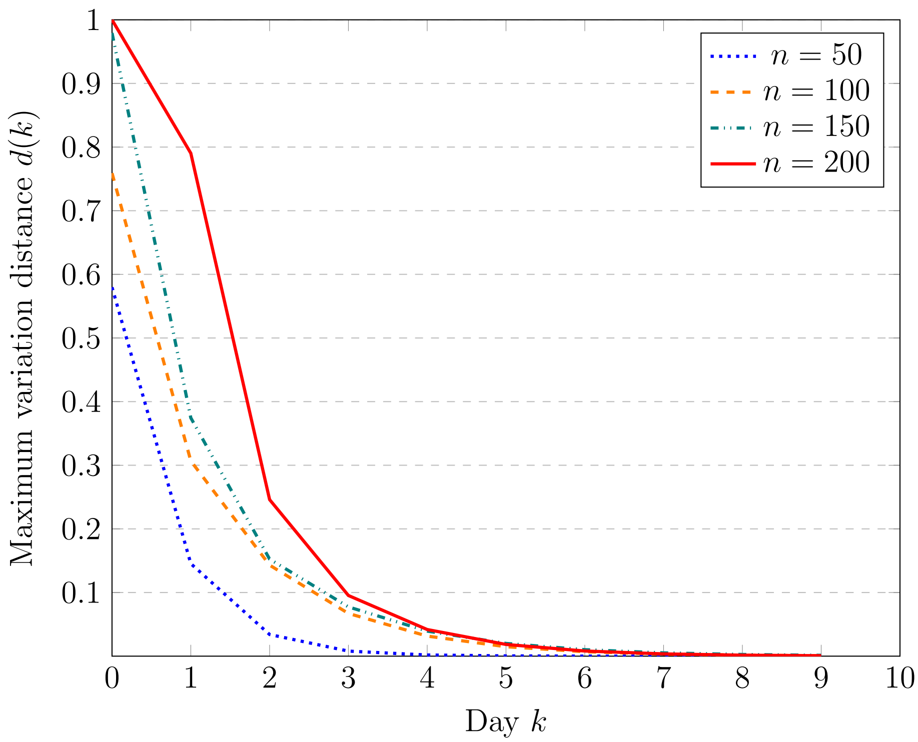


Figure 4.5: Variation distance of Markov chains associated with the optimal policy

4.6.1 Variants

Other network-wide objectives

By defining the average costs/rewards for a state-action pair differently we can find dynamic pricing policies that optimize other objectives.

Maximizing the probability of convergence to a target state: Suppose we wish to increase the probability of finding the system in a particular state in the long run. We will henceforth refer to this state as the *target state*. While a SO flow solution is an obvious choice for the target state, one could think of other target states based on other network wide objectives such as emissions. Also, in the presence of multiple NE solutions, one equilibrium may be favored over another and be chosen as the target state. In order to achieve this objective, we define the *rewards* (instead of costs) as follows:

$$g(\mathbf{x}, \mathbf{u}) = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is the target state} \\ 0 \text{ otherwise} \end{cases} \tag{4.26}$$

Thus, every time the system leaves the target state we receive a reward of 1 unit and therefore the long run probability of being in the target state is the average reward per stage. Instead, if we want to increase the probability of finding the system in set of states (i.e., we have a set of target states), we could just set $g(\mathbf{x}, \mathbf{u}) = 1$ for all $\mathbf{x}$ that belong to such a set and 0 otherwise.

One can think of extensions to these problems in which tolls are disallowed at the target state, i.e., if the network is at the target state on a particular day, then the system manager may choose to not collect any tolls for the next day. This mechanism has the same flavor as that of punishment strategies in repeated games that are used to force players to cooperate. This feature can easily be modeled by setting the action space at the target state to the empty set. The objective is likely to be lower when tolls are disallowed. Yet, by pricing all states except the target state, this formulation can lead to an increase in the probability of reaching the target state.

Minimizing the expected deviation from TSTT($\mathbf{x}_{SO}$): Suppose we want to minimize the deviation from the TSTT of the SO state. This objective could be useful in the context of improving travel time reliability as it can help reduce the variance in travel times and may be achieved by defining the stage costs as follows:

$$g(\mathbf{x}, \mathbf{u}) = (\mathrm{TSTT}(\mathbf{x}) - \mathrm{TSTT}(\mathbf{x}_{SO}))^2 \, \forall \, \mathbf{u} \in U(\mathbf{x}) \tag{4.27}$$

Incentives and revenue maximization

Assume that the system manager can incentivize travelers in addition to collecting tolls. Suppose that we model incentives as negative tolls. The optimal policy in such cases may require the system manager to pay something to travelers on an average. We can avoid this by adding side constraints (4.31) to the LP model (see Bertsekas (2007) for details) for the average cost MDP as shown below. Let $b(\mathbf{x}, \mathbf{u})$ be the expected revenue/cost for the system manager when $\mathbf{u}$ is chosen at state $\mathbf{x}$. The idea behind adding the side constraints is similar to budget balance mechanisms that are studied in mechanism design in which payment schemes that ensure zero net payments to all players are sought.

$$\lambda^* = \min \sum_{\mathbf{x}\in S} \sum_{\mathbf{u}\in U(\mathbf{x})} z(\mathbf{x},\mathbf{u}) g(\mathbf{x},\mathbf{u}) \tag{4.28}$$

$$\text{s.t.} \sum_{\mathbf{u}\in U(\mathbf{y})} z(\mathbf{y},\mathbf{u}) = \sum_{\mathbf{x}\in S} \sum_{\mathbf{u}\in U(\mathbf{x})} z(\mathbf{x},\mathbf{u}) p_{\mathbf{xy}}(\mathbf{u}) \ \ \forall\, \mathbf{y} \in S \tag{4.29}$$

$$\sum_{\mathbf{x}\in S} \sum_{\mathbf{u}\in U(\mathbf{x})} z(\mathbf{x},\mathbf{u}) = 1 \tag{4.30}$$

$$\sum_{\mathbf{x}\in S} \sum_{\mathbf{u}\in U(\mathbf{x})} z(\mathbf{x},\mathbf{u}) b(\mathbf{x},\mathbf{u}) \geq 0 \tag{4.31}$$

$$z(\mathbf{x},\mathbf{u}) \geq 0 \ \ \forall\, \mathbf{x} \in S, \mathbf{u} \in U(\mathbf{x}) \tag{4.32}$$

In the above LP model, equation (4.29) represents the balance equations and equation (4.30) is the normalization constraint. The optimal values of $z^*(\mathbf{x},\mathbf{u})$ can be used to construct the steady state distribution and the optimal policy. More precisely, for ergodic MDPs, for every state $\mathbf{x} \in S$, there exists exactly one $\mathbf{u} \in U(\mathbf{x})$ for which $z^*(\mathbf{x},\mathbf{u}) > 0$. Setting $\boldsymbol{\mu}^*(x)$ to $\mathbf{u} \in U(\mathbf{x})$ for which $z^*(\mathbf{x},\mathbf{u}) > 0$ gives the optimal policy. Further, $z^*(\mathbf{x}, \boldsymbol{\mu}^*(x))$ denotes the steady state probability of finding the system in state $\mathbf{x}$ under the optimal policy. Hence, the objective represents the expected TSTT and the left hand side of constraint (4.31) computes the expected revenue/cost.

However LP models are well suited for problems with small state and action spaces. We may therefore use the approximation methods developed in Section 4.4 and formulate the LP using the aggregated state space. Also, since travelers do not perceive incentives and tolls the same way, one can use prospect theory (Kahneman and Tversky, 1979) to distinguish between these.

In addition, if the system manager wishes to achieve a certain target revenue (which could in turn be used for maintaining the tolling infrastructure), we can set the right hand side of constraint (4.31) to the expected profit or target value.

4.6.2 Summary

In this chapter, we studied a dynamic day-to-day pricing model that can help a system manager minimize the expected TSTT. Specifically, we formulated the problem as an infinite horizon average cost MDP which provides stationary policies that are a function of the state of the system. Since practical problems involve a large number of travelers and exponential state spaces, we proposed approximate solution methods and performed a few numerical experiments to test their quality.

The results indicate that (1) for a large number of travelers, the approximate policies obtained by state space aggregation methods result in a significant reduction of expected TSTT compared to the no-toll case and (2) the computation times for obtaining an approximate optimal policy are reduced to a tractable degree after aggregating states. In the next chapter, we exploring scenarios which relax some of the assumptions made in this chapter and compute pricing policies using reinforcement learning techniques.

Chapter 5

Day-to-Day Pricing: Inferred Route Choice Dynamics

5.1 Introduction

In the previous chapter, the dynamic pricing problem in a day-to-day setting was described and formulated as an average cost MDP. A few key assumptions allowed us to compute the optimal toll policy. First, we assumed that all travelers have the same value of time. Second, travelers' day-to-day decisions were assumed to fit a route choice dynamic (such as the logit model) known to the system manager.

In practice, both these assumptions may not hold. Literature on congestion pricing has explored the possibility of differences in travelers' values of time by assuming a distribution for VOT (Dial, 1999b,c). It is also possible that travelers' decisions are not accurately captured by a model with closed form expressions for path choice probabilities such as the logit and C-logit model (Koppelman and Wen, 2000), but other models such as the probit may fit the data better. Furthermore, instead of using a single route choice model, the choices of different travelers may be better represented using different models and it may be hard for a planner to guess what model(s) to use for each traveler.

In such cases, model-free MDP approaches may be used which do not need the explicit knowledge of the underlying transition probabilities but infers or simulates them while computing the optimal policy simultaneously. Model-free methods appear in two flavors – off-line and online. In the off-line version, we develop a simulator that mimics travelers' choices and when these choices are aggregated, a sample future state is obtained. On the other hand, online methods gather information on future states from real world experiments after an action is taken at the current state. In both cases, however, we need assumptions to ensure that the MDP is ergodic. Hence, we continue to suppose that travelers' route choice processes places positive weight on every path for every toll vector so as to ensure that the Markov chain induced by every stationary policy is irreducible.

5.2 Q-Learning for Average Cost MDPs

Reinforcement learning methods have been of interest to the optimization and computer science community for over two decades. Q-learning is one such model-free approach that was first proposed for discounted cost MDPs (Watkins, 1989; Watkins and Dayan, 1992). Several researchers extended it to the average cost MDP problem and performed empirical analysis (Schwartz, 1993; Singh, 1994; Mahadevan, 1996). Abounadi et al. (2001) and Gosavi (2004) proposed update rules that were shown to theoretically convergence to the optimal values using ODE methods. In this thesis, we use the relative value iteration Q-learning algorithm proposed by Abounadi et al. (2001). Recall that the Bellman's equation for the average cost MDP problem takes the form

$$\lambda^* + h^*(\mathbf{x}) = \min_{\mathbf{u}\in U(\mathbf{x})} \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h^*(\mathbf{y}) \right\} \ \forall\ \mathbf{x} \in S \tag{5.1}$$

Define Q-factors $Q(\mathbf{x},\mathbf{u})$ for all state-action pairs as follows

$$Q(\mathbf{x},\mathbf{u}) = \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) h^*(\mathbf{y}) \right\} - \lambda^* \tag{5.2}$$

Notice that from the above two equations, $h^*(\mathbf{x}) = \min_{\mathbf{u}\in U(\mathbf{x})} Q(\mathbf{x},\mathbf{u})$. Re-writing Bellman's equations in terms of the Q-factors,

$$Q(\mathbf{x},\mathbf{u}) = \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) \min_{\mathbf{u}'\in U(\mathbf{y})} Q(\mathbf{y},\mathbf{u}') \right\} - \lambda^* \tag{5.3}$$

The relative value iteration algorithm that was discussed in Chapter 4 can now be reformulated using Q-factors. The update rule for such an algorithm would be

$$Q^{k+1}(\mathbf{x},\mathbf{u}) = \left\{ g(\mathbf{x},\mathbf{u}) + \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u}) \min_{\mathbf{u}'\in U(\mathbf{y})} Q^k(\mathbf{y},\mathbf{u}') \right\} - Q^k(\mathbf{s},\mathbf{v}), \tag{5.4}$$

where $(\mathbf{s},\mathbf{v})$ is a state-action pair that is arbitrarily chosen.

Equation (5.4) still contains the transition probability term $p_{\mathbf{xy}}(\mathbf{u})$ which may be unknown to the system manager. In order to derive an update rule without $p_{\mathbf{xy}}(\mathbf{u})$, we sample the state to which the system goes to when an action $\mathbf{u}$ is taken in state $\mathbf{x}$ using a simulator or data from a real world experiment.

The intuition behind Q-learning, which is briefly discussed next, can be explained using Robbins and Monro (1951)'s method for estimating the mean of a random variable from observations or samples (see Gosavi, 2014).

Consider a random variable X. Suppose we wish to estimate the expected value of the random variable $\mathbb{E}[X]$ in an iterative manner by collecting samples $x_0, x_1, \ldots, x_k, \ldots$. Define, $X_k = \frac{1}{k}\sum_{i=0}^{k-1} x_i$. Using the strong law of large numbers, $\mathbb{E}[X] = \lim_{k\to\infty} X_k$. Therefore,

$$\begin{aligned} X_{k+1} &= \frac{1}{k+1}\sum_{i=0}^{k} x_i & (5.5)\\ &= \frac{k}{k+1}X_k + \frac{1}{k+1}x_k & (5.6)\\ &= \Big(1 - \frac{1}{k+1}\Big)X_k + \frac{1}{k+1}x_k & (5.7) \end{aligned}$$

Let us now rewrite equation (5.4) in the following manner.

$$Q^{k+1}(\mathbf{x},\mathbf{u}) = \sum_{\mathbf{y}\in S} p_{\mathbf{xy}}(\mathbf{u})\Big\{\min_{\mathbf{u}'\in U(\mathbf{y})} Q^k(\mathbf{y},\mathbf{u}') + g(\mathbf{x},\mathbf{u}) - Q^k(\mathbf{s},\mathbf{v})\Big\} \quad (5.8)$$

For a fixed $(\mathbf{x},\mathbf{u})$ pair, the right hand side computes the expected value of some random variable with support S and the term inside the curly braces can be treated as a realization or a sample (analogous to x_k in equation (5.7) of Robbins and Monro's algorithm). Thus, in order to update the Q-factors, we draw a sample future state $\xi(\mathbf{x},\mathbf{u})$ after choosing $\mathbf{u}$ in state $\mathbf{x}$ and use the following equation.

$$Q^{k+1}(\mathbf{x},\mathbf{u}) = \big(1-\gamma_k\big)Q^k(\mathbf{x},\mathbf{u}) + \gamma_k\bigg(\min_{\mathbf{u}'} Q^k\big(\xi(\mathbf{x},\mathbf{u}),\mathbf{u}'\big) + g(\mathbf{x},\mathbf{u}) - Q^k(\mathbf{s},\mathbf{v})\bigg) \quad (5.9)$$

where γ_k is a diminishing sequence of step sizes satisfying $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$ (e.g., $\gamma_k = 1/(k+1)$).

Equation (5.9) represents the synchronous Q-learning algorithm (Abounadi et al., 2001) in which at each iteration k, the Q-factors for all state-action pairs $(\mathbf{x}, \mathbf{u})$ are updated. This Q-learning version is more useful in an off-line setting in which the state transitions are complicated to be expressed analytically but can be simulated. For the dynamic pricing framework proposed in this chapter, the synchronous Q-learning algorithm can be used in the following scenarios:

- Travelers' choices are be represented by a closed form route choice model such as the logit model, but on each day, a fixed number of users sampled from a population of agents having a VOT distribution decide to travel.
- Travelers' choices are better described using other discrete choice models such as the probit model. In fact, since an individual's probability of selecting routes is being simulated, one could relax the usual assumptions on the independence of error terms in the utility expressions and instead assume that they are correlated. Capturing correlations between unobserved factors such as number of signals or left turns along a route could lead to models with better fit.

However, if the only source of information on travelers' route choices is transitions observed from the field over different days, an online asynchronous Q-learning algorithm which is also known to converge asymptotically can be used. In the asynchronous version, the system manager observes a state and picks a toll vector from the action space randomly (usually with uniform probability) and observes the state to which the system moves on the next day. The Q-values are then updated and another toll vector is randomly picked for the next day and the process is repeated.

When implementing the asynchronous version, one must realize that the Q-factors are not updated for all state-action pairs for the same number of times. In order to guarantee convergence, it is necessary that all state-action pairs are visited infinitely often. Hence, when in state $\mathbf{x}$, we choose $\mathbf{u}$ from $U(\mathbf{x})$ with probability $1/|U(\mathbf{x})|$. The update step for asynchronous Q-learning (see Algorithm 7) is very similar to that of the synchronous version except for the step sizes which take into account the number of times the Q-factors were updated for a state-action pair (represented using $\eta(\mathbf{x}, \mathbf{u})$). The Q-values are updated for k_{max} iterations after which Step 3 is used to construct a policy that is used at the end of the learning period.

While Q-learning does not require the knowledge of the transition probabilities, its major drawback is that the convergence is asymptotic and requires sufficient exploration of the state and action spaces. Hence, the number of iterations required to learn the optimal strategy are large and are proportional to the number of state-action pairs. Synchronous Q-learning, however, uses an off-line method to find a policy and also does not compute and store the transition probabilities and hence can be a very effective method to solve the day-to-day

Algorithm 7 *Pseudocode for asynchronous Q-learning*

Step 1:
$k \leftarrow 0$
On day 0, suppose the state of the system is $\mathbf{x}_0$
Arbitrarily select a state-action pair $(\mathbf{s}, \mathbf{v})$
$Q(\mathbf{x}, \mathbf{u}) \leftarrow 0\, \forall\, \mathbf{x} \in S, \mathbf{u} \in U(\mathbf{x})$ (they could be initialized to arbitrary values)
$\eta(\mathbf{x}, \mathbf{u}) \leftarrow 0\, \forall\, \mathbf{x} \in S, \mathbf{u} \in U(\mathbf{x})$

Step 2:
while $k < k_{max}$ **do**
  Select $\mathbf{u}_k$ randomly from $U(\mathbf{x}_k)$ with uniform probability
  $$Q(\mathbf{x}_k, \mathbf{u}_k) \leftarrow \big(1 - \gamma_{\eta(\mathbf{x}_k, \mathbf{u}_k)}\big) Q(\mathbf{x}_k, \mathbf{u}_k) + \gamma_{\eta(\mathbf{x}_k, \mathbf{u}_k)} \Bigg(\min_{\mathbf{u}'} Q\big(\xi(\mathbf{x}_k, \mathbf{u}_k), \mathbf{u}'\big) + g(\mathbf{x}_k, \mathbf{u}_k) - Q(\mathbf{s}, \mathbf{v}) \Bigg)$$
  $\eta(\mathbf{x}_k, \mathbf{u}_k) \leftarrow \eta(\mathbf{x}_k, \mathbf{u}_k) + 1$
  $\mathbf{x}_k \leftarrow \xi(\mathbf{x}_k, \mathbf{u}_k)$
  $k \leftarrow k + 1$
end while

Step 3:
Construct the Q-learning policy $\boldsymbol{\mu}(\mathbf{x}) \in \arg\min_{\mathbf{u} \in U(\mathbf{x})} Q(\mathbf{x}, \mathbf{u})$

dynamic pricing problem. Asynchronous methods on the other hand work only when the state and action spaces are small. In order to circumvent this issue, we work with an aggregated state space as described in the previous chapter.

5.3 Demonstration

In this section, we apply Q-learning to the Braess network with three routes that was introduced in Figure 4.4. Since Q-learning is not well suited for problems with large state-action pairs, we restrict our action space to $\{0, 4\}^3$, i.e., we either collect a toll of 0 or 4 units on each of the three routes. A demand of 100 travelers is assumed between node 1 and 4. The expected TSTT for the no-tolls case is 830.27 and the optimal policy yields an expected TSTT of 767.61.

In order to simulate the route choices of travelers, we assume that the travelers follow the logit model with a unit value of time. Note that transition probabilities are not directly used by the system manager and are only used to draw a sample future state when a toll vector is chosen at the current state. Although different values of time and other route choice models could be used for the purpose of simulating the evolution of the system, the logit model was chosen because we can compute the exact optimal solution and benchmark the performance of the Q-learning algorithms.

The values of the transition probabilities for the aggregated state space are obtained directly using equation 4.4 and are not derived using the multivariate

Gaussian distribution. In this process, we make a minor modification to the aggregated state space and define it as $\mathcal{S} = \{\mathcal{X} \in \mathcal{I}^r : \mathcal{X} \cap S \neq \emptyset\}$ to ensure that at each aggregated state contains at least one original state (a feasible integral flow solution). As before, let the transition probabilities of moving from state $\mathcal{X}$ to $\mathcal{Y}$ under action $\mathbf{u}$ be denoted as $p_{\mathcal{X}\mathcal{Y}}(\mathbf{u})$. The transition probabilities for the aggregated state space may be expressed in terms of the transition probabilities of the original state space in the following manner.

$$p_{\mathcal{X}\mathcal{Y}}(\mathbf{u}) = \frac{\sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathbf{y} \in \mathcal{Y}} p_{\mathbf{x}\mathbf{y}}(\mathbf{u})}{\sum_{\mathbf{x}' \in \mathcal{X}} \sum_{\mathbf{y}' \in S} p_{\mathbf{x}'\mathbf{y}'}(\mathbf{u})} \tag{5.10}$$

5.3.1 Synchronous Q-learning

The synchronous Q-learning algorithm is suited for instances in which traveler choices may be simulated using complicated route choice models or when the VOT is not constant across the population. In order to test the performance of this algorithm, we first construct a policy using the Q-factors obtained after k_{max} iterations. Given an aggregated state, this policy prescribes an action to be taken when in that state. Note that the same action is taken for all states in the original state space that belong to the aggregated state. In this way, one can define a Markov chain on the original state space. Finally, the transition probabilities of the Markov chain (which are known from the logit choice model) are used to estimate the expected TSTT for the Q-learning policy.

Since the actions at every state in each iteration leads to a future state that

Table 5.1: Results of synchronous Q-learning

k_{max}	$\delta = 4$	$\delta = 8$
5	(794.8,805.5)	(809.5,815.7)
10	(779.3,787.7)	(800.0,804.7)
20	(771.5,776.5)	(790.5,794.6)
30	(771.1,776.5)	(789.7,793.9)

is random, the policy obtained at the end of k_{max} iterations need not be the same if the experiment is repeated. Hence, in order to compare the solution quality with the TSTT of the optimal solution or the no-tolls case, we run the Q-learning algorithm multiple times (100 for the results in this chapter) and construct a confidence interval for the expected TSTT associated with the Q-learning policy. Table 5.1 shows the 95% confidence interval for different learning periods and two levels of state space aggregation.

The mean estimate of expected TSTT is plotted in Figure 5.1. Notice that the algorithm converges to the optimal solution as k_{max} increases and does better than the no-tolls case even if terminated after a few iterations. However, the results for the $\delta = 4$ case were superior to that for $\delta = 8$ probably because the latter case involved more states, and hence learning the transition probabilities along with the optimal policy might have been difficult.

5.3.2 Asynchronous Q-learning

The asynchronous Q-learning algorithm described in Algorithm 7 was tested for k_{max} ranging from 10^3 to 10^5. The expected TSTT of the policy obtained

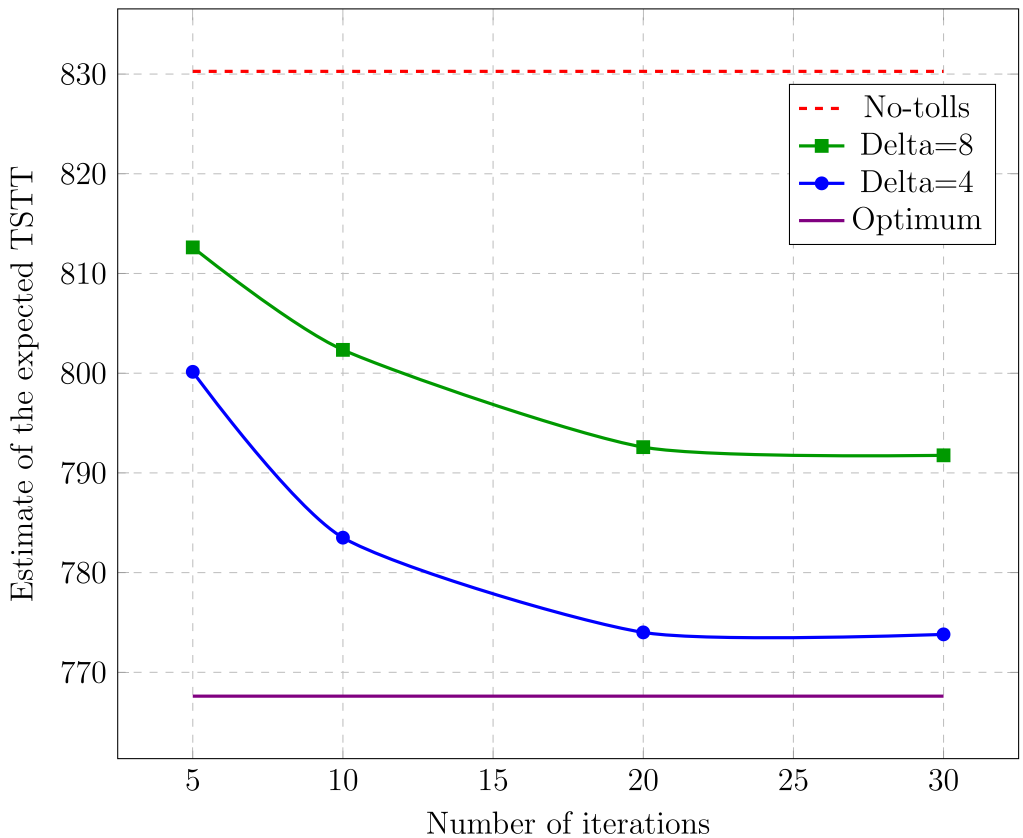


Figure 5.1: Expected TSTT for of the synchronous Q-learning policy.

at the end of the learning period is evaluated as explained in Section 5.3.1. Note again that policies from asynchronous Q-learning are dependent on the initial state and the sample path followed up to day k_{max}. Hence, we construct confidence intervals as before with random initial conditions (see Table 5.2)

Figure 5.2 shows the expected TSTT for varying levels of δ averaged over different runs. Unlike the results presented in the previous chapter, the solution quality does not necessarily improve as we move from a coarser to finer level of aggregation. The plot indicates that the Q-learning algorithm discovers a policy that is close to the optimal only for aggregation levels of δ equal to 4 and 5. In fact, in some cases, the resulting policy can do worse than the do-nothing (no-tolls) case. The reason for this behavior is straightforward. As δ increases, the learning algorithm has to deal with a large number of state-action pairs which makes it difficult for it to learn the optimal policy. At the same time, having a very small δ induces errors as we are forced to take the same action at all route flow patterns belonging to an aggregated state. Hence, there is a trade-off between possibility of discovering an optimal solution and difficulty associated with learning it.

We also studied the performance of the Q-learning policy by varying the number of iterations for the learning period. Table 5.3 shows the 95% confidence intervals for the expected TSTT of the Q-learning policy for different values of k_{max}. Two levels of aggregation $\delta = 4$ and $\delta = 8$ were considered.

Figure 5.3 shows the sample mean of the expected TSTT as k_{max} increases.

Table 5.2: 95% confidence intervals for expected TSTT of the Q-learning policy for different levels of aggregation.

δ	$k_{max} = 2000$	$k_{max} = 40000$	$k_{max} = 100000$
3	(866.8,874.8)	(876.0,882.0)	(888.7,888.7)
4	(810.2,819.5)	(771.8,776.7)	(769.6,770.4)
5	(808.1,816.5)	(796.4,804.9)	(777.9,785.3)
6	(836.8,844.2)	(808.2,815.7)	(802.5,810.0)
7	(842.2,846.9)	(831.8,835.6)	(831.1,836.1)
8	(845.1,850.0)	(832.1,835.2)	(828.1,831.5)
9	(842.3,846.7)	(832.2,836.0)	(827.1,830.8)
10	(843.7,848.3)	(834.7,837.4)	(831.5,834.6)

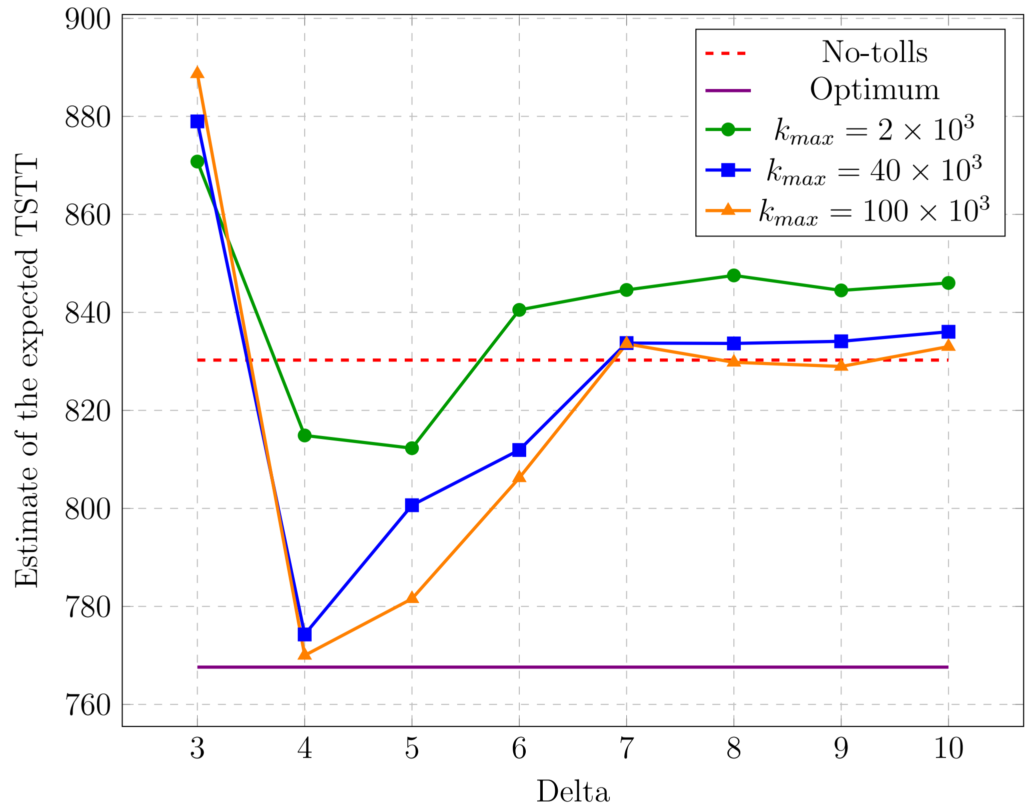


Figure 5.2: Expected TSTT for different levels of aggregation.

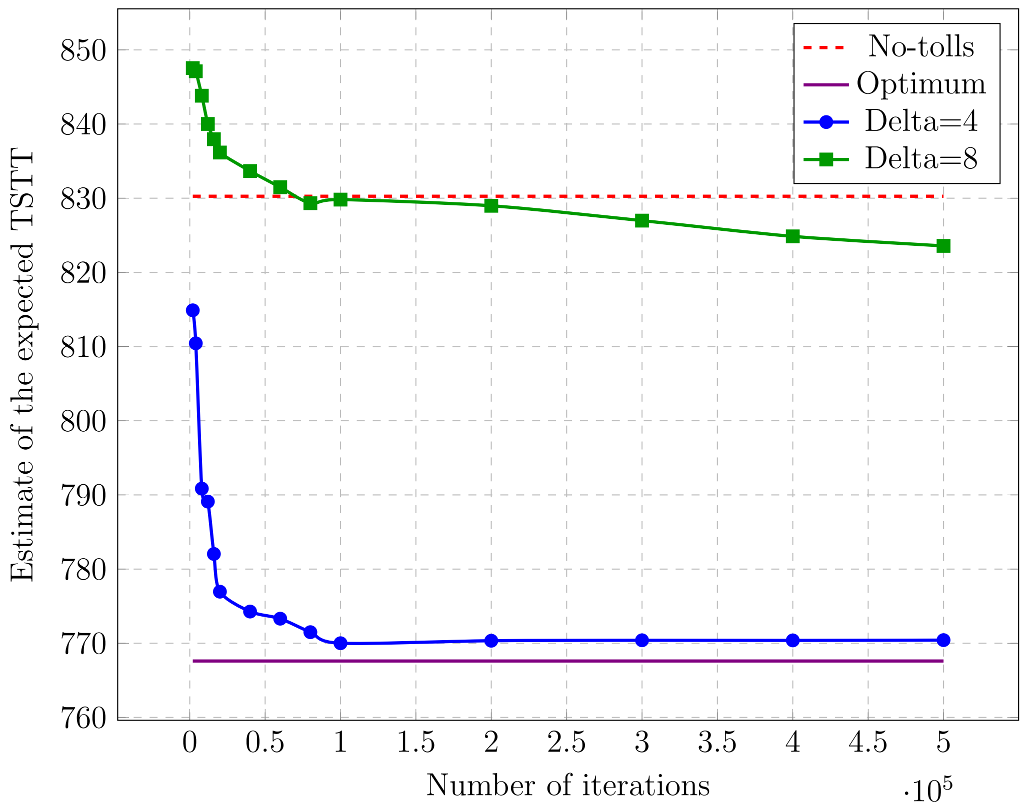


Figure 5.3: Expected TSTT for different learning periods.

Table 5.3: 95% confidence intervals for the expected TSTT of the Q-learning policy for different learning periods.

k_{max} (in '000)	$\delta = 4$	$\delta = 8$
2	(810.2,819.5)	(845.1,850.0)
4	(805.6,815.3)	(844.5,849.7)
8	(785.9,795.8)	(841.6,846.1)
12	(784.4,793.8)	(837.7,842.4)
16	(778.2,785.8)	(836.1,839.8)
20	(774.2,779.7)	(834.4,837.9)
40	(771.8,776.7)	(832.1,835.2)
60	(771.2,775.5)	(830.3,832.7)
80	(769.9,773.1)	(826.9,831.8)
100	(769.6,770.4)	(828.1,831.5)
200	(770.0,770.7)	(828.2,829.8)
300	(770.3,770.5)	(825.5,828.5)
400	(770.3,770.4)	(823.0,826.8)
500	(770.4,770.4)	(821.5,825.7)

As expected, the performance of the Q-learning policy is close to the optimal solution for larger k_{max}. When $\delta = 4$, the Q-learning policy performs better than the no-tolls case even for lower values of k_{max}. However, as k_{max} increases, it can only provide a solution with an expected TSTT equal to 770.424 whereas the optimal solution is 767.61. This difference is a result of the error induced due to aggregation which may be reduced using a finer aggregation scheme at the cost of longer learning periods. For instance, when $\delta = 8$, the algorithm can find an optimal policy with an expected TSTT of 767.61 but only after about 10^7 iterations.

5.4 Summary

In this chapter, we addressed the problem of pricing when the system manager lacks explicit closed form knowledge of route choice mechanisms employed by users in a day-to-day setting. Standard Q-learning approaches for the average cost MDP were explained and demonstrated on a small network. Two types of algorithms were implemented. First, we discussed a synchronous Q-learning algorithm which could be of use when users' route choices are difficult to express mathematically but can be simulated or when users are sampled from a population with heterogeneous values of time. Since the policy is computed off-line, these methods can find the optimal policy without much difficulty. However, it is necessary to ensure that users' route choices are accurately captured by the simulation framework that is used to draw samples of future states.

The asynchronous version which can be applied in real-time was also tested. Although the assumptions in this learning model are least restrictive, we found that this algorithm is sensitive to the level of aggregation and may take a staggeringly large number of iterations before discovering the optimal policy. Thus, using this method can be quite challenging but may still be of use in small networks in which travelers choose between managed and general purpose lanes.

Chapter 6

Conclusion

6.1 Summary

Two dynamic congestion pricing models were developed in this dissertation. The first one, discussed in Chapters 2 and 3, addressed the externalities associated with non-recurring within-day congestion by setting tolls that depended on the network state. Travelers were assumed to be fully-rational and made online routing decisions to minimize their expected travel times. Supply-side uncertainty was modeled using probabilistic link performance functions and a static equilibrium with recourse model was formulated to understand the steady-state conditions. It was mathematically shown that state-dependent marginal tolls minimize the total expected travel times of all the users in the network. The sub-optimality of static tolls was demonstrated on the Sioux Falls test network using the Frank-Wolfe algorithm and a value iteration method was used to compute the optimal routing policies. Interestingly, in many cases, fixed tolls were found to perform worse than the no-tolls or the do-nothing scenario.

Like in deterministic TAPs, the set of tolls that leads to a socially optimum state is not unique. Hence, a minimum expected revenue pricing problem was formulated to compute an alternate optimal toll vector that also generates the least expected revenue. This problem was formulated as a linear program using data from the solution to a reformulation of the equilibrium with recourse model. Such solutions have the potential to not only minimize system inefficiency but can also make congestion pricing more acceptable.

In the second pricing framework, discussed in Chapters 4 and 5, the problem of setting dynamic tolls in day-to-day traffic models was studied. Traveler behavior was assumed to be captured using route choice models that minimize expected disutility. The uncertainty in the route choices are assumed to result from perception errors or unobserved factors. Tolls are revealed to the travelers on each day before they make their trips and are set using the state of the network on the previous day(s). Travelers' route choice probabilities are a function of historic travel times and the toll on the current day. The problem of setting dynamic state-dependent tolls was formulated as an average cost MDP with an aim to minimize the total expected system travel time over an infinite horizon. However, finding the optimal tolling policy was computationally expensive as one needs to enumerate the state space. Hence, approximate state space aggregation methods were suggested to improve tractability for small networks with a large number of travelers.

This dynamic tolling framework was first studied in a setting in which all users follow the logit choice model. Existence of a closed form route choice dynamic helped us formulate and compute the optimal tolling policy. We then extended this approach to instances in which route choices cannot be mathematically described but can either be simulated or observed from practice. Our findings indicate that it is efficient to first construct an appropriate route choice model and then compute the optimal tolling policy using an off-line simulator.

6.2 Future Work

The pricing models developed in this dissertation motivate several topics for future research. On the topic of state-dependent tolls under supply-side uncertainty in within-day traffic models, (1) the sub-optimality of static tolls call for accurately estimating supply-side variables and their distributions using historic incident and weather data. (2) Improving the run times using more advanced algorithms such as origin-based assignment and conjugate and bi-conjugate Frank-Wolfe can help compute marginal tolls for regional networks with more link-states and is another potential topic for exploration. (3) While travelers' actions were assumed to be conditioned on the downstream link-states, with V2X technologies, travelers will have access to reliable real-time state and toll information at a network level. Also, the current work assumes that the state of each link is independent of other link-states which may be restrictive depending on the type of disruption and the scale of the network. Thus, it would be worthwhile to develop more sophisticated policy-based routing and tolling models along the lines of this dissertation. (4) Also allowing only a fraction of travelers to replan can help model more practical scenarios that include both human and autonomous drivers. (5) Finally, one could explore methods such as Dantzig-Wolfe decomposition (Bai et al., 2004) for solving the minimum expected revenue linear programs more efficiently.

On the other hand, day-to-day pricing models can be extended to (1) larger networks in which threshold type tolling policies may be sought which are not necessarily optimal but perform better than the no-tolls scenario. (2) It would be of interest to explore situations in which dynamic tolls are levied from a subset of regular travelers or "members", while the remaining users pay a fixed but higher fee for choosing a route. This way, the route choices of the regular set of travelers can be influenced to improve system efficiency. (3) For models in which no route choice data is available and the optimal tolls are learnt from experimentation, it might be fruitful to investigate more promising reinforcement learning algorithms (Kearns and Singh, 2002; Brafman and Tennenholtz, 2002; Strehl et al., 2006) and other randomized schemes for selecting actions at each state, which may improve the rate of convergence by efficiently addressing the tradeoff between exploration and exploitation.

In both pricing models, the traffic flow component was simplified by assuming that the travel time on each link is a function of its flow or the number of travelers on it. Relaxing this assumption can improve traffic predictions and make the dynamic state-dependent pricing models more realistic.

Bibliography

Jinane Abounadi, D Bertsekas, and Vivek S Borkar. Learning algorithms for markov decisionprocesses with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

Lihui Bai, Donald W Hearn, and Siriphong Lawphongpanich. Decomposition techniques for the minimum toll revenue problem. *Networks*, 44(2):142–150, 2004.

Hillel Bar-Gera. Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 36(4):398–417, 2002.

Hillel Bar-Gera. Traffic assignment by paired alternative segments. *Transportation Research Part B*, 44(8–9):1022–1046, 2010.

Martin Beckmann, CB McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.

Pia Bergendorff, Donald W. Hearn, and Motakuri V. Ramana. *Network Optimization*, chapter Congestion Toll Pricing of Traffic Networks, pages 51–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 4th edition edition, 2007.

Dimitri P. Bertsekas, John N. Tsitsiklis, and Cynara Wu. Rollout algorithms for combinatorial optimization. *Journal of Heuristics*, 3(3):245–262, 1997.

Lawrence Blume. Population games. Game Theory and Information 9607001, EconWPA, July 1996.

Stephen D. Boyles. *Operational, Supply-Side Uncertainty in Transportation Networks: Causes, Effects, and Mitigation Strategies*. PhD thesis, The University of Texas at Austin, 2009.

Stephen D. Boyles and Tarun Rambha. A note on detecting unbounded instances of the online shortest path problem. *Networks*, 67(4):270–276, 2016.

Stephen D Boyles and S Travis Waller. A mean-variance model for the minimum cost flow problem with stochastic arc costs. *Networks*, 56(3):215–227, 2010.

Stephen D. Boyles, Kara M. Kockelman, and S. Travis Waller. Congestion pricing under operational, supply-side uncertainty. *Transportation Research Part C: Emerging Technologies*, 18(4):519 – 535, 2010.

Stephen D. Boyles, Shoupeng Tang, and Avinash Unnikrishnan. Parking search equilibrium on a network. *Transportation Research Part B: Methodological*, 81, Part 2:390 – 409, 2015.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

George W. Brown. Iterative solution of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 374–376. Wiley, New York, 1951.

George W. Brown and John Von Neumann. Solutions of games by differential equations. Technical report, DTIC Document, 1950.

Giulio E. Cantarella and Ennio Cascetta. Dynamic processes and equilibrium in transportation networks: towards a unifying theory. *Transportation Science*, 29(4):305–329, 1995.

Ennio Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research Part B: Methodological*, 23(1):1 – 17, 1989.

Ennio Cascetta and Giulio E. Cantarella. A day-to-day and within-day dynamic stochastic assignment model. *Transportation Research Part A: General*, 25(5):277 – 291, 1991.

Guan-Yu Chen. *The cutoff phenomenon for finite Markov chains*. PhD thesis, Cornell University, 2006.

Yi-Chang Chiu, Jon Bottom, Michael Mahut, Alex Paz, Ramachandran Balakrishna, Travis Waller, and Jim Hicks. A Primer for Dynamic Traffic Assignment, 2010. Prepared by the Transportation Network Modeling Committee of the Transportation Research Board (ADB30).

Stella C. Dafermos. Traffic equilibrium and variational inequalities. *Transportation Science*, 14(1):42–54, 1980.

Carlos F. Daganzo and Yosef Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11(3):253–274, 1977.

Gary A. Davis and Nancy L. Nihan. Large population approximations of a general stochastic traffic assignment model. *Operations Research*, 41(1): 169–178, 1993.

Daniela Pucci de Farias and Benjamin Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620, 2006.

Andr de Palma and Robin Lindsey. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies*, 19(6):1377 – 1399, 2011.

Persi Diaconis. The cutoff phenomenon in finite markov chains. *Proceedings of the National Academy of Sciences*, 93(4):1659–1664, 1996.

Robert B. Dial. A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, 5:83–111, 1971.

Robert B. Dial. Minimal-revenue congestion pricing part I: A fast algorithm for the single-origin case. *Transportation Research Part B*, 33:189–202, 1999a.

Robert B. Dial. Network-optimized road pricing: Part I: a parable and a model. *Operations Research*, 47(1):54–64, 1999b.

Robert B. Dial. Network-optimized road pricing: Part II: algorithms and examples. *Operations Research*, 47(2):327–336, 1999c.

Robert B. Dial. Minimal-revenue congestion pricing part II: An efficient algorithm for the general case. *Transportation Research Part B: Methodological*, 34(8):645 – 665, 2000.

Robert B. Dial. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B*, 40(10):917–936, 2006.

Liisa Ecola and Thomas Light. Equity and congestion pricing. *Rand Corporation*, pages 1–45, 2009.

Farhad Farokhi and Karl H. Johansson. A piecewise-constant congestion taxing policy for repeated routing games. *Transportation Research Part B: Methodological*, 78(0):123 – 143, 2015.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

Terry L. Friesz, David Bernstein, Nihal J. Mehta, Roger L. Tobin, and Saiid Ganjalizadeh. Day-to-day dynamic network disequilibria and idealized traveler information systems. *Operations Research*, 42(6):1120–1136, 1994.

Terry L. Friesz, David Bernstein, and Niko Kydes. Dynamic congestion pricing in disequilibrium. *Networks and Spatial Economics*, 4(2):181–202, 2004.

Song Gao. Modeling strategic route choice and real-time information impacts in stochastic and time-dependent networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1298–1311, 2012.

Abhijit Gosavi. Reinforcement learning for long-run average cost. *European Journal of Operational Research*, 155(3):654–674, 2004.

Abhijit Gosavi. *Simulation-based optimization: parametric optimization techniques and reinforcement learning*, volume 55. Springer, 2014.

Ren-Yong Guo, Hai Yang, and Hai-Jun Huang. A discrete rational adjustment process of link flows in traffic networks. *Transportation Research Part C: Emerging Technologies*, 34(0):121 – 137, 2013.

Ren-Yong Guo, Hai Yang, Hai-Jun Huang, and Zhijia Tan. Link-based day-to-day network traffic dynamics and equilibria. *Transportation Research Part B: Methodological*, 71(0):248 – 260, 2015.

Younes Hamdouch and Siriphong Lawphongpanich. Schedule-based transit assignment model with travel strategies and capacity constraints. *Transportation Research Part B: Methodological*, 42(78):663 – 684, 2008.

Younes Hamdouch and Siriphong Lawphongpanich. Congestion pricing for schedule-based transit networks. *Transportation Science*, 44(3):350–366, 2010.

Younes Hamdouch, Patrice Marcotte, and Sang Nguyen. A strategic model for dynamic traffic assignment. *Networks and Spatial Economics*, 4(3):291–315, 2004.

Younes Hamdouch, W.Y. Szeto, and Y. Jiang. A new schedule-based transit assignment model with travel strategies and supply uncertainties. *Transportation Research Part B: Methodological*, 67:35 – 67, 2014.

Lanshan Han and Lili Du. On a link-based day-to-day traffic assignment model. *Transportation Research Part B: Methodological*, 46(1):72 – 84, 2012.

Martin L. Hazelton and David P. Watling. Computation of equilibrium distributions of markov traffic-assignment models. *Transportation Science*, 38 (3):331–342, 2004.

Xiaozheng He, Xiaolei Guo, and Henry X. Liu. A link-based day-to-day traffic assignment model. *Transportation Research Part B: Methodological*, 44(4): 597 – 608, 2010.

Donald W. Hearn and Motakuri V. Ramana. *Solving Congestion Toll Pricing Models*, pages 109–124. Springer US, Boston, MA, 1998.

R. Jayakrishnan, Wei T. Tsai, Joseph N. Prashker, and Subodh Rajadhyaksha. A faster path-based algorithm for traffic assignment. *Transportation Research Record*, 1443:75–83, 1994.

Dusica Joksimovic, Michiel C.J. Bliemer, and Piet H.L. Bovy. Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. *Transportation Research Record: Journal of the Transportation Research Board*, 1923(1):61–72, 2005.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):pp. 263–292, 1979.

Michihiro Kandori and Rafael Rob. Evolution of equilibria in the long run: A general theory and applications. *Journal of Economic Theory*, 65(2): 383–414, 1995.

Michihiro Kandori, George J. Mailath, and Rafael Rob. Learning, mutation, and long run equilibria in games. *Econometrica*, 61(1):pp. 29–56, 1993.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Frank S. Koppelman and Chieh-Hua Wen. The paired combinatorial logit model: properties, estimation and application. *Transportation Research Part B: Methodological*, 34(2):75 – 89, 2000.

Vidyadhar G Kulkarni. *Modeling and analysis of stochastic systems*. CRC Press, 2009.

Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12):1608–1622, 1998.

Torbjörn Larsson and Michael Patriksson. Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science*, 26:4–17, 1992.

Fabien M Leurent. Curbing the computational difficulty of the logit equilibrium assignment model. *Transportation Research Part B: Methodological*, 31(4):315 – 326, 1997.

David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.

Jiaqi Ma, Brian L. Smith, and Xuesong Zhou. Personalized real-time traffic information provision: Agent-based optimization model and solution framework. *Transportation Research Part C: Emerging Technologies*, 64:164 – 182, 2016.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1-3):159–195, 1996.

Maria Mitradjieva and Per Olov Lindberg. The stiff is moving — conjugate direction Frank-Wolfe methods with application to traffic assignment. *Transportation Science*, 47(2):280–293, 2013.

Ravi R. Montenegro and Prasad Tetali. *Mathematical aspects of mixing times in Markov chains*. Now Publishers Inc, 2006.

Anna Nagurney and Ding Zhang. Projected dynamical systems in the formulation, stability analysis, and computation of fixed-demand traffic network equilibria. *Transportation Science*, 31(2):pp. 147–158, 1997.

Yu (Marco) Nie. A note on bar-gera's algorithm for the origin-based traffic assignment problem. *Transportation Science*, 46(1):27–38, 2012.

Michael Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

Srinivas Peeta and AthanasiosK. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3-4):233–265, 2001.

Claude M. Penchina. Minimal-revenue congestion pricing: some more good-news and bad-news. *Transportation Research Part B: Methodological*, 38(6): 559 – 570, 2004.

Arthur C. Pigou. *The Economics of Welfare*. Macmillan and Co., London, 1920.

George H. Polychronopoulos and John N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27(2):133–143, 1996.

J. Scott Provan. A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2):115–125, 2003.

Martin L. Puterman. *Dynamic Programming and Optimal Control*. Wiley-Interscience, 1st edition edition, 2005.

Tarun Rambha and Stephen D. Boyles. Game theory and traffic assignment. Technical report, SWUTC/13/600451-00065-1, 2013.

Tarun Rambha and Stephen D. Boyles. Dynamic pricing in discrete time stochastic day-to-day route choice models. *Transportation Research Part B: Methodological*, 2016. To Appear.

Tarun Rambha, Stephen D. Boyles, Avinash Unnikrishnan, and Peter Stone. Marginal cost pricing for system optimal traffic assignment with recourse under supply-side uncertainty. In Review in *Transportation Research Part B: Methodological*, 2016.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Julia Robinson. An iterative method of solving a game. *Annals of Mathematics*, 54(2):pp. 296–301, 1951.

Timothy Avelin Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, 2002.

William H. Sandholm. *Population games and evolutionary dynamics*. MIT Press, 2010.

Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the tenth international conference on machine learning*, volume 298, pages 298–305, 1993.

Yosef Sheffi. *Urban Transportation Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

Yosef Sheffi and Warren B. Powell. An algorithm for the equilibrium assignment problem with random link times. *Networks*, 12(2):191–207, 1982.

Satinder P Singh. Reinforcement learning algorithms for average-payoff markovian decision processes. In *AAAI*, volume 94, pages 700–705, 1994.

Maynard J. Smith and GR Price. The logic of animal conflict. *Nature*, 246: 15, 1973.

Tony E. Smith, Erik A. Eriksson, and Per O. Lindberg. Existence of optimal tolls under conditions of stochastic user-equilibria. In Brje Johansson and Lars-Gran Mattsson, editors, *Road Pricing: Theory, Empirical Assessment and Policy*, Transportation Research, Economics and Policy, pages 65–87. Springer Netherlands, 1995.

Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888. ACM, 2006.

Shoupeng Tang, Tarun Rambha, Reese Hatridge, Stephen Boyles, and Avinash Unnikrishnan. Modeling parking search on a network by using stochastic shortest paths with history dependence. *Transportation Research Record: Journal of the Transportation Research Board*, 2467:73–79, 2014.

Peter D. Taylor and Leo B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical biosciences*, 40(1):145–156, 1978.

Valentina Trozzi, Guido Gentile, Michael G.H. Bell, and Ioannis Kaparias. Dynamic user equilibrium in public transport networks with passenger congestion and hyperpaths. *Transportation Research Part B: Methodological*, 57:266 – 285, 2013.

Satish V. S. K. Ukkusuri. *Accounting for Uncertainty, Robustness, and Online Information in Transportation Networks*. PhD thesis, The University of Texas at Austin, 2005.

Avinash Unnikrishnan. *Equilibrium Models Accounting for Uncertainty and Information Provision in Transportation Networks*. PhD thesis, The University of Texas at Austin, 2008.

Avinash Unnikrishnan and S. Travis Waller. User equilibrium with recourse. *Networks and Spatial Economics*, 9(4):575–593, 2009.

Erik T. Verhoef, Richard H. M. Emmerink, Peter Nijkamp, and Piet Rietveld. Information provision, flat and fine congestion tolling and the efficiency of road usage. *Regional Science and Urban Economics*, 26:505–529, 1996.

Dirck Van Vliet. The frank-wolfe algorithm for equilibrium traffic assignment viewed as a variational inequality. *Transportation Research Part B: Methodological*, 21(1):87 – 89, 1987.

S. Travis Waller and Athanasios K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8 (3-4):279–292, 1992.

Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge England, 1989.

David Watling. Asymmetric problems and stochastic process models of traffic assignment. *Transportation Research Part B: Methodological*, 30(5):339 – 357, 1996.

David Watling and Martin L. Hazelton. The dynamics and equilibria of day-to-day assignment models. *Networks and Spatial Economics*, 3(3):349–370, 2003.

David P. Watling and Giulio E. Cantarella. Model representation and decision-making in an ever-changing world: The role of stochastic process models of transportation systems. *Networks and Spatial Economics*, pages 1–40, 2013.

Byung-Wook Wie and Roger L. Tobin. Dynamic congestion pricing models for general traffic networks. *Transportation Research Part B: Methodological*, 32(5):313 – 327, 1998.

Feng Xiao, Hongbo Ye, and Hai Yang. Optimal pricing of day-to-day flow dynamics. In *5th International Symposium on Dynamic Traffic Assignment*, 2014.

Fan Yang. Day-to-day dynamic optimal tolls with elastic demand. In *Transportation Research Board 87th Annual Meeting*, number 08-0305, 2008.

Fan Yang and Ding Zhang. Day-to-day stationary link flow pattern. *Transportation Research Part B: Methodological*, 43(1):119 – 126, 2009.

Hai Yang. System optimum, stochastic user equilibrium, and optimal link tolls. *Transportation Science*, 33(4):354–360, 1999.

Hai Yang and Xiaolei Wang. Managing network mobility with tradable credits. *Transportation Research Part B: Methodological*, 45(3):580 – 594, 2011.

Mehmet Bayram Yildirim and Donald W. Hearn. A first best toll pricing framework for variable demand traffic assignment problems. *Transportation Research Part B: Methodological*, 39(8):659 – 678, 2005.

Yafeng Yin and Yingyan Lou. Dynamic tolling strategies for managed lanes. *Journal of Transportation Engineering*, 135(2):45–52, 2009.

H. Peyton Young. The evolution of conventions. *Econometrica*, 61(1):pp. 57–84, 1993.

H Peyton Young. *Strategic learning and its limits*, volume 2002. Oxford University Press, 2004.

Ding Zhang, Anna Nagurney, and Jiahao Wu. On the equivalence between stationary link flow patterns and traffic network equilibria. *Transportation Research Part B: Methodological*, 35(8):731 – 748, 2001.