Marginal Cost Pricing for System Optimal Traffic Assignment with Recourse under Supply-Side Uncertainty

Tarun Rambha,* Stephen D. Boyles,† Avinash Unnikrishnan,‡ Peter Stone§

Abstract

Transportation networks are often subject to fluctuations in supply-side parameters such as capacity and free-flow travel time due to factors such as incidents, poor weather, and bottlenecks. In such scenarios, assuming that network arcs exist in a finite number of states with different delay functions with different probabilities, a marginal cost pricing scheme that leads to a socially optimal outcome is proposed. The suggested framework makes the behavioral assumption that travelers do not just choose paths but follow routing policies that respond to *en route* information. Specifically, it is assumed that travelers are fully-rational and that they compute the optimal online shortest path assuming full-reset. However, such policies may involve cycling, which is unrealistic in practice. Hence, a network transformation that helps restrict cycles up to a certain length is devised and the problem is reformulated as a convex optimization problem with symmetric delay functions. The results of numerical tests on the Sioux Falls test network are presented using the Frank-Wolfe algorithm.

Keywords: equilibrium with recourse; marginal cost pricing; supply-side uncertainty; online shortest paths

1 Introduction

Static traffic assignment models 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 paper 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

*Ph.D. Candidate, Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin (email: tarun.1988@gmail.com)

†Associate Professor, Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin (email: sboyles@utexas.edu)

‡Associate Professor, Department of Civil and Environmental Engineering, Portland State University (email: uavinash@pdx.edu)

§Professor, Department of Computer Science, The University of Texas at Austin (email: pstone@cs.utexas.edu)

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. For example, if we are modeling a three-hour peak period and if a minor accident usually reduces the capacity of a link for 15 minutes, it is reasonable to assume that 1/12 of the travelers will observe the link in an accident state and 11/12 of the travelers will not. The same holds true if there are multiple minor accidents in the peak period that reduce the capacity for a total of 15 minutes. 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 and they reset each time the traveler revisits the 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 paper 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.

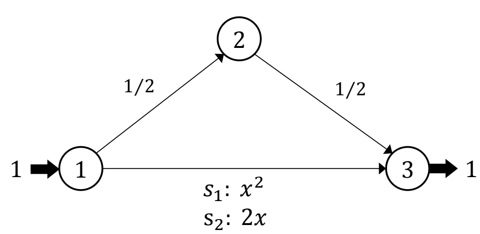

Figure 1: Demonstration of system optimal solutions with recourse.

To illustrate the basic SOR model, consider the example in Figure 1. Suppose that the 1 unit of demand between nodes 1 and 3 is infinitely divisible. Since we are modeling a nonatomic version of the problem, the terms 'travelers' and 'users' are to be interpreted as flow rates. Links (1,2) and (2,3) have a constant travel time of 1/2 units and is always incident free. The link (1,3) on the other hand is congestible and exists in two states with link performance/delay functions x^2 (under normal operating conditions) and $2x$ (when there is an incident on the link) with probabilities 0.6 and 0.4 respectively. These states on link (1,3) are referred to as s_1 and s_2. As mentioned earlier, the probability of a link-state represents the fraction of time for which the link is expected to be in that state. Note that the state of the link may change between s_1 and s_2 multiple times within the peak period but the total duration for which it is in states s_1 and s_2 is 60% and 40% of the peak period respectively. Of the 1 unit of demand arriving at node 1, 0.6 and 0.4 units of flow see arc (1,3) in states s_1 and s_2 respectively. (In general, if η travelers arrived at node 1, 0.6η and 0.4η travelers observe arc (1,3) 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 may require a traveler to head to node 2 if the state s_1 is observed at node 1 and move to node 3 otherwise. The action space at node 2 is a singleton and hence can be ignored while defining policies. Thus, each traveler has four policies to choose from (see Table 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 four policies. The cost of a policy is a random variable and hence we suppose that travelers choose policies which minimize the expected travel time.

Table 1: Flows on policies at UER and SOR states.

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

The system optimal solution may assign a positive demand to all four policies, whereas at equilibrium, all travelers select arc (1,3). Throughout this paper, we assume that all travelers have the same value of time (VOT) and the units for the tolls are chosen such that the VOT for each traveler equals 1. Extending the proposed models to scenarios involving multiple user classes with different VOTs (Dial, 1999b,c; Yang and Huang, 2004; Wu and Huang, 2014) falls beyond the scope of this paper and will be left for future work. 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 arc (1,3) 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 arcs (1,2) and (2,3) 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 paper 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.

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, 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. Also, the proposed SOR model is not a day-to-day pricing problem (Ye et al., 2015; Guo et al., 2015; Rambha and Boyles, 2016; Rambha, 2016), but is a "within-day pricing" problem in the sense that the system manager constantly monitors the network and every time the state of a link changes, a different toll is collected.

The rest of the paper is organized as follows. Section 2 introduces notation and describes the stochastic network model. In section 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 4, we detail the algorithms that can be used to compute the SOR solution and the optimal tolls. We then propose 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 5 contains some numerical experiments on the Sioux Falls test network and in Section 6 we summarize the findings in this paper and provide pointers for future research.

2 Preliminaries

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 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 4.2. A policy π *terminates at* i if the only eigenvector of $\mathbf{R}_\pi$ is the i-th standard basis $\mathbf{e}^T_i$, 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 $\rho^{s\pi}_{ij} = \sum_{\theta\in\Theta_i:\pi(i,\theta)=j,\theta_{ij}=s} q^\theta$ 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^s_{ij}. We will later consider the case where the travel times depend on flows and until then write it without reference to the link-state flows. The expected travel time C^π_i from each node i to the destination v can be calculated using the following equations

$$C^\pi_v = 0 \tag{1}$$

[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).

$$C_i^\pi = \sum_{j \in \Gamma(i)} \sum_{s \in S_{ij}} \rho_{ij}^{s\pi} (t_{ij}^s + C_j^\pi) \qquad \forall\, i \in N \backslash \{v\} \tag{2}$$

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

$$\mathcal{C}_{ij}^{s\pi} = t_{ij}^s + C_j^\pi \qquad \forall\, (i, j) \in A, s \in S_{ij} \tag{3}$$

or, upon eliminating the C variables, as the system

$$\mathcal{C}_{ij}^{s\pi} = t_{ij}^s + \sum_{k \in \Gamma(j)} \sum_{\bar{s} \in S_{jk}} \rho_{jk}^{\bar{s}\pi} \mathcal{C}_{jk}^{\bar{s}\pi} \qquad \forall\, (i, j) \in A, s \in S_{ij} \tag{4}$$

This equation can be expressed in matrix form as

$$\boldsymbol{\mathcal{C}}^\pi = \mathbf{t} + \mathbf{P}_\pi \boldsymbol{\mathcal{C}}^\pi \tag{5}$$

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 network in Figure 2 taken from Waller and Ziliaskopoulos (2002). 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.

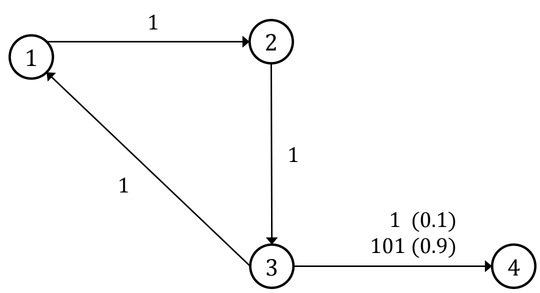

Figure 2: Computing the cost of a policy.

Suppose a user is traveling from node 1 to 4. Consider a policy π 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. (This policy is in fact optimal for the traveler.) For the assumed policy, the $\mathbf{P}_\pi$ matrix may be populated as shown in equation (6).

$$\mathbf{P}_\pi = \begin{array}{cc} & \begin{array}{ccccc} (1,2),[1] & (2,3),[1] & (3,1),[1] & (3,4),[1] & (3,4),[101] \end{array} \\ \begin{array}{c} (1,2),[1] \\ (2,3),[1] \\ (3,1),[1] \\ (3,4),[1] \\ (3,4),[101] \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{6}$$

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:

$$x_{ij}^{s\pi} = \rho_{ij}^{s\pi}\eta_i^\pi \qquad \forall (i,j) \in A, \pi \in \Pi \tag{7}$$

$$\eta_i^\pi = y_i^\pi + \sum_{h\in\Gamma^{-1}(i)} \sum_{\bar{s}\in S_{hi}} x_{hi}^{\bar{s}\pi} \qquad \forall i \in N, \pi \in \Pi \tag{8}$$

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

$$x_{ij}^{s\pi} = \rho_{ij}^{s\pi} y_i^\pi + \rho_{ij}^{s\pi} \sum_{h\in\Gamma^{-1}(i)} \sum_{\bar{s}\in S_{hi}} x_{hi}^{\bar{s}\pi} \qquad \forall\, (i,j) \in A, s \in S_{ij}, \pi \in \Pi \tag{9}$$

This equation can be expressed in matrix form as

$$\mathbf{x}^\pi = \mathbf{b}^\pi + \mathbf{P}_\pi^T \mathbf{x}^\pi \tag{10}$$

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 (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, 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 Figure 3a. This process can be repeated for the 0.9 units of demand that cycles back to node 3 (see Figure 3b) and for the 0.81 units of demand that cycles twice (see Figure 3c).

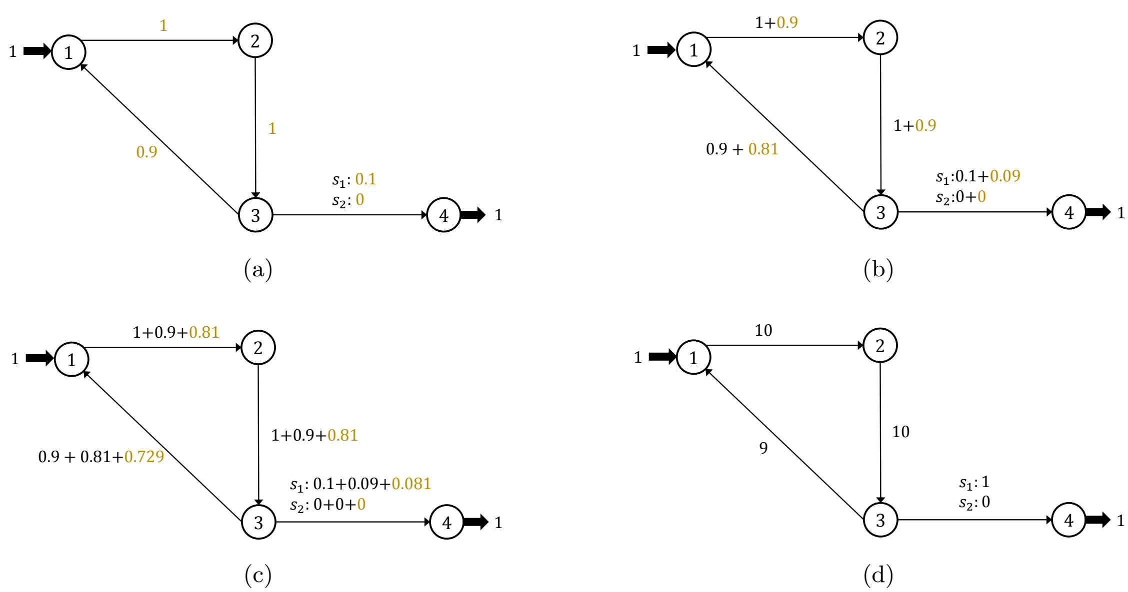

Figure 3: Network loading travelers iteratively.

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 3d. Alternately, we can solve the flow conservation equations $\mathbf{x}^\pi = (\mathbf{I} - \mathbf{P}_\pi^T)^{-1}\mathbf{b}^\pi$, described by equation (11), to obtain link-state flows.

$$\left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right]^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 9 \\ 1 \\ 0 \end{pmatrix} \tag{11}$$

3 SOR and Marginal Cost Pricing

Let $\mathbf{y}$ denote the vector $(\mathbf{y}^\pi)_{\pi \in \Pi_v}$ and let $\mathbf{x} = (x_{ij}^s)_{(i,j)\in S, s\in S_{ij}}$ 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

$$\begin{aligned}
\text{TETT} = \sum_{(u,v)\in Z^2} \sum_{\pi \in \Pi_v} y_u^\pi C_u^\pi(\mathbf{y}) &= \sum_{v\in Z} \sum_{\pi\in\Pi_v} \sum_{i \in N} y_i^\pi C_i^\pi(\mathbf{y}) && (12)\\
&= \sum_{\pi\in\Pi} \sum_{i\in N} y_i^\pi \sum_{j\in\Gamma(i)} \sum_{s\in S_{ij}} \rho_{ij}^{s\pi} \mathcal{C}_{ij}^{s\pi}(\mathbf{y}) \qquad \text{[using (2) and (3)]} && (13)\\
&= \sum_{\pi\in\Pi} \sum_{(i,j)\in A} \sum_{s\in S_{ij}} y_i^\pi \rho_{ij}^{s\pi} \mathcal{C}_{ij}^{s\pi}(\mathbf{y}) && (14)\\
&= \sum_{\pi\in\Pi} (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T \mathbf{b}^\pi && (15)\\
&= \sum_{\pi\in\Pi} (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T \mathbf{A}_\pi \mathbf{A}_\pi^{-1} \mathbf{b}^\pi && (16)\\
&= \sum_{\pi\in\Pi} (\mathbf{A}_\pi^T \boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T \mathbf{x}^\pi \qquad \text{[using (10)]} && (17)\\
&= \sum_{\pi\in\Pi} \sum_{(i,j)\in A} \sum_{s\in S_{ij}} t_{ij}^s(x_{ij}^s) x_{ij}^{s\pi} \qquad \text{[using (5)]} && (18)\\
&= \sum_{(i,j)\in A} \sum_{s\in S_{ij}} t_{ij}^s(x_{ij}^s) x_{ij}^s && (19)
\end{aligned}$$

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

$$\begin{aligned}
\min_{\mathbf{y},\mathbf{x},\mathbf{x}^\pi,\mathbf{b}^\pi} \quad & \sum_{(i,j)\in A} \sum_{s\in S_{ij}} x_{ij}^s t_{ij}^s(x_{ij}^s) && \text{(SOR)} && (20)\\
\text{s.t.} \quad & \sum_{\pi\in\Pi_v} y_u^\pi = d_{uv} && \forall (u,v)\in Z^2 && (21)\\
& \sum_{\pi\in\Pi} x_{ij}^{s\pi} = x_{ij}^s && \forall (i,j)\in A, s\in S_{ij} && (22)\\
& \mathbf{A}_\pi \mathbf{x}^\pi = \mathbf{b}^\pi && \forall \pi\in\Pi && (23)\\
& y_u^\pi \geq 0 && \forall \pi\in\Pi, u\in Z && (24)
\end{aligned}$$

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

$$y_u^\pi > 0 \Rightarrow C_u^\pi(\mathbf{y}) = \min_{\pi' \in \Pi_v} C_u^{\pi'}(\mathbf{y}) \qquad \forall v \in Z, \pi \in \Pi_v \,. \tag{25}$$

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

$$\min_{\mathbf{y},\mathbf{x},\mathbf{x}^\pi,\mathbf{b}^\pi} \sum_{(i,j)\in A} \sum_{s \in S_{ij}} \int_0^{x_{ij}^s} t_{ij}^s(x)\, dx \qquad \text{(UER)} \tag{26}$$

$$\text{s.t.} \quad \sum_{\pi \in \Pi_v} y_u^\pi = d_{uv} \qquad \forall (u,v) \in Z^2 \tag{27}$$

$$\sum_{\pi \in \Pi} x_{ij}^{s\pi} = x_{ij}^s \qquad \forall (i,j) \in A, s \in S_{ij} \tag{28}$$

$$\mathbf{A}_\pi \mathbf{x}^\pi = \mathbf{b}^\pi \qquad \forall \pi \in \Pi \tag{29}$$

$$y_u^\pi \geq 0 \qquad \forall \pi \in \Pi, u \in Z \tag{30}$$

Proposition 1. *The optimal solutions to the convex program* (26)–(30) *correspond exactly to policy flows satisfying the UER definition* (25).

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 (27) (with multipliers $\boldsymbol{\kappa}$), and substitute constraints (28) and (29) 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

$$\mathcal{L}(\mathbf{y}, \boldsymbol{\kappa}) = \sum_{(i,j)\in A} \sum_{s \in S_{ij}} \int_0^{\sum_{\pi\in\Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_\pi^{-1} \mathbf{b}^\pi} t_{ij}^s(x)\, dx + \sum_{(u,v)\in Z^2} \kappa_{uv}(d_{uv} - \sum_{\pi \in \Pi_v} y_u^\pi) \tag{31}$$

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

$$\sum_{(i,j)\in A} \sum_{s \in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi} \left(\sum_{\bar\pi \in \Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_{\bar\pi}^{-1} \mathbf{b}^{\bar\pi} \right) - \kappa_{uv} \geq 0 \qquad \forall \pi \in \Pi_v \tag{32}$$

$$y_u^\pi \left(\sum_{(i,j)\in A} \sum_{s \in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi} \left(\sum_{\bar\pi \in \Pi} (\mathbf{e}_{ij}^s)^T \mathbf{A}_{\bar\pi}^{-1} \mathbf{b}^{\bar\pi} \right) - \kappa_{uv} \right) = 0 \qquad \forall \pi \in \Pi_v, u \in Z \tag{33}$$

$$\sum_{\pi \in \Pi_v} y^\pi = d_{uv} \qquad \forall (u,v) \in Z^2 \tag{34}$$

$$y_u^\pi \geq 0 \qquad \forall \pi \in \Pi, u \in Z \tag{35}$$

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 (32) and (33) 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

$$\begin{aligned}
\sum_{(i,j)\in A}\sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) &= \mathbf{t}^T\mathbf{d} && (36)\\
&= \left((\mathbf{I}-\mathbf{P}_\pi)\boldsymbol{\mathcal{C}}^\pi(\mathbf{y})\right)^T\mathbf{d} \qquad \text{[using (5)]} && (37)\\
&= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T(\mathbf{I}-\mathbf{P}_\pi)^T\mathbf{d} && (38)\\
&= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T\mathbf{A}_\pi\mathbf{d} && (39)
\end{aligned}$$

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

$$\begin{aligned}
\frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) &= \frac{\partial}{\partial y_u^\pi}\left((\mathbf{e}_{ij}^s)^T\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) && (40)\\
&= (\mathbf{e}_{ij}^s)^T\frac{\partial}{\partial y_u^\pi}\left(\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) && (41)
\end{aligned}$$

which implies that $\mathbf{d} = \mathbf{I}\frac{\partial}{\partial y_u^\pi}\left(\mathbf{A}_\pi^{-1}\mathbf{b}^\pi\right) = \mathbf{A}_\pi^{-1}\frac{\partial \mathbf{b}^\pi}{\partial y_u^\pi}$. Therefore, equation (39) can be rewritten as

$$\begin{aligned}
\sum_{(i,j)\in A}\sum_{s\in S_{ij}} t_{ij}^s \frac{\partial}{\partial y_u^\pi}\left(\sum_{\bar\pi\in\Pi}(\mathbf{e}_{ij}^s)^T\mathbf{A}_{\bar\pi}^{-1}\mathbf{b}^{\bar\pi}\right) &= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T\mathbf{A}_\pi\mathbf{A}_\pi^{-1}\frac{\partial \mathbf{b}^\pi}{\partial y_u^\pi} && (42)\\
&= (\boldsymbol{\mathcal{C}}^\pi(\mathbf{y}))^T\frac{\partial \mathbf{b}^\pi}{\partial y_u^\pi} && (43)\\
&= \sum_{(u,j)\in A}\sum_{s\in S_{uj}} \rho_{uj}^{s\pi}\mathcal{C}_{uj}^{s\pi}(\mathbf{y}) && (44)\\
&= C_u^\pi(\mathbf{y}) \qquad \text{[using (2) and (3)]} && (45)
\end{aligned}$$

Therefore, a solution to the convex program (26)–(30) 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 $\hat t_{ij}^s(x_{ij}^s) = t_{ij}^s(x_{ij}^s) + x_{ij}^s(t_{ij}^s)'(x_{ij}^s)$.

[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.

Proposition 2. *An feasible solution to the convex program (UER) with respect to tolled link performance functions $\hat{t}^s_{ij}$ is an optimal solution to (UER) if and only if it is optimal to (SOR) with respect to the original link performance functions t^s_{ij}.*

Proof. Consider a generic term $\int_0^{x^s_{ij}} \hat{t}^s_{ij}(x)\ dx$ in the objective function (26). Using the definition of $\hat{t}^s_{ij}$, this can be rewritten as

$$\int_0^{x^s_{ij}} t^s_{ij}(x)\ dx + \int_0^{x^s_{ij}} x(t^s_{ij})'(x)\ dx\,. \tag{46}$$

Integrating the second term by parts, we have

$$\int_0^{x^s_{ij}} xt'_{ij}(x)\ dx = x^s_{ij}t^s_{ij}(x^s_{ij}) - \int_0^{x^s_{ij}} t^s_{ij}(x)\ dx \tag{47}$$

which, upon substitution into (46) shows that $\int_0^{x^s_{ij}} \hat{t}^s_{ij}(x)\ dx = x^s_{ij}t^s_{ij}(x^s_{ij})$. 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. ∎

4 Solution Methods

The UER and SOR models defined in Section 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.

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 (1) and (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 $\leftarrow \infty$
iteration $\leftarrow 0$

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 $\leftarrow (\mathbf{t}\cdot\mathbf{x})\left(\sum_{u\in Z}\sum_{v\in Z} d_{uv} C_u^{\pi_v^*}\right)^{-1} - 1$
  $\varphi^* \leftarrow \arg\min_{\varphi\in[0,1]} \sum_{(i,j)\in A}\sum_{s\in S_{ij}} \int_0^{(1-\varphi)x_{ij}^s + \varphi x_{ij}^{s*}} \hat{t}_{ij}^s(x)dx$ ▷ Optimal Step Size
  if *iteration* $= 0$ **then** $\varphi^* \leftarrow 1$, GAP $\leftarrow \infty$
  $\mathbf{x} \leftarrow (1-\varphi^*)\mathbf{x} + \varphi^*\mathbf{x}^*$
  $\mathbf{t} \leftarrow \hat{\mathbf{t}}(\mathbf{x})$
  iteration $\leftarrow$ *iteration* $+ 1$
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 REDUCE($\boldsymbol{\xi}', \boldsymbol{\lambda}'$) returns vectors $\boldsymbol{\lambda} = [5 \;\; 7 \;\; 8 \;\; 1 \;\; 6]$ and $\boldsymbol{\xi} = [0.25 \;\; 0.125 \;\; 0.25 \;\; 0.25 \;\; 0.125]$.

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

Step 1: Initialize Labels
$C_v \leftarrow 0$
$C_i \leftarrow \infty \, \forall \, i \in N \backslash \{v\}$
SEL $\leftarrow \Gamma^{-1}(v)$

Step 2:
while SEL $\neq \varnothing$ **do**
  Remove i from SEL
  $\boldsymbol{\xi} \leftarrow [1]$, $\boldsymbol{\lambda} \leftarrow [\infty]$
  for $j \in \Gamma(i)$ **do** ▷ Step 2.1
    $\boldsymbol{\xi}' \leftarrow [\ ]$, $\boldsymbol{\lambda}' \leftarrow [\ ]$
    for $s \in S_{ij}$ **do**
      for $k = 1, \ldots, |\boldsymbol{\xi}|$ **do**
        $\boldsymbol{\xi}' \leftarrow [\boldsymbol{\xi}' \ \ \xi_k p_{ij}^s]$
        if $t_{ij}^s + C_i < \lambda_k$ **then**
          $\boldsymbol{\lambda}' \leftarrow [\boldsymbol{\lambda}' \ \ t_{ij}^s + C_i]$
        else
          $\boldsymbol{\lambda}' \leftarrow [\boldsymbol{\lambda}' \ \ \lambda_k]$
        end if
      end for
    end for
    $(\boldsymbol{\xi}, \boldsymbol{\lambda}) \leftarrow \textsc{Reduce}(\boldsymbol{\xi}', \boldsymbol{\lambda}')$
  end for
  if $\boldsymbol{\xi} \cdot \boldsymbol{\lambda} < C_i$ **then** ▷ Step 2.2
    $C_i \leftarrow \boldsymbol{\xi} \cdot \boldsymbol{\lambda}$
    $SEL \leftarrow SEL \cup \Gamma^{-1}(i)$
  end if
end while

Step 3: Choose Optimal Policy
for $i \in N, \theta \in \Theta_i$ **do**
  if $i = v$ **then**
    $\pi_v^*(i, \theta) = v$
  else
    $\pi_v^*(i, \theta) \in \arg\min_{j \in \Gamma(i)} \{t_{ij}^{\theta_{ij}} + C_j\}$
  end if
end for
return π_v^*

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]

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

For example, consider the network in Figure 4. 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 free flow travel time 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 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 directly or 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.

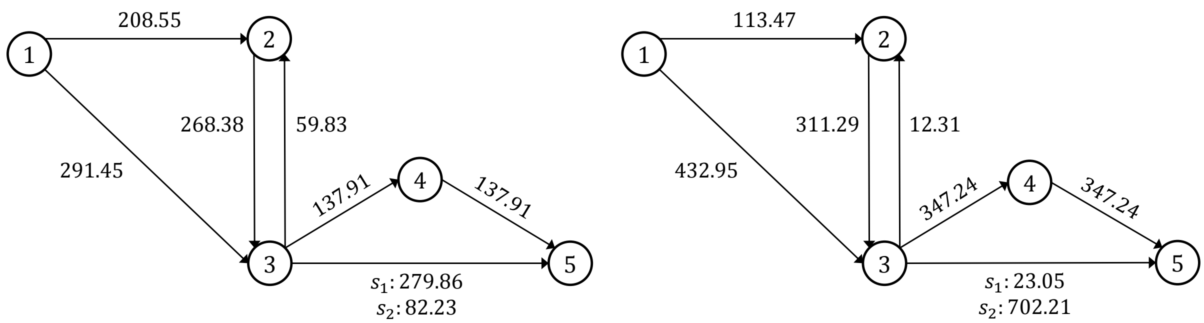

Figure 4: Optimal flows (left) and tolls (right) in a network that illustrates cycling.

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 problem) is difficult due to Proposition 4.2.

Proposition 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

[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.

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 criterion 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 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 (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 intuition behind this comes from the fact that the SOR and UER problems on acyclic graphs do not involve cyclic policies. The suggested transformation exploits this fact by eliminating cycles of certain lengths which makes the underlying graph relatively "acyclic". We discuss this technique using 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 5).

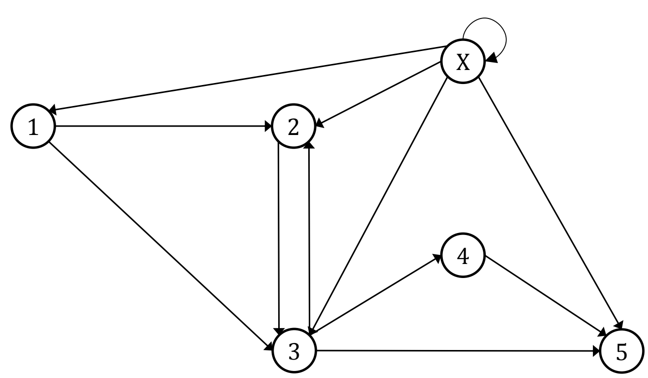

Figure 5: Network with dummy node to enumerate recently visited nodes.

[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, it is not trivial to solve the UER and SOR problems using faster equilibrium algorithms such as bush and origin-based methods (Dial, 2006; Bar-Gera, 2002, 2010).

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 5, when $m = 2$, $M_2(2) = \{1, 2, 3, X\}$. The dummy node X is useful 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 ∈ N do
    Use BFS to find nodes that can reach i
    for j = 0, 1, ..., m do
        Populate M_i(j) using the BFS distance labels
    end for
    M_i ← ×_{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 $(i, j) \in A$ (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 5, when $m = 1$, the network $\mathcal{G}$ (see Figure 6) 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 $(i, j) \in A$.

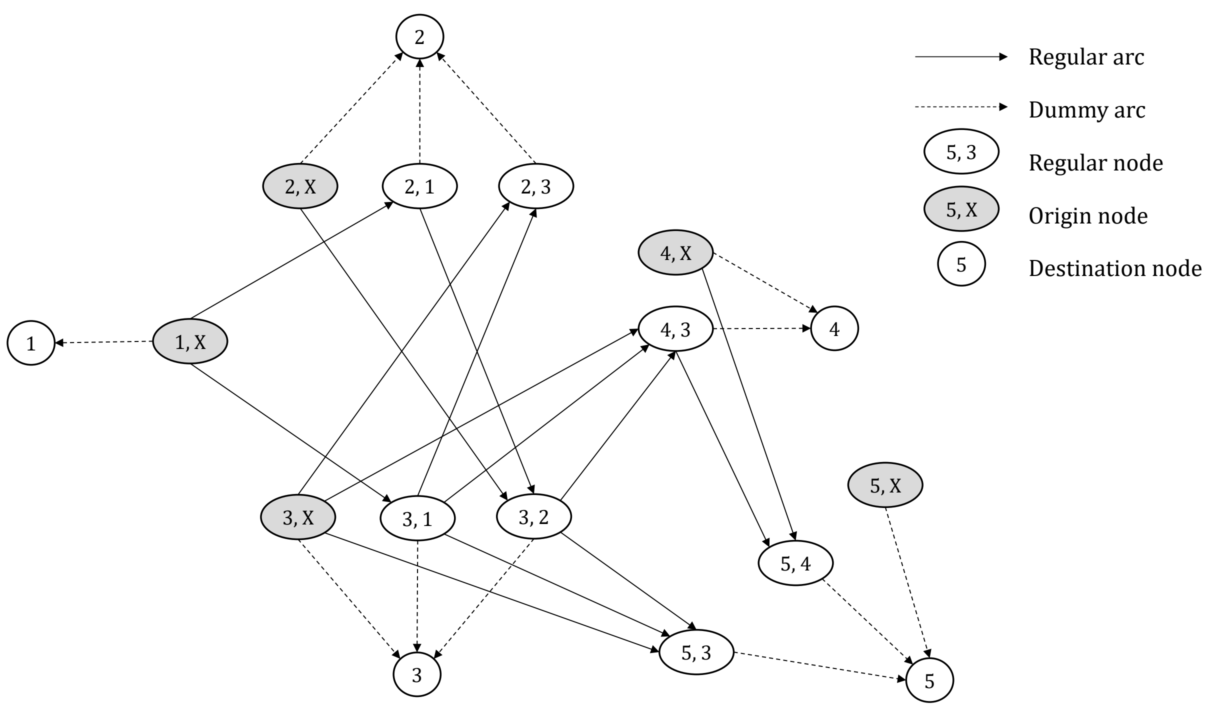

Figure 6: Network transformation to restrict cycling.

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 5, if d_{15} users travel from node 1 to node 5, then the demand between $(1, X)$ and 5 in Figure 6 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 5 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 users 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., $\partial t^s_{k_1 l_1}/\partial x^s_{k_2 l_2} = \partial t^s_{k_2 l_2}/\ \partial x^s_{k_1 l_1}\ \forall (k_1, l_1), (k_2, l_2) \in \mathcal{A}$. This holds because $t^s_{kl}(\mathbf{x}_{\mathcal{A}}) = t^s_{ij}(\sum_{(\hat{k},\hat{l})\in\mathcal{A}_{ij}} x^s_{\hat{k}\hat{l}})$ for all regular arcs and because the travel times on the dummy arcs are zero. Hence, the FW method can be applied to find the equilibrium solution and the optimal tolls (Vliet, 1987).

For the network in Figure 4, 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 7. 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.

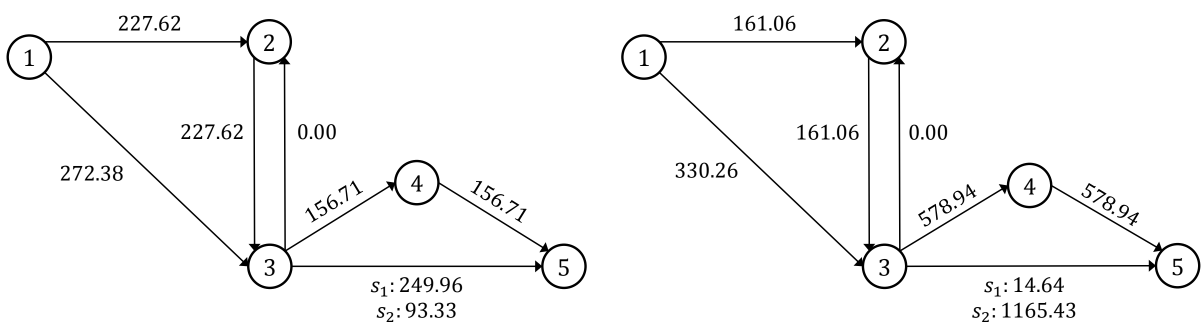

Figure 7: Optimal flows (left) and tolls (right) for the 1-SOR problem.

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 8). 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: 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

Table 2 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.

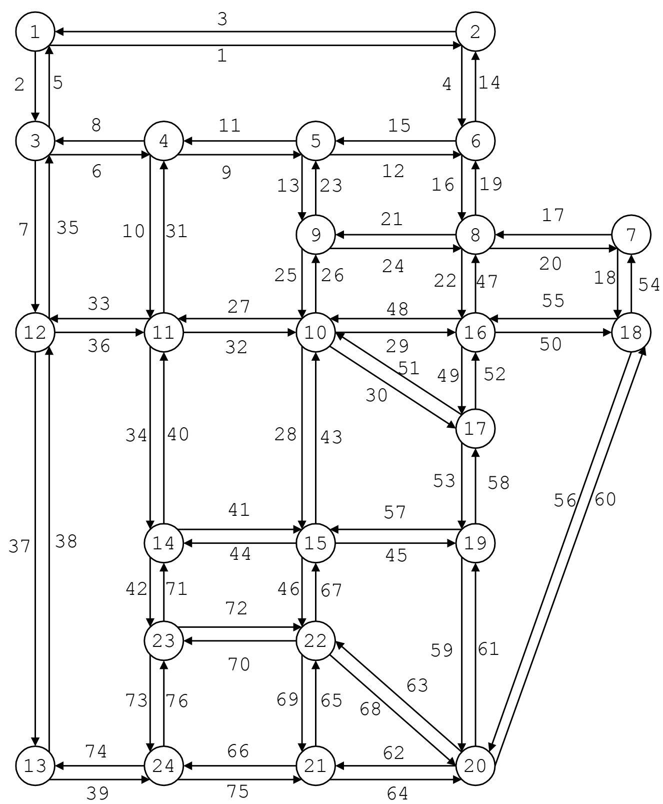

Figure 8: Sioux Falls network.

A comparison of the marginal tolls for different values of m is presented in Table 3. 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 3: 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 9 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.

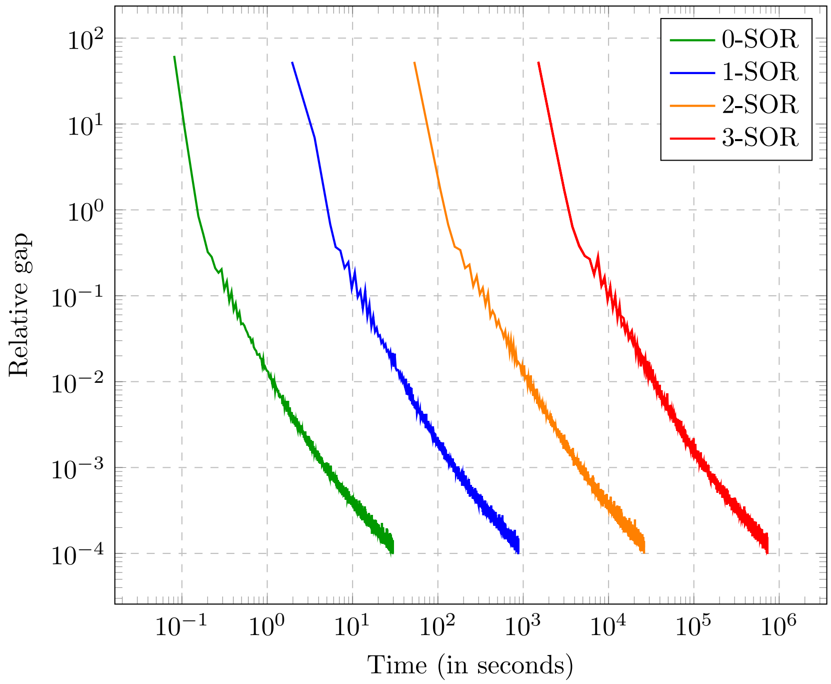

Figure 9: Computational performance of the FW method for different SOR variants.

Impact of Static Tolls and Sensitivity to Input Parameters

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 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.

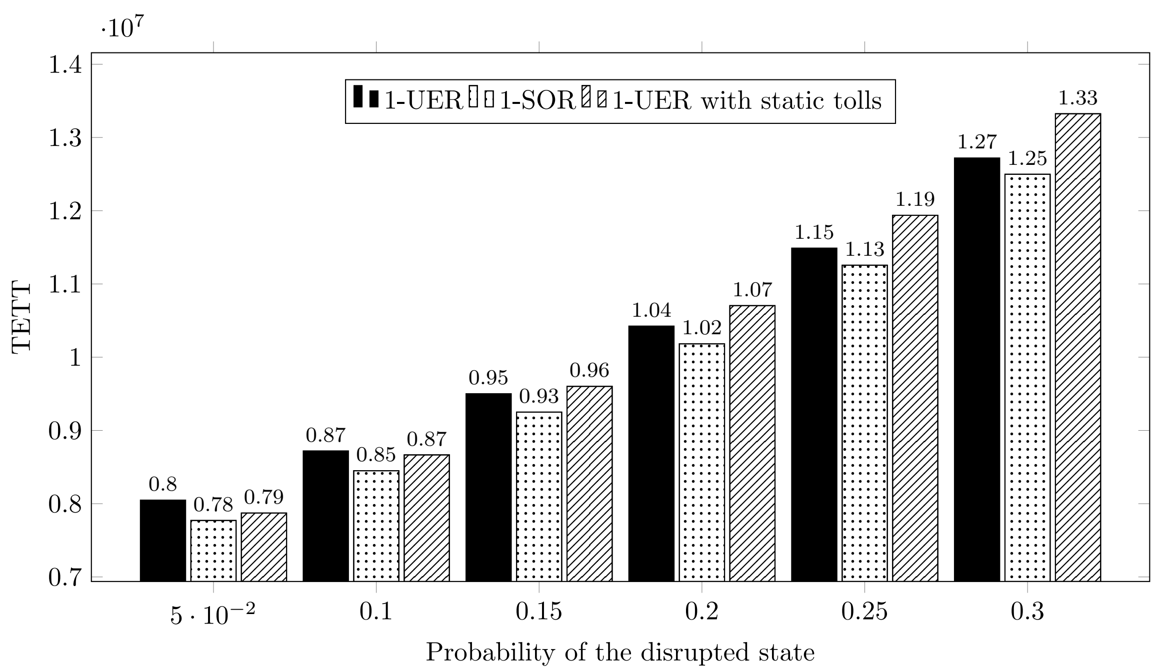

Figure 10: Impact of static and state-dependent marginal tolls for different disruption probabilities.

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 10 and 11. 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 10 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).

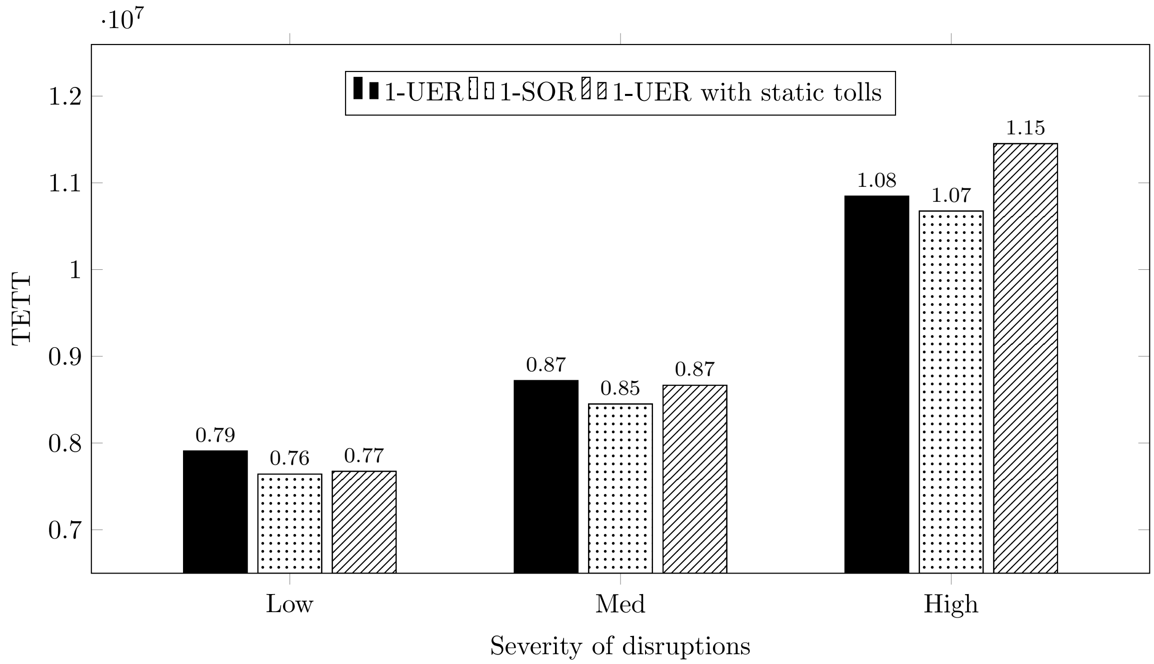

Figure 11: Impact of static and state-dependent marginal tolls for different levels of disruption severity.

Sensitivity analysis with respect to severity of disruptions revealed a similar trend in the TETT values (see Figure 10). 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.

6 Summary

In this paper, 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 the cycle lengths while computing the optimal policies. 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.

The framework developed in this paper motivates several topics for future research. (1) The sub-optimality of static tolls calls for accurately estimating supply-side variables and their distributions using historic incident and weather data. (2) Improving the run times using advanced algorithms such as 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) In this study, travelers' actions were assumed to be conditioned on the downstream link-states. However, with V2X technologies, travelers will have access to reliable real-time state and toll information at a network level. Also, the current paper 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 paper. (4) Formulating the minimum revenue congestion pricing problem (Bergendorff et al., 1997; Dial, 1999a, 2000; Penchina, 2004) for system optimum with recourse may make the state-dependent pricing model more appealing by potentially reducing or eliminating tolls, especially on disrupted links. (5) While it was assumed that all users have the ability to process *en route* information, relaxing this assumption by allowing only a fraction of them to replan can help model more practical scenarios that include both human and autonomous drivers. Also, as noted by Gao (2012) and Du et al. (2015), disseminating information strategically can improve network performance and it would be interesting to compare its impacts with marginal tolls. (6) Lastly, incorporating demand-side uncertainty (Gardner et al., 2008; Sumalee and Xu, 2011; Wang et al., 2014; Bansal et al., 2017) into the SOR model would provide a much needed holistic approach to mitigate congestion using tolls.

Acknowledgments

The authors would like to acknowledge the support of National Science Foundation (Grant Nos. 1069141/1157294, 1254921, and 1562109) and the Data-Supported Transportation Planning and Operations University Transportation Center for conducting this research. The authors also thank Michael Levin, Michael Albert, Guni Sharon, and Josiah Hanna for their useful comments. Comments from two anonymous reviewers are also greatly appreciated.

References

Bansal, P., R. Shah, and S. D. Boyles (2017). Robust network pricing and system optimization under combined long-term stochasticity and elasticity of travel demand. *Transportation*, 1–30.

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

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

Bergendorff, P., D. W. Hearn, and M. V. Ramana (1997). *Network Optimization*, Chapter Congestion Toll Pricing of Traffic Networks, pp. 51–71. Berlin, Heidelberg: Springer Berlin Heidelberg.

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

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

Boyles, S. D., S. Tang, and A. Unnikrishnan (2015). Parking search equilibrium on a network. *Transportation Research Part B: Methodological 81, Part 2*, 390 – 409.

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

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

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

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

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

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

Du, L., L. Han, and S. Chen (2015). Coordinated online in-vehicle routing balancing user optimality and system optimality through information perturbation. *Transportation Research Part B: Methodological 79*, 121 – 133.

Frank, M. and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly 3*, 95–110.

Gao, S. (2012). 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.

Gardner, L., A. Unnikrishnan, and S. Waller (2008). Robust pricing of transportation networks under uncertain demand. *Transportation Research Record: Journal of the Transportation Research Board 2085*, 21–30.

Guo, R.-Y., H. Yang, H.-J. Huang, and Z. Tan (2015). Link-based day-to-day network traffic dynamics and equilibria. *Transportation Research Part B: Methodological 71*(0), 248 – 260.

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

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

Hamdouch, Y., P. Marcotte, and S. Nguyen (2004). A strategic model for dynamic traffic assignment. *Networks and Spatial Economics 4*, 291–315.

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

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

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

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

Pigou, A. C. (1920). *The Economics of Welfare*. London: Macmillan and Co.

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

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

Rambha, T. (2016). *Dynamic congestion pricing in within-day and day-to-day network equilibrium models.* Ph. D. thesis, The University of Texas at Austin.

Rambha, T. and S. D. Boyles (2016). Dynamic pricing in discrete time stochastic day-to-day route choice models. *Transportation Research Part B: Methodological 92, Part A*, 104 – 118. Special issue Day-to-Day Dynamics in Transportation Networks.

Sheffi, Y. (1985). *Urban Transportation Networks*. Englewood Cliffs, NJ: Prentice-Hall.

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

Sumalee, A. and W. Xu (2011). First-best marginal cost toll for a traffic network with stochastic demand. *Transportation Research Part B: Methodological 45*(1), 41 – 59.

Tang, S., T. Rambha, R. Hatridge, S. Boyles, and A. Unnikrishnan (2014). 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.

Trozzi, V., G. Gentile, M. G. Bell, and I. Kaparias (2013). Dynamic user equilibrium in public transport networks with passenger congestion and hyperpaths. *Transportation Research Part B: Methodological 57*, 266 – 285.

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

Unnikrishnan, A. (2008). *Equilibrium Models Accounting for Uncertainty and Information Provision in Transportation Networks.* Ph. D. thesis, The University of Texas at Austin.

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

Vliet, D. V. (1987). The Frank-Wolfe algorithm for equilibrium traffic assignment viewed as a variational inequality. *Transportation Research Part B: Methodological 21*(1), 87 – 89.

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

Wang, C., X. V. Doan, and B. Chen (2014). Price of anarchy for non-atomic congestion games with stochastic demands. *Transportation Research Part B: Methodological 70*, 90 – 111.

Wu, W.-X. and H.-J. Huang (2014). Finding anonymous tolls to realize target flow pattern in networks with continuously distributed value of time. *Transportation Research Part B: Methodological 65*, 31 – 46.

Yang, H. and H.-J. Huang (2004). The multi-class, multi-criteria traffic network equilibrium and systems optimum problem. *Transportation Research Part B: Methodological 38*(1), 1 – 15.

Ye, H., H. Yang, and Z. Tan (2015). Learning marginal-cost pricing via a trial-and-error procedure with day-to-day flow dynamics. *Transportation Research Part B: Methodological 81, Part 3*, 794 – 807.