Marginal cost pricing for system optimal traffic assignment with recourse under supply-side uncertainty

0 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.

1 Introduction

- Links in the network can be modeled using different states indicating accident conditions.
- To achieve the TETT, tolls change as a function of network states. The states of each link can be obtained through historic data, that is, the probability of each state corresponds to the time of that specific scenario.
- 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.
- Routing policy represents the adaptive routing behavior of the travelers informed of en-route information.
- To avoid NP-hard, assume that 1)the states observed at a node are independent of the states observed by any other traveler arriving at that node and 2)they **reset** each time the traveler revisits the node.
- However, this may cause cycling, and solution for this is to restrict the routing policies to only allow certain length of cycle.
- SOR 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 instead of a "day-to-day" problem.

2 Preliminaries

PARAMETERS	ANNOTATIONS
$G = (N, A)$	Strongly connected transportation network
$Z \subseteq N$	The subset of nodes where trips begin and end
$\Gamma(i)$ and $\Gamma^{-1}(i)$	The downstream and upstream nodes of node i
d_{uv}[E3]	The demand from origin u to destination v
S_{ij}	The set of states of each arc
$t_{ij}^s(x_{ij}^s)$	The link performance function for state $s \in S_{ij}$
x_{ij}^s	The number of travelers using link (i-j) in state s
$S = \cup_{(i,j)\in A} S_{ij}$	The set of all link-states
$\|N\|$ and $\|S\|$	The number of nodes and link-states
$\Theta_i = \times_{(i,j)\in A} S_{ij}$	The set of possible message that can be received at node i
$\theta \in \Theta_i$	A message vector informs the state of each link leaving node i
θ_{ij} or simply s	The state of link (i, j) corresponding to message θ

PARAMETERS	ANNOTATIONS
q^{θ}[E1]	The probability of receiving message vector $\theta \in \Theta_i$
$\Phi = \{(i, \theta) : i \in N, \theta \in \Theta_i\}$	The set of node-states which correspond to the decision point
$\pi : \Phi \to N$	A policy is a function that maps each node-state to the node associated with this node-state or a downstream node
$\mathbf{R}_{\pi} \in \mathbb{R}_{+}^{\lvert N \rvert \times \lvert N \rvert}$[E2]	A Markov chain with a transition matrix represents the probabilities of moving from each node to any other with respect to policy π.
$\prod_v$	The set of non-waiting policies terminating at v
$\prod = \cup_{v \in Z} \prod_v$	All the non-waiting policies
$\rho_{ij}^{s\pi}$[P1]	The probability of leaving node i via link (i, j) in state $s \in S_{ij}$
C_i^{π}	The expected travel time from node i to the destination
$C_{ij}^{s\pi}$	The expected travel time to the destination starting at the upstream of link (i, j) in state s
y_i^{π}	The number of travelers originating at node i and choose policy π
η_i^{π}	The number of travelers arriving at node i using policy π
$x_{ij}^{s\pi}$	The number of travelers on policy π who experience link (i, j) in state s

Equation 1:

$$q_i^{\theta} = \prod_{(i,j) \in A} p_{ij}^{\theta_{ij}}$$

Equation 2:

$$\mathbf{R}_{\pi}(i, j) = \sum_{\theta \in \Theta_i : \pi(i, \theta) = j} q^{\theta}$$

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 by traversing exactly m unique arcs. An optimal policy can be cyclic because of the full-reset assumption. A policy π terminates at i if the only eigenvector of Rπ is the ith standard basis e_i^T, and is **non-waiting** if $\pi(i, \theta) = i$ only exists when π terminates at i.

Process 1-Calculate the expected costs of each link-state

Consider a policy $\pi \in \Pi_v$. The probability of leaving node i via link (i, j) in state $s \in S_{ij}$: $\rho_{ij}^{s\pi} = \sum_{\theta \in \Theta_i : \pi(i,\theta)=j, \theta_{ij}=s} q^\theta$

The **expected** travel time C_i^π from each node i to the destination v:

$$C_v^\pi = 0 \tag{1}$$

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

The expected travel time to the destination v from a traveler starting at the upstream end of on link (i, j) in state s: $C_{ij}^{s\pi}$.

$$C_{ij}^{s\pi} = t_{ij}^s + C_j^\pi \quad \forall\, (i,j) \in A, s \in S_{ij} \tag{3}$$

Then eliminate C variables:

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

This equation can be expressed in matrix form as:

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

Process 2-Calculate the flow on each link-state

The number of travelers originating at node i and choose policy π: y_i^π. Flow conservation requires $y_u^\pi \geq 0$ for all origins u and policies π. A vector $\mathbf{y}^\pi$ is feasible if it satisfies flow conservation.

Equation 3:

$$d_{uv} = \sum_{\pi \in \Pi_v} y_u^\pi$$

The number of travelers arriving at node i using policy π: η_i^π. The number of travelers on policy π who experience link (i, j) in state s: $x_{ij}^{s\pi}$.

Any $\mathbf{y}^\pi$ defines a vector η^π as well as the vector $\mathbf{x}^\pi$, through linear system:

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

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

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

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

This equation can be expressed in matrix form as:

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

where $\mathbf{b}^{\pi} = vec(\rho_{ij}^{s\pi} y_i^{\pi})$. Thus, we may write $\mathbf{x}^{\pi} = \mathbf{A}_{\pi}^{-1} \mathbf{b}^{\pi}$ where $\mathbf{A}_{\pi} = \mathbf{(I - P_{\pi}^{T})}$.

Note that the columns of A_{π}^{-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[2].

3 SOR and marginal cost pricing

Let $\mathbf{y}$ denote the vector $(\mathbf{y}^{\pi})_{\pi \in \Pi_v}$, let $\mathbf{x} = (x_{ij}^{s})_{(i,j) \in S, s \in S_{ij}}$ denote link flows for each state aggregated by policies and let $\mathbf{C}^{\pi}\mathbf{(y)}$ denote the policy costs. The system-optimal with recourse problem is to find $\mathbf{y}$ minimizing the TETT:

$$TETT = \sum_{(u,v) \in Z^2} \sum_{\pi \in \Pi_v} y_u^{\pi} C_u^{\pi}(\mathbf{y}) = \sum_{v \in Z} \sum_{\pi \in \Pi_v} \sum_{i \in N} y_i^{\pi} C_i^{\pi}(\mathbf{y}) \tag{12}$$

$$= \ldots \tag{13-18}$$

$$= \sum_{(i,j) \in A} \sum_{s \in S_{ij}} t_{ij}^{s}(x_{ij}^{s}) x_{ij}^{s} \tag{19}$$

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

$$\min_{\mathbf{y}, \mathbf{x}, \mathbf{x}^{\pi}, \mathbf{b}^{\pi}} TTET = \sum_{(i,j) \in A} \sum_{s \in S_{ij}} t_{ij}^{s}(x_{ij}^{s}) x_{ij}^{s} \quad (SOR) \tag{20}$$

$$s.t. \begin{cases} \sum_{\pi \in \Pi_v} y_u^{\pi} = d_{uv} & \forall (u, v) \in Z^2 \\ \sum_{\pi \in \Pi} x_{ij}^{s\pi} = x_{ij}^{s} & \forall (i, j) \in A, s \in S_{ij} \\ \mathbf{A}_{\pi} \mathbf{x}^{\pi} = \mathbf{b}^{\pi} & \forall\, \pi \in \Pi \\ y_y^{\pi} \geq 0 & \forall\, \pi \in \pi, u \in Z \end{cases} \tag{21-24}$$

The SOR state is such that all routing choices are made to minimize expected travel time for the entire system. But this won't arise spontaneously, as **1)**drivers do not have enough information to determine which RP to follow and **2)**there's no incentive to do so if that information is informed. Instead, the UER state will be achieved.

$$\min_{\mathbf{y},\mathbf{x},\mathbf{x}^{\pi},\mathbf{b}^{\pi}} \sum_{(i,j)\in A} \sum_{s\in S_{ij}} \int_0^{x_{ij}^s} t_{ij}^s(x)dx \quad (UER) \tag{25}$$

$$s.t. \begin{cases} \sum_{\pi\in\Pi_v} y_u^{\pi} = d_{uv} \quad \forall (u,v) \in Z^2 \\ \sum_{\pi\in\Pi} x_{ij}^{s\pi} = x_{ij}^s \quad \forall (i,j) \in A, s \in S_{ij} \\ \mathbf{A}_{\pi}\mathbf{x}^{\pi} = \mathbf{b}^{\pi} \quad \forall\, \pi \in \Pi \\ y_y^{\pi} \geq 0 \quad \forall\, \pi \in \pi, u \in Z \end{cases} \tag{26-30}$$

Luckily, 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. 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:

$$\hat{t}_{ij}^s(x_{ij}^s) = t_{ij}^s(x_{ij}^s) + x_{ij}^s(t_{ij}^s)'(x_{ij}^s)$$

4 Solution methods

Caution:

[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.

For the Convex Program in Eqs.[20, 25], utilizing **Frank-Wolfe algorithm**. The sub-problem of finding the optimal routing policies for an all-or-nothing assignment within each FW iteration can be solved by **TD-OSP**(Waller2002). But, with reset assumption, **problem remains solving: cycling in the optimal RP**.

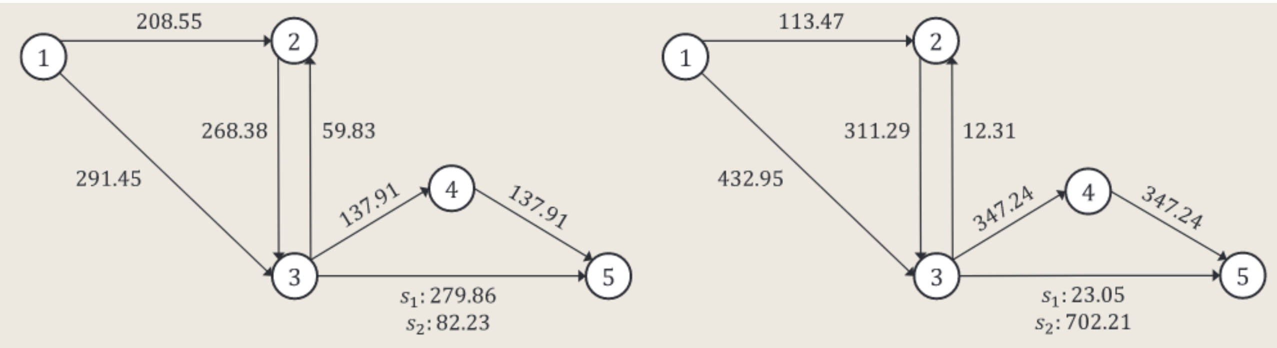

Fig. 4. Optimal flows (left) and tolls (right) in a network that illustrates cycling.

Two methods can be analyzed:

1. ~~Assuming no-reset. But this is not realistic.~~
2. Assuming travelers choose only acyclic policies. This is the breach.

However, it's proved that **Acyclic OSP is NP-hard**. At first authors 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). Nevertheless, two issues occur:

- Similar to logit-based SUE, the equilibrium limits to the subgraph made up with reasonable links.
- Each iteration the descent direction may be non-optimal or even worse than the starting point.

Instead, **cycles above a certain length** is permitted to address these, and this is achieved by modifying the state of the traveler to include a vector of m most recently visited nodes in addition to the node-message pair. This variant of the SOR and UER problems are referred to as m-UER and m-SOR. States of travelers expand which results a larger transition matrix, and if doing so, previous model needs adjusting with some difficulty. Instead, note that **acyclic graphs do not involve cyclic policies**, we could directly transform the network itself.

Transformation of network

Phase I

First add a dummy node X and connect it to all the nodes in the network including itself(i.e., a loop). Let $M_i(j)$ represent the set of nodes which can reach node i by traversing at most j arcs. Then to enumerate the set of all the possible recently visited node M_i utilizing the algorithm below:

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 the transformed network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \cup_{i \in N} M_i \cup M$ and $M = N$. 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. Let $\mathcal{A}_{ij} \subset \mathcal{A}$ represent the set of arcs in $\mathcal{A}$ which share the same parent arc. Finally, dummy arcs with one state of zero free flow travel time and infinite capacity are created in $\mathcal{G}$ to connect nodes in M_i and $i \in M$. The subset of nodes $M \in N$ play the role of destinations and the nodes $\{(i, X, \ldots, X) : i \in N\}$ serve as origins.

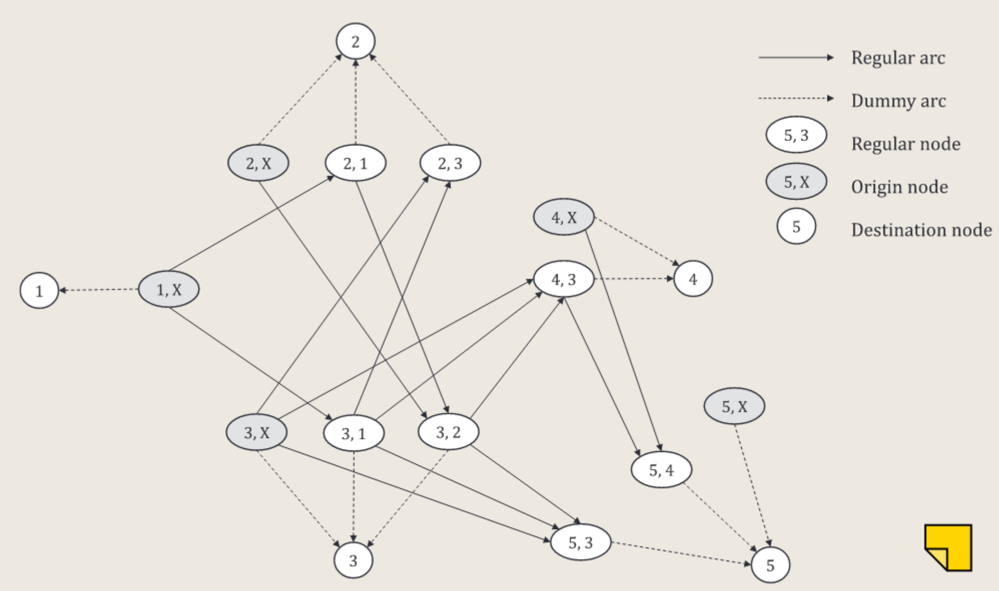

Fig. 6. Network transformation to restrict cycling.

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**. Up to now, FW algorithm can be conducted on the transformed network.

1-SOR problem can be used to eliminate cycling between nodes 2 and 3. **However, note that there is a wide variation in tolls when compared to the 0-SOR model**.

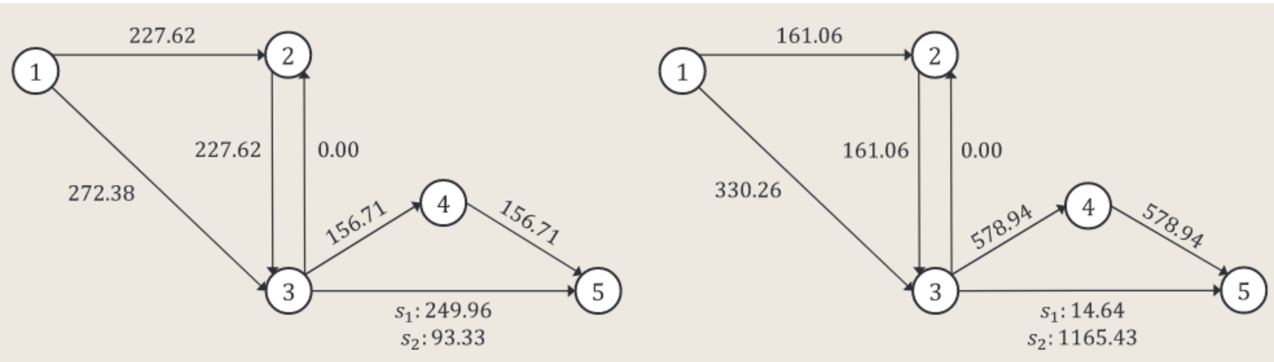

Fig. 7. Optimal flows (left) and tolls (right) for the 1-SOR problem.

5 Demonstration

BASIC	ILLUSTRATION
Network Name	Sioux Falls
Link States	State 1 and state 2 with probability 0.9 and 0.1
Changes in States	Disrupted state: a 50% reduction in capacity
The ϵ value	10^{-4}
Range of m	0,1,2 and 3
Network type	When m=1,2, 3 choose the transformed network

Check Points

- TETT values for m=0 and m=1 are significantly different while the difference between the TETT of variants of larger m is minimal.
- Static marginal tolls(SMTs) will lead to suboptimal system performance or even an increase in the TETT when compared to the no-tolls (UER). SMTs are calculated using a traditional system optimum model with expected link capacities.

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 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

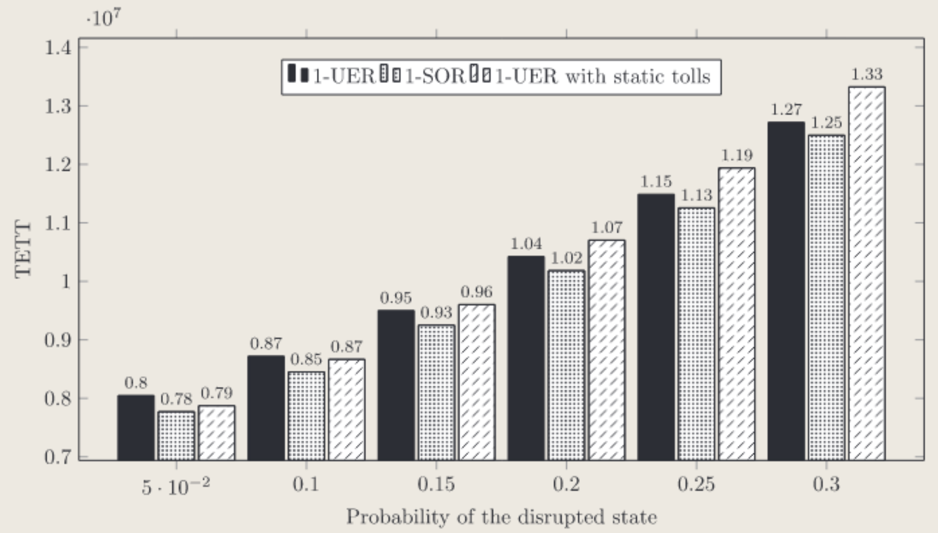

Fig. 10. Impact of static and state-dependent marginal tolls for different disruption probabilities.

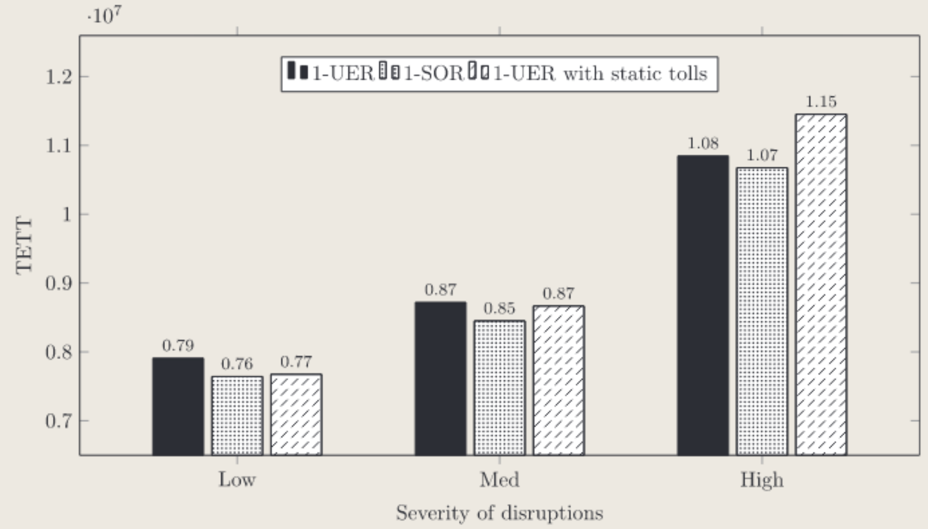

Fig. 11. Impact of static and state-dependent marginal tolls for different levels of disruption severity.

6 Insights

- Consider multiple user classes with different VOTs.
- Multi-step TD-OSP or SD-OSP may be considered.
- Consider mixed-user, that is, only a fraction can obtain/utilize en-route information. Or policy-makers only disseminate information about certain links.
- Extend to UER to SUER(SOR)

Notes

1 Note about m-SOR

需要正确理解m的含义。m+1表明一个RP如果包含循环，那么这个循环包含的弧数量必须大于m+1。如m=1，就能够避免一个包含2条弧的循环，但是一个包含3条弧的循环则不能保证。文中说到:

> *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.*

这一假设解决了m理论上能取到无穷大的窘境。也就是说，当m取到可能5或者6的时候，模型可以近似认为实现了acyclic的RP。TD-OSP算法(Waller2002)求得的最优RP实际上包含cyclic，并且对应文章中的例子故意地设置了一个无循环图作为例子。作者认为该假设不合理，意图在不改变各个算法实现的基础上避免cyclic的出现，故进行了Network transformation。总的来说，0-SOR的求解直接用原网络，其余m-SOR则需要Network transformation，但是使用的算法都是一样的。

2 VOT

> *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.*

文章假设所有旅行者的时间价值（Value of Time, VOT）是相同的，并且通过选择合适的收费单位，使得每位旅行者的时间价值等于1。这种假设简化了模型的分析和计算，因为不需要考虑不同个体之间的时间价值差异。

1. **统一的时间价值**：假设所有旅行者的时间价值相同，意味着不考虑个人差异，比如收入水平、时间紧迫性等对时间价值的影响。
2. **单位选择**：将收费单位设定为使每位旅行者的时间价值等于1，意味着在模型中，时间和金钱之间的转换率被标准化。例如，如果时间价值为1，意味着节省1单位时间等同于节省1单位金钱。

在这种情况下，旅行者在选择路线时，只需比较“总成本”即可，因为时间和金钱已经用相同的单位表示。

3 Fundamental matrix

> *Note that the columns of A^{π}_{-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.*

在马尔可夫链理论中，状态可以分为**吸收态**和**非吸收态**。

- **吸收态**：一旦进入该状态，系统就无法离开。也就是说，从吸收态转移到其他状态的概率为零。
- **非吸收态**：系统可以在这些状态之间转移，并且有可能最终进入吸收态。

非吸收态之间的转移概率矩阵，通常用符号 (Q) 表示，是描述系统在非吸收态之间的转移概率的矩阵。具体来说，矩阵 Q的元素 (Q_{ij}) 表示系统在状态i时，转移到状态j的概率，其中i和 j 都是非吸收态。这个矩阵的重要性在于：

- **分析系统行为**：通过 (Q) 矩阵，我们可以研究系统在进入吸收态之前的动态行为。
- **计算基本矩阵**：利用 (Q)，我们可以计算**基本矩阵** ($N = (I - Q)^{-1}$)，其中 (I) 是单位矩阵。基本矩阵 (N) 的元素 (N_{ij}) 表示从非吸收态 (j) 出发，预期会经过非吸收态 (i) 的总次数。