Network Optimized Congestion Pricing: A Parable, Model and Algorithm

PB97-151344

May 1995

TMIP

Travel Model Improvement Program

Department of Transportation
Federal Highway Administration
Federal Transit Administration
Office of the Secretary

Environmental Protection Agency

Department of Energy

U.S.Department of Transportation

U.S.Environmental Protection Agency

Technical Report Documentation Page

1. Report No.	2.	3. Recipient's Catalog No.
DOT-T-95-20	PB97-151344	

4. Title and Subtitle	5. Report Date
Network Optimized Congestion Pricing: A Parable, Model and Algoithm	May 1995
	6. Performing Organizational Code
7. Author(s) Robert B. Dial	8. Performing Organization Report No.
9. Performing Organization Name and Address Volpe National Transportation System Center Kendal Square Cambridge, MA 02142	10. Work Unit No. (TRAIS)
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Transit Administration Federal Highway Administration Office of the Secretary US Environmental Protection Agency	13. Type of Report and Period Covered Final Report May 1995
	14. Sponsoring Agency Code **DOT/RSPA**

15. Supplementary Notes

Released in cooperation with the U.S. Department of Transportation's Technology Sharing Program

16. Abstract

This report recites a parable, forumlates a model and devises an algorithm for optimizing tolls on a road network. Such tolls induce an equilibrium traffic flow that is at once system-optimal and user-optimal. The parable introduces the network-wide congestion-pricing problem; the model permits the marginal value of time to be a random variable having a different distribution for each origin-destination pair; and the algorithm provides the ability to determine the best toll for each link in the network.

17. Key Words	18. Distribution Statement
traffic; congestion; tolls	**Availability is unlimited. Document is being released for sale to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161**

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	**Unclassified**	36	

Form DOT F 1700.7 (8-72) Per FORM PRO/Delrina 04/04/94 Reproduction of completed page authorized

Network Optimized Congestion Pricing: A Parable, Model and Algorithm

Final Report
May 1995

Prepared by
Robert B. Dial
Volpe National Transportation Systems Center
Kendall Square
Cambridge, MA 02142

Prepared For
US Department of Transportation
Federal Transit Administration
Federal Highway Administration
Office of the Secretary
US Environmental Protection Agency

Distributed in Cooperation with
Technology Sharing Program
Research and Special
Programs Administration
US Department of Transportation
Washington, DC 20590

DOT-T-95-20

Abstract

This paper recites a parable, formulates a model and devises an algorithm for optimizing tolls on a road network. Such tolls induce an equilibrium traffic flow that is at once system-optimal and user-optimal.

- The parable introduces the network-wide congestion-pricing problem, intending to emphasize the significance of the variability of users' value of time and the importance of not restricting tolls to certain links *a priori.*
- The model permits the marginal value of time to be a random variable having a different distribution for each origin-destination pair. Assuming each trip uses a path that minimizes its own particular perception of generalized cost, it shows what economists have always known: to minimize the total perceived cost of time, the best toll for a link is its expected value of the social component of its marginal cost.
- The algorithm provides what transportation planners have never had: the ability to determine this best toll for each link in the network. Being both space- time-efficient, it can solve networks with thousands of nodes in reasonable time on a 486-class PC.

Contents

List of Figures

1 Introduction

In their classic *Studies in the Economics of Transportation*, Beckmann, McGuire and Winsten end their discussion on road tolls with:

> One can convince himself that the charging of suitable, discriminatory tolls [...] could result in an efficient utilization of roads [...]. But the proper choice of such tolls is a formidable problem.[3]

This paper solves their "formidable problem." Under common assumptions regarding user-optimal behavior [17], it shows what economists have always known: the best toll for a link is its expected value of the social component of its "marginal cost." It also provides what economists have never known: an algorithm for determining the best tolls—where to impose them and how much to charge.

Background. Road pricing [11, 14, 16] is no longer unamerican. A political outcast for thirty years, it suddenly emerged four years ago as a player in U. S. transportation policy. Federal law prescribes it: the Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA) orders the U. S. Department of Transportation (DOT) to execute up to five congestion pricing pilot projects and earmarks $125 million to fund them. While Europe and Asia showed interest in the subject for decades, U.S. DOT had previously never sponsored a single significant road-pricing project.

Regrettably, as current events teach and re-teach, legislation only highlights problems; it never solves them. Money is always necessary but seldom sufficient. Intelligence is also required. Before congestion pricing can solve traffic problems, transportation planners and engineers must know how, when, where, and at what price to levy tolls on urban roads. Until now, the four years since ISTEA had not provided means to answer these questions.

Relationship to Prior Work. As shown below, the optimal tolls problem is equivalent to a bicriterion traffic assignment problem, which uses disaggregated probability distributions for the value of time and has time and cost both being flow-dependent. The only formulation and solution to this latter problem are in [6], which bites off a more general problem and proposes an elementary (i.,e., path-based) solution algorithm. This paper, besides providing a more efficient (tree-based) algorithm, is a specific and self-contained application of that paper's model to congestions pricing.

Before delving into the technutiae of model formulation and algorithm design, we introduce the problem with a parable.

1.1 A Toll for Two Towns

The Governor was in a huff when he called Johnstown's City Manager. "Ed," he growled, "my traffic engineers tell me Johnstown's local traffic reduces the turnpike's speed from 55 to 25 mph, causing a shock wave that wreaks havoc on upstream traffic. Johnstown residents use the turnpike for free. This must stop. From now on, I want you to charge them a toll, one that is high enough to deter enough traffic that the turnpike speed through Johnstown is at least 45 miles an hour. You got that?" Hearing Ed gulp, "yes, sir," the Governor hung up without saying good bye.

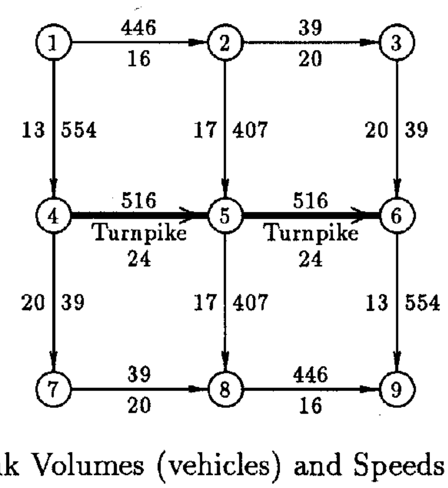


Figure 1: Johnstown (Marysville) Traffic Before Tolls

Ed unrolled his traffic engineer's map of Johnstown's roads, showing local traffic volumes and speeds (Figure 1). The only local trips in Johnstown were 1000 cars going from the HAL corporate headquarters at the northwest corner (node 1) to the Country Club at the southeast corner (node 9). Running east-west, the Turnpike bisected Johnston. The map showed 516 of these cars on each of the turnpike links, (4→5) and (5→6). Their speed was 24 mph, just as the Governor had said.

Ed knew that a toll would reduce turnpike usage; however, he did not know the toll's best value. If too small, the toll would discourage too few trips: the turnpike speeds would not increase to 45. If too large, it would discourage too many trips: the revenue would not even pay for toll collection. Math problems always gave him a headache. Accordingly, Ed did what he always did with such conundrums. He consulted the Oracle.

After asking the Oracle what the toll should be on each of the turnpike links, Ed then answered her questions: he told her about the 1000 trips going from node 1 to node 9, and he presented her with his traffic engineer's "BPR curves" in Figure 2 relating each link's traversal time to its traffic volume.

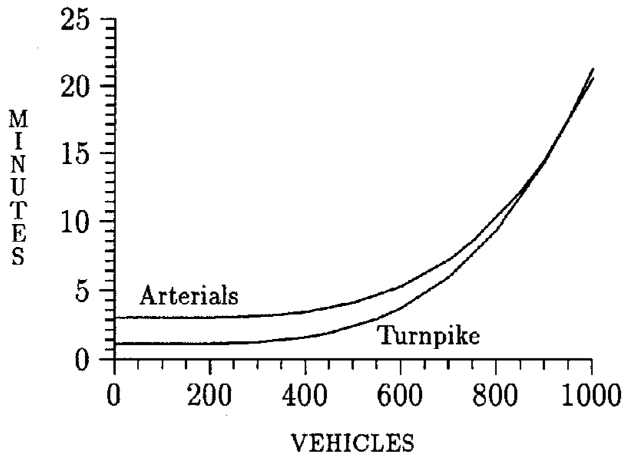


Figure 2: Link Time vs Volume (BPR)

Satisfied, the Oracle stared hard into her crystal ball and while in her trance, drew the picture appearing in Figure 3:

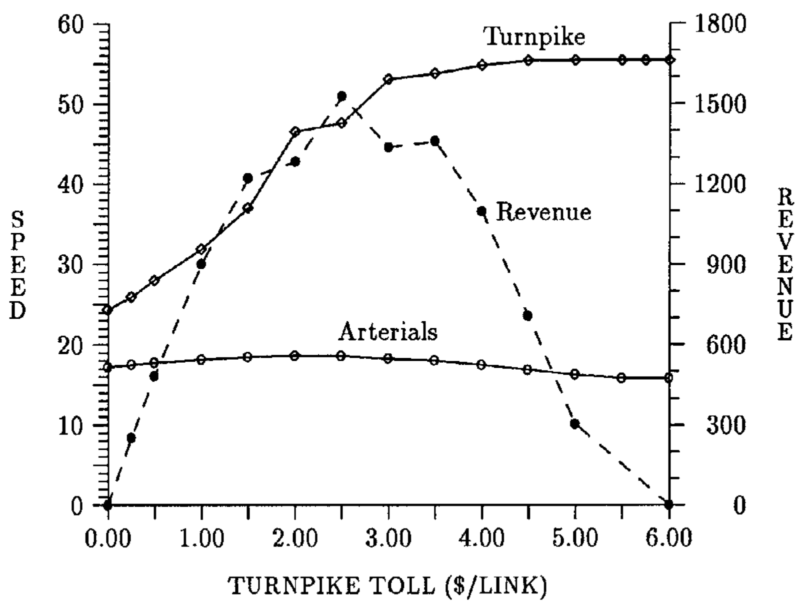


Figure 3: Johnstown Toll Impact: Speeds (mph) and Revenue ($)

The Oracle explained to Ed that her picture showed speeds that resulted from various toll levels. "You can see," she said as Ed fidgeted, "that as tolls rise to six dollars, the speed on the turnpike rises from the current 24 mph up to the 55-mph speed limit. With a toll of two dollars, you will achieve your target speed of 45 along with total toll revenue of $1,280 per day. The other, arterial links' average speed will also improve, though less dramatically . . ."

But by then, Ed had thrown down the Oracle's five-dollar fee, darted from the room, and was running back to his office. There, he ordered construction of two toll plazas in the grand monumental style the Governor favored. When the plazas were complete and christened, local travelers began paying two dollars to use each of the turnpike links. The resulting traffic flows and speeds settled down to a consistent average, shown in Figure 4. Ed was not surprised that the 45-mph turnpike speed and the $1,280 revenue matched the Oracle's prediction exactly. She was always right.

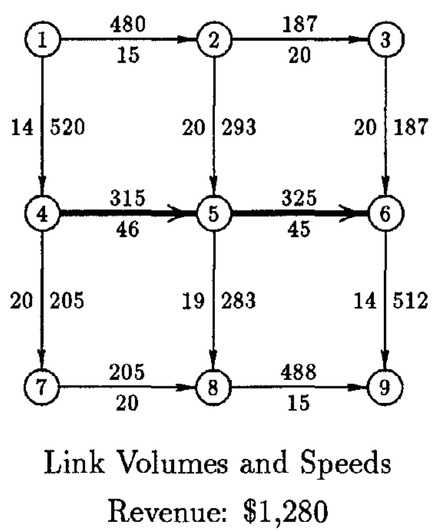


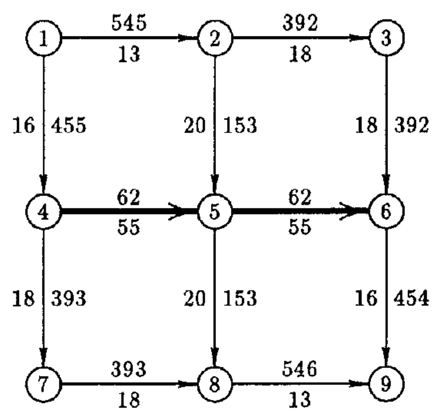
Figure 4: Johnstown Traffic (Turnpike Toll: $2.00/Link)

The next day, the Governor phoned: "Ed, my boy, you're a genius! You raised those speeds right to where I wanted them. I've received hundreds of calls of congratulations. By the way, how *did* we come up with just the right tolls?" "Just good common sense, sir," Ed replied. He went on to talk about his shrewd managerial style, but by then the Governor had hung up.

The phone rang again. It was the Chairperson of the Board of Supervisors of Marysville, the town on Johnstown's eastern border, just downstream on the turnpike. The Chairperson offered Ed the job of Marysville City Manager, at twice his current salary, to figure out tolls for the turnpike in Marysville. "You see," pleaded the Chairperson, "I received this call this morning from the Governor . . ." Ed accepted immediately.

Ed already knew that Marysville's network was an exact copy of Johnstown's. To his amazement, its local traffic was also identical: precisely 1000 cars went from its node 1 to node 9. Conditions mimicked those in Figure 1 perfectly.

With Marysville's network, travel demand, and traffic duplicating Johnstown's, Ed reasoned not to waste money on the Oracle. He went ahead and built the toll plazas. The day they were complete, he levied a two-dollar toll on Marysville's turnpike links. Then, sitting back in his leather chair and admiring his Matisse original, he awaited the down pore of revenue and praise.



Link Volumes and Speeds

Revenue: $248

Figure 5: Marysville Traffic (Turnpike Toll: $2.00/Link)

He waited in vain. When his traffic engineer brought in the map shown in Figure 5, Ed saw that the $2.00 toll in Marysville resulted only 62 trips using the turnpike. Instead of the expected torential $1,280 of revenue, only $248 trickled in. This would not even cover the toll plazas' debt service. Before Ed had finished his morning cappuccino, the Chairperson fired him.

Unemployed and confused, Ed returned to the Oracle, who nodded slowly as he told her about the 62 trips. "Everything was the same," Ed whined. "What happened?" He slid forward a five-dollar bill.

The oracle looked into her crystal ball and drew another picture like the one she had done for Johnstown (Figure 6). "Yes," she explained, moving her bony nicotine-stained finger along the dashed line," your $2.00 toll had the expected result. In fact, the highest revenue you could hope to receive from turnpike tolls in Marysville is only about $425, from a toll of $1.00 per link."

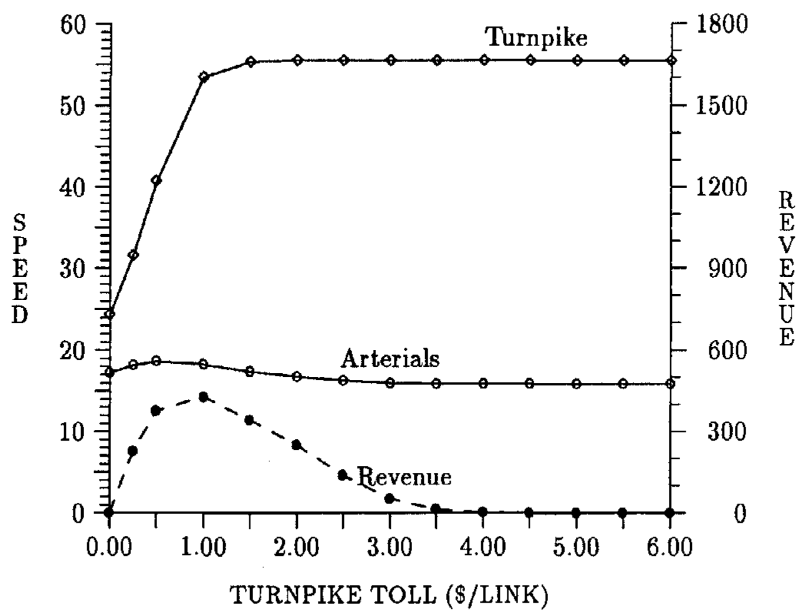


Figure 6: Marysville Toll Impact: Speeds (mph) and Revenue ($)

As usual the Oracle's explanation befuddled Ed; now it irritated him. "Enough, already, of your complicated charts! I *know* how much revenue I got—or didn't get. Just tell me why I didn't get more. Why I'm out of a job."

Accepting another five dollars, the Oracle returned to her crystal ball. Again, she drew a picture, this one having two curves (Figure 7). While Ed drummed his fingers and chewed his lips, she explained: "These probability densities show how much the travelers in Johnstown and Marysville value their travel time. Note that compared to Johnstown, Marysville is skewed way to the left. Johnstown's median value of time is 50 cents per minute, while Marysville's is only about 15 cents. Seeing Ed's eyelids droop , she softened her tone, gently patted his hand, and said, "Marysville is K-Mart; Johnstown is Gucci. This difference is the whole reason their revenue isn't the same."

Head pounding with pain, Ed staggered from the room and, in pursuit of a lucrative mid-life career change, enrolled in a computer programming course.

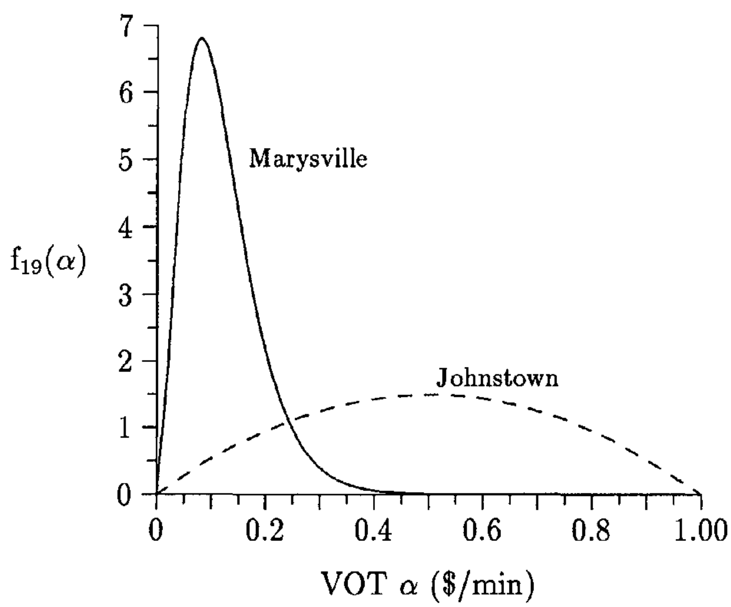


Figure 7: Johnstown and Marysville VOT Distribution

The Oracle continued to study her crystal ball. After jotting some notes, she phoned Marysville to offer her services. The Chairperson gratefully hired her. When the Oracle recommended the tolls in Figure 8 and shared her predictions, the Chairperson instituted them at once. (She also tore down the toll plazas, but that is another story.)

The results appear in figure 8(b). More links had tolls, the average toll per trip was higher, and at least ten frugaloids even drove from east to west—on links (6→5) and (5→4)—to reduce toll costs. Nevertheless, the Marysville residents were pleased. They had more path-cost options, and for most, trip time was faster. The Governor gushed: turnpike speeds were 47 mph. The Chairperson gloated: daily revenue exceeded $3,000—nearly triple snooty Johnstown's. The Oracle yawned: her predictions were correct, as usual.

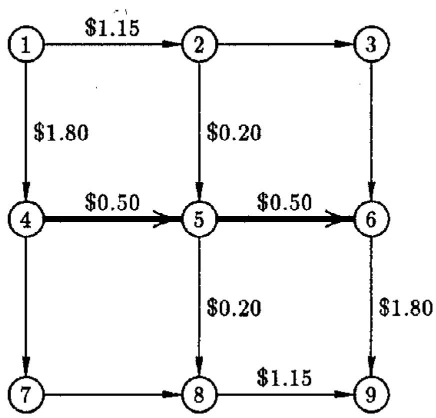


Figure 8(a): Link Tolls

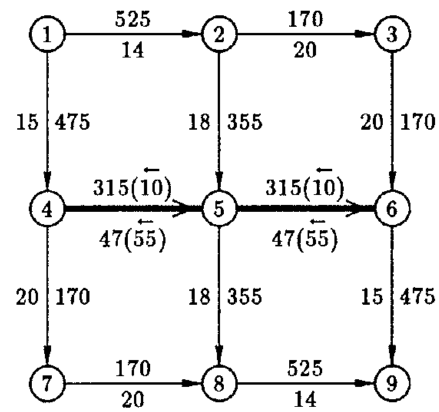


Figure 8(b): Volume and Speeds

Figure 8: Marysville Oracle-Priced Tolls and Traffic

Only Ed remained unhappy. Frustrated and unemployed, having repeatedly failed his computer programming course, he would lie awake all night on his park bench wondering how the Oracle did it. Once, in the haze of approaching dawn, her image materialized above him. "I minimized the total perceived cost of time," she whispered, then faded away before he could ask what in the world she was talking about. As his life withered away in destitution, Ed would never come to understand the lessons his Oracle had tried to teach him. Through the weeds overgrowing his tombstone, one reads:

R.I.P.

Two steadfast truths
Ye must realize:
Each trip has its price
For time that it buys,

And only the fool
Sets tolls on his roads
Before hearing what
The Oracle bodes.

Beneath this cold stone
Lies toll planner Ed.
These truths he ignored,
And that's why he's dead.

Figure 9: Ed's Tombstone

2 The Model

Given the following inputs:

- network topology (e.g., Figure 1),
- volume-time function for each link in the network (e.g., Figure 2),
- travel-demand matrix, and
- value-of-time (VOT) probability density function (PDF) for each origin-destination pair (e.g., Figure 7),

the model predicts the "social-optimizing" toll and resultant traffic volume for each link in the network. Qualitatively, the model's results are what an economist would expect from marginal cost pricing: congested bottleneck links have the highest tolls, and links with no congestion are free.

2.1 Preview

The model is a stochastic multi-commodity non-linear network optimization that, subject to flow conservation and nonnegativity constraints, minimizes the total perceived cost of time V, where

$$V(\mathbf{x}) = \sum_{e} \overline{\alpha}_e x_e t_e(x_e) \tag{1}$$

and $\mathbf{x}$ is the link-flow decision variable. The total traffic volume on link e is x_e and $t_e(x_e)$ is the link's travel time (min) at that volume. The average value-of-time ($/min) of the trips actually using link e is $\overline{\alpha}_e$.[1] The tolls themselves do not appear explicitly in the objective function:

> From the point of view of the community, the tolls do not constitute costs but are available again for redistribution ....[3]

Tolls do, however, enter implicitly. Each link e's volume x_e depends on path choices that depend on both time and toll.

Link Costs. Besides the link's volume-dependent time function t_e, we will associate with each link a cost (toll) function c_e, which depends on the link's total volume and the average value-of-time among trips composing that volume. Assuming that a trip with VOT α "prices" a path p of time t_p and total toll c_p as:

$$\begin{aligned} g_p(\alpha) &= \alpha t_p + c_p \\ &= \sum_{e \in p} (\alpha t_e(x_e) + c_e) \end{aligned} \tag{2}$$

[1]This paper uses the term value-of-time, or simply VOT, without qualification to mean the *marginal* value of time.

and always uses a path with the lowest price, or "generalized cost", we will show that if link e's toll c_e is the "marginal social cost" of the link:

$$c_e = \overline{\alpha}_e t'_e(x_e)x_e, \tag{3}$$

then the equilibrium flow of a bicriterion traffic assignment minimizes (1). Therefore, the value of a link's cost function at its user-optimal flow decides its system-optimizing toll.

Organization of this Section. The model's detailed formulation begins with a listing of all its objects: sets, input data, decision variables, and state variables along with their relationships (constraints). Next, we formulate necessary and sufficient conditions for a user-optimal equilibrium bicriterion traffic assignment, and finally show them equivalent to those signifying a minimum of (1).

2.2 Sets

$$\begin{aligned}
\Re_+ &= \{\text{nonnegative real numbers}\} \\
\mathcal{A} &= \{\text{(marginal) values of time (VOT) } \alpha \in \mathcal{R}_+\} \\
\mathcal{N} &= \{\text{network nodes}\} \\
\mathcal{E} &= \{\text{network links } e = (i_e, j_e) \in \mathcal{N} \times \mathcal{N}\} \\
\mathcal{X} &= \{\text{feasible traffic assignments}\}.
\end{aligned}$$

All the above is self-explanatory except $\mathcal{X}$, whose definition must wait until we discuss the decision variable $\mathbf{x}$.

2.3 Inputs

$$\begin{aligned}
\mathcal{G} &= \text{the network: } \{\mathcal{N}, \mathcal{E}\} \\
t_e &= \text{time function for link } e \in \mathcal{E} \\
f_{od} &= \text{VOT PDF of trips going from } o \in \mathcal{N} \text{ to } d \in \mathcal{N} \\
v_{od} &= \text{total trips going from } o \in \mathcal{N} \text{ to } d \in \mathcal{N}.
\end{aligned}$$

The Network. The network $\mathcal{G}$ contains and connects all nodes o and d referenced in the trip matrix $((v_{od}))$ as well as other, "transshipment" points.[2] Defining the network's topology, some pairs of these nodes are connected by links (i.e., directed arcs). Each link has its own time and cost functions, which contribute to the disutility each trip bears to use the link. Both functions are volume-dependent.

[2] To avoid distinguishing between these node types, we may assume $((v_{od}))$ is a $|\mathcal{N}|$-by-$|\mathcal{N}|$ matrix with zeros in the rows and columns of transshipment nodes.

Link Time Function ($t_e : \Re_+ \to \Re_+$). This function maps the total volume on link e into minutes to traverse the link. It is a nondecreasing nonnegative function of link e's total volume. That is, more traffic implies more time, and all trips traverse link e in the same amount of time, whatever their value of time. In addition, t_e is assumed continuous, twice differentiable, and convex.

VOT PDF ($f_{od} : \mathcal{A} \to \Re_+$). The value-of-time probability density function f_{od} may have any shape whatever—continuous or discrete in any combination—provided it maps and $\alpha \in \mathcal{A}$ into $\Re_+$ and integrates to one.[3] As indicated, each origin–destination pair may have its distinct VOT PDF. The PDF's degree of disaggregation, while crucial for accurate predictions, neither negates the veracity of the model nor cripples the performance of the algorithm.

O-D Trips (v_{od}). The trip matrix $((v_{od}))$ gives the *fixed*, nonnegative number of trips between each o–d pair. (Note that demand is assumed inelastic. While the model readily extends to include elastic demand, this generalization does not appear in this paper.) The number of trips with VOT α going from node o to node d can be written as:

$$v_{od}(\alpha)d\alpha = v_{od}f_{od}(\alpha)d\alpha \tag{4}$$

which would be interpreted as the number of trips from o to d whose VOT is in the interval $(\alpha, \alpha + d\alpha]$. For brevity, however, we henceforward drop the infinitesimal notation $d\alpha$ for all such VOT-disaggregated variables.

Most important, the VOT PDF $f_{od}(\alpha)$ facilitates disaggregating the total trips from o to d into those trips with a VOT in any interval $(\alpha^{lb}, \alpha^{ub}]$:

$$\begin{aligned} v_{od}(\alpha^{lb}, \alpha^{ub}) &= \text{trips from } o \text{ to } d \text{ with VOT between } \alpha^{lb} \text{ and } \alpha^{ub} \\ &= v_{od}\mathbf{Prob}[\alpha^{lb} < \alpha \leq \alpha^{ub}] \\ &= v_{od}\int_{\alpha^{lb}}^{\alpha^{ub}} f_{od}(\alpha)d\alpha. \end{aligned}$$

2.4 Decision Variable

The model formulated here has only one decision variable, comprising all the VOT- and origin-disaggregated link volumes. Let

$$x_{oe}(\alpha) = \text{trips using link } e \text{ that originate at node } o \text{ and have VOT } \alpha.$$

Then intuitively the decision variable can be imagined as an "infinite" three-dimensional matrix

$$\mathbf{x} = (((x_{oe}(\alpha)))) \in \Re_+^{|\mathcal{N}|\times|\mathcal{E}|\times|\mathcal{A}|}. \tag{5}$$

[3] In this paper, the integral sign always means Riemann-Stieltjes integration.

The cardinalities of $\mathcal{N}$ and $\mathcal{E}$ are finite, but $\mathcal{A}$'s is typically infinite. The notation $\mathbf{x}_e$ means the projection of $\mathbf{x}$ with respect to link e. Despite the model's seeking optimal link *tolls*, its only decision variable is link *volumes*, whose optimal value determines the tolls.

2.5 State Variables

Five state variables aid the model's formulation and its solution algorithm. All are link variables and functions of their decision-variable component $x_{oe}(\alpha)$.

$$\begin{aligned}
x_e(\alpha) &= \sum_{o\in\mathcal{N}} x_{oe}(\alpha) &&= \text{trips with VOT } \alpha \in \mathcal{A} \text{ using link } e \in \mathcal{E} && (6)\\
x_e &= \int_{\mathcal{A}} x_e(\alpha)d\alpha &&= \text{total trips using link } e \in \mathcal{E} && (7)\\
u_e &= \int_{\mathcal{A}} \alpha x_e(\alpha)d\alpha &&= \text{“first moment” of } \alpha \in \mathcal{A} \text{ on link } e \in \mathcal{E} && (8)\\
\overline{\alpha}_e &= \frac{u_e}{x_e} &&= \text{mean VOT for all trips using link } e && (9)\\
c_e &= \overline{\alpha}_e x_e t_e'(x_e) &&= \text{marginal social cost of using link } e \in \mathcal{E}. && (10)
\end{aligned}$$

Total Trips Using Arc e: (x_e). The total volume on link e determines the time to traverse the link. It also decides in part the cost to use the link. Essentially, x_e is the same variable found in conventional traffic assignment, although here it is more difficult to compute.

Mean VOT of Trips Using Arc e: ($\overline{\alpha}_e$). The mean value-of-time for trips using link e is an interesting state variable. Besides being in the model's objective function, it reduces the problem to finite dimension. If there is no volume on link e, that is, $x_{oe}(\alpha) = 0$ for all $o \in \mathcal{N}$ and $\alpha \in \mathcal{A}$, then $\overline{\alpha}_e$ is undefined; however, this will pose no problem to the model or its algorithm.

First Moment of α on Arc e: (u_e). This state variable will prove to be invaluable. It is a close relative of the average value-of-time on link e:

$$u_e = \overline{\alpha}_e x_e.$$

The state variable u_e will occupy a refined version of the model's objective function (1): and thus plays a leading role in the solution algorithm.

Arc Cost: ($c_e : \Re_+ \times \Re_+ \rightarrow \Re_+$). Arc cost c_e is the dollar cost to use link e—a toll, which applies equally to all users of that link. It is a function of the link's total volume x_e and average value-of-time $\overline{\alpha}_e$. We can use the state variable u_e to rewrite (3) as:

$$c_e = u_e t_e'(x_e). \tag{11}$$

It will be seen momentarily, that the state variable c_e is just the "social component" of the link's (total) marginal (generalized) cost. Its value at the user-optimal equilibrium is the desired system-optimal toll. That is,

$$c_e^{opt} = u_e^{opt} t'_e(x_e^{opt}). \tag{12}$$

Where naturally

$$\begin{aligned} t'_e &= \frac{\partial t_e}{\partial x_e} \\ x_e^{opt} &= \int_{\mathcal{A}} \sum_o x_{oe}^{opt}(\alpha) d\alpha &= \int_{\mathcal{A}} x_e^{opt}(\alpha) d\alpha \\ u_e^{opt} &= \int_{\mathcal{A}} \alpha \sum_o x_{oe}^{opt}(\alpha) d\alpha = \int_{\mathcal{A}} \alpha x_e^{opt}(\alpha) d\alpha. \end{aligned}$$

2.6 Constraints

As mentioned earlier, the model's only decision variable is $\mathbf{x} = (((x_{oe}(\alpha))))$. Besides being nonnegative, $\mathbf{x}$ must satisfy a system of flow conservation constraints, which guarantees that all trips only use paths connecting their origin to their destination, and that these trips account exclusively for all link volumes. Formally, for $o \in \mathcal{N}$, and $\alpha \in \mathcal{A}$:

$$\sum_{\{e \in \mathcal{E} | j_e = d\}} x_{oe}(\alpha) - \sum_{\{e \in \mathcal{E} | i_e = d\}} x_{oe}(\alpha) = v_{od}(\alpha), \quad \text{for all } d \in \mathcal{N}. \tag{13}$$

Any nonnegative instance of the decision variable $\mathbf{x}$ that satisfies (13) we call a "feasible traffic assignment." We denote the set of all feasible traffic assignments as $\mathcal{X}$.

2.7 Objective Function

Using the above definitions, we can restate the model's objective more conveniently: find the minimand

$$\mathbf{x}^{opt} \in \arg\min_{\mathbf{x} \in \mathcal{X}} V(\mathbf{x})$$

where

$$V(\mathbf{x}) = \sum_{e \in \mathcal{E}} u_e t_e(x_e) \tag{14}$$

and determine (c_e^{otp}) from Equation (12). The difference between (14) and (1) reflects the substitution of u_e for $\overline{\alpha}_e x_e$, which also eliminates any problems attending $\overline{\alpha}_e$ being undefined when x_e is zero.

2.8 Optimality Conditions

This section derives the optimality conditions for (14). In addition, it expresses these conditions compactly in terms of $x_e(\alpha)$ instead of $x_{oe}(\alpha)$, honoring these two variables' relationship (6) as a constraint. This compaction is possible because traffic equilibrium depends on $x_e(\alpha)$ and because the partial derivative of $x_e(\alpha)$ with respect to $x_{oe}(\alpha)$ is unity.

Theorem 1 (Convexity). V is convex, and therefore any local minimum of V is a global minimum—although there may be multiple minimands that yield the same value of V.

Proof.

$$\begin{aligned} V(\mathbf{x}) &= \sum_{e \in \mathcal{E}} u_e t_e(x_e) \\ &= \sum_{e \in \mathcal{E}} t_e(x_e) \int_{\alpha \in \mathcal{A}} \alpha x_e(\alpha) d\alpha \\ &= \sum_{e \in \mathcal{E}} \int_{\alpha \in \mathcal{A}} t_e(x_e) \alpha x_e(\alpha) d\alpha. \end{aligned} \tag{15}$$

By assumption, t_e is convex in x_e and, therefore, convex in $x_{oe}(\alpha)$. That both α and $x_e(\alpha)$ are similarly nonnegative and differentiable makes the integrand in (15) convex. Therefore V is convex.

A necessary and sufficient condition for a global minimum of the convex function V at $\mathbf{x}^{opt}$ is that all directional derivatives there be nonnegative:

$$\nabla V(\mathbf{x}^{opt})(\mathbf{x} - \mathbf{x}^{opt}) \geq 0. \tag{16}$$

The next two lemmas and theorem derive a compact reformulation of (16) and restate this condition in a form an algorithm can better understand. Along the way, a brief remark ascribes economic significance to the results.

Lemma 1 (Gradient). The gradient of V has the component

$$\frac{\partial V}{\partial x_{oe}(\alpha)} = \alpha t_e(x_e) + u_e t_e'(x_e). \tag{17}$$

Proof.

$$\begin{aligned} \frac{\partial V}{\partial x_{oe}(\alpha)} &= \frac{\partial V}{\partial x_{oe}(\alpha)} \frac{\partial x_{oe}(\alpha)}{\partial x_e(\alpha)} \\ &= \frac{\partial}{\partial x_e(\alpha)} \sum_{e \in \mathcal{E}} u_e t_e(x_e) \end{aligned}$$

$$\begin{aligned}
&= t_e(x_e)\frac{\partial u_e}{\partial x_e(\alpha)} + u_e\frac{\partial t_e(x_e)}{\partial x_e(\alpha)} \\
&= t_e(x_e)\frac{\partial}{\partial x_e(\alpha)}\left(\int_{\mathcal{A}} \sigma x_e(\sigma)d\sigma\right) + u_e\frac{\partial t_e(x_e)}{\partial x_e}\frac{\partial x_e}{\partial x_e(\alpha)} \\
&= \alpha t_e(x_e) + u_e\frac{\partial t_e(x_e)}{\partial x_e}\frac{\partial}{\partial x_e(\alpha)}\left(\int_{\mathcal{A}} x_e(\sigma)d\sigma\right) \\
&= \alpha t_e(x_e) + u_e t_e'(x_e) \\
&= \underbrace{\alpha t_e(x_e)}_{\text{private cost}} + \underbrace{\overline{\alpha}_e x_e t_e'(x_e)}_{\text{social cost}}. \qquad (18)
\end{aligned}$$

Remark. Economists call the two terms in (18) "private cost" and "social cost." Private cost is the link's time weighted by that trip's specific VOT. Social cost is the change in the link's expected total trip-minutes one additional trip causes. This latter cost becomes the link's toll, to be levied equally upon all trips using the link.

The sum of both costs is called the trip's "generalized cost," abbreviated GC. A path's GC is the sum of its links'. A path that has the smallest GC for a particular trip, we call the trip's min-GC path. At *user*-optimal equilibrium, each trip uses its min-GC path and also pays according to the *total* cost of its use: a policy that transportation economists consider equitable and efficient.

Lemma 2 (Directional Derivative). The directional derivative of V at $\mathbf{x}^o$ in the direction $(\mathbf{x} - \mathbf{x}^o)$ is

$$\nabla V(\mathbf{x}^o)(\mathbf{x} - \mathbf{x}^o) = \int_{\mathcal{A}} \sum_{e\in E} (\alpha t_e(x_e^o) + u_e^o t_e'(x_e^o))\,(x_e(\alpha) - x_e^o(\alpha))\,d\alpha. \qquad (19)$$

Proof. From Lemma 1 and the definition of a directional derivative in the context of this problem,

$$\begin{aligned}
\nabla V(\mathbf{x}^o)(\mathbf{x} - \mathbf{x}^o) &= \int_{\mathcal{A}} \sum_{e\in E}\sum_{o\in\mathcal{N}} (\alpha t_e(x_e^o) + u_e^o t_e'(x_e^o))\,(x_{oe}(\alpha) - x_{oe}^o(\alpha))\,d\alpha \\
&= \int_{\mathcal{A}} \sum_{e\in E} (\alpha t_e(x_e^o) + u_e^o t_e'(x_e^o)) \sum_{o\in\mathcal{N}} (x_{oe}(\alpha) - x_{oe}^o(\alpha))\,d\alpha \\
&= \int_{\mathcal{A}} \sum_{e\in E} (\alpha t_e(x_e^o) + u_e^o t_e'(x_e^o))\,(x_e(\alpha) - x_e^o(\alpha))\,d\alpha.
\end{aligned}$$

Theorem 2 (Optimality Conditions). A necessary and sufficient condition for a global minimum of V at $\mathbf{x}^{opt}$ is

$$\nabla V(\mathbf{x}^{opt})(\mathbf{x}-\mathbf{x}^{opt}) = \int_{\mathcal{A}} \sum_{e \in E} \left(\alpha t_e(x_e^{opt}) + u_e^{opt} t_e'(x_e^{opt})\right) \left(x_e(\alpha) - x_e^{opt}(\alpha)\right) d\alpha \geq 0. \tag{20}$$

Proof. At any local minimand $\mathbf{x}^{opt}$, all directional derivatives are non negative, and vice versa. From Lemma 2, this implies (20) for all $\mathbf{x} \in \mathcal{X}$. Since V is convex, the local minimum is also a global minimum.

We now show that (20) is satisfied iff each trip uses a path that minimizes its particular generalized cost. First, we must introduce the notion of bicriterion traffic assignment.

2.9 Bicriterion Traffic Assignment

When the trip matrix is decomposed by VOT according to the $f_{od}()$'s, and every trip is assigned to a path that minimizes its particular GC; we say we have a T2 (user-optimal) Equilibrium Traffic Assignment [6], or T2-ETA for short. "T2" denotes that two factors, i.e., cost and time, compose the generalized cost. "User-optimal" implies that all trips simultaneously use their particular min-GC path. "Equilibrium" connotes that the volume-dependent link times and costs are at levels precisely in balance with min-GC path selection and resultant link volumes. The following Lemma provides a necessary and sufficient condition for such an assignment.

Lemma 3 (T2-ETA). The flow $\mathbf{x}^{opt} = (\mathbf{x}_e^{opt})$ is a user-optimal equilibrium traffic assignment iff the following (variational inequality) holds for all $\mathbf{x} \in \mathcal{X}$:

$$\int_{\mathcal{A}} \sum_{e \in E} \left(\alpha t_e(\mathbf{x}_e) + c_e(\mathbf{x}_e)\right) \left(x_e(\alpha) - x_e^{opt}(\alpha)\right) d\alpha \geq 0. \tag{21}$$

Proof (⟹). Let c_e^{opt} and x_e^{opt} be the (albeit unknown) values of arc cost and time at equilibrium. Now consider trips for one given VOT $\alpha \in \mathcal{A}$. The total generalized trip cost at equilibrium is calculated by summing the products of the time and cost at equilibrium with the arc volumes resulting from a conventional, constant-VOT "all-or-nothing" assignment:

$$\sum_{e \in E} \left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e^{opt}(\alpha) = \min_{\mathbf{x}_e(\alpha) \in \mathcal{X}(\alpha)} \sum_{e \in E} \left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e(\alpha). \tag{22}$$

Where $\mathcal{X}(\alpha)$ is the projection of X for a particular VOT α. Equation (22) is true because at equilibrium, every trip with the same o and d use a path with identical (minimal) GCs [17]. Rewrite (22) as, for all $\alpha \in \mathcal{A}$:

$$\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e^{opt}(\alpha) \leq \sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e(\alpha). \quad (23)$$

Or

$$\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right)\left(x_e(\alpha) - x_e^{opt}(\alpha)\right) \geq 0. \quad (24)$$

Integrating both sides of (24) over $\alpha \in \mathcal{A}$ proves necessity.

Proof ($\Longleftarrow$). To prove sufficiency, assume the contrary: that (21) is true and all trips are not in equilibrium. That is, there exists some nonempty set $\mathcal{A}' \subseteq \mathcal{A}$ such that $((x_e^{opt}(\alpha)))$ is not an equilibrium flow for $\alpha \in \mathcal{A}'$. Then, letting $((x_e^{min}(\alpha)))$ be the minimand of (22), for $\alpha \in \mathcal{A}'$,

$$\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e^{opt}(\alpha) > \sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right) x_e^{min}(\alpha). \quad (25)$$

That is,

$$\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right)\left(x_e^{min}(\alpha) - x_e^{opt}(\alpha)\right) < 0 \quad (26)$$

for all $\alpha \in \mathcal{A}'$. While for $\alpha \notin \mathcal{A}'$, equilibrium prevails:

$$\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right)\left(x_e^{min}(\alpha) - x_e^{opt}(\alpha)\right) = 0. \quad (27)$$

Integrating both sides of (26) and (27) over $\alpha \in \mathcal{A}'$ and $\alpha \notin \mathcal{A}'$ respectively,

$$\int_{\alpha\in\mathcal{A}'}\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right)\left(x_e^{min}(\alpha) - x_e^{opt}(\alpha)\right) d\alpha \;<\; 0 \quad (28)$$

$$\int_{\alpha\notin\mathcal{A}'}\sum_{e\in E}\left(\alpha t_e(\mathbf{x}_e^{opt}) + c_e(\mathbf{x}_e^{opt})\right)\left(x_e^{min}(\alpha) - x_e^{opt}(\alpha)\right) d\alpha \;=\; 0. \quad (29)$$

Adding (28) to (29) contradicts (21) and completes the proof of Lemma 3.

Theorem 3 (Optimal Tolls and T2-ETA). For all $e \in \mathcal{E}$, let $\mathbf{x}_e \in \mathcal{X}$ and

$$t_e(\mathbf{x}_e) = t(x_e) \quad (30)$$

$$c_e(\mathbf{x}_e) = u_e t_e'(x_e) \quad (31)$$

and let $\mathbf{x}_e^{opt}$ be a solution to the corresponding T2-ETA problem. Then

$$c_e(\mathbf{x}_e^{opt}) = u_e^{opt} t_e'(x_e^{opt}) \quad (32)$$

solves the optimal tolls problem, and vice versa.

Proof. Under the above definitions (30) and (31) for of t_e and c_e, the directional derivative of the optimal tolls problem (20) and the variational inequality of the T2-ETA problem (21) are equivalent. Therefore, V is minimized with and only with a time-cost bicriterion user-optimal equilibrium traffic assignment (T2-ETA), and the optimal tolls are given by (32).

3 The Algorithm

This section develops procedures to compute a T2-ETA and thus solve the optimal tolls problem. Since the objective function (14) is convex and the constraints (13) comprise a closed and bounded convex set, the minimum of (14) exists, and the well-known Frank-Wolfe (FW) algorithm can find it [7]. For purposes here, we summarize FW as follows:

1. **Initialization.** Start with any feasible approximate solution $\mathbf{x}^o \in \mathcal{X}$.

2. **Direction.** Find the minimand

$$\mathbf{x}^{min} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \nabla V(\mathbf{x}^o)(\mathbf{x} - \mathbf{x}^o)$$

   of the directional derivative of the objective function at the approximate solution $\mathbf{x}^o$.

3. **Termination.** If convergence is satisfactory, quit with solution $\mathbf{x}^{opt} \approx \mathbf{x}^o$; otherwise continue to Step 4.

4. **Combination.** Replace the approximate solution with the convex combination of itself and the minimand of the directional derivative giving the smallest objective function value. That is, letting $\Delta\mathbf{x} = \mathbf{x}^{min} - \mathbf{x}^o$,

$$\mathbf{x}^o \leftarrow \mathbf{x}^o + \lambda^{opt}\Delta\mathbf{x}$$

   and

$$\lambda^{opt} = \arg\min_{\lambda\in[0,1]} V(\mathbf{x}^o + \lambda\Delta\mathbf{x}).$$

   Return to step 2.

3.1 Preview

At first blush, FW applied to (14) appears absurd: the decision variable $\mathbf{x}$ has an infinite number of components. Fortunately, however, because only $2|\mathcal{E}|$ state variables, u_e and x_e for $e \in \mathcal{E}$, appear in the objective function, we can show that:

- the algorithm operates with a descent direction in terms these $2|\mathcal{E}|$ variables, and
- the values of these variables result from a finite number of conventional min-path traffic assignments for each origin node.

The version of FW that solves our tolls problem appears in Figure 10.

Algorithm T2-ETA/Optimal Tolls

1. **Initialization.** Set arc times and costs for zero volume:

$$(c_e^o) \leftarrow 0(x_e^o) \leftarrow (u_e^o) \leftarrow 0, \quad (t_e^o) \leftarrow (t_e(0))$$

and for each $o \in \mathcal{N}$ do

$$\text{algorithm T2-MPA } (o, (x_e^o), (u_e^o))$$

in Figure 11 to obtain initial feasible state variables (x_e^o) and (u_e^o).

2. **Direction.** Set arc times and costs for current solution:

$$(t_e^o) \leftarrow (t_e(x_e^o)), \quad (c_e^o) \leftarrow u_e^o t'(x_e^o)$$

and initialize state variables

$$(x_e^{min}) \leftarrow (u_e^{min}) \leftarrow 0.$$

Then for each $o \in \mathcal{N}$ do

$$\text{algorithm T2-MPA } (o, (x_e^{min}), (u_e^{min}))$$

to obtain a direction of steepest descent:

$$(\Delta x_e) \leftarrow (x_e^{min}) - (x_e^o), \quad (\Delta u_e) \leftarrow (u_e^{min}) - (u_e^o).$$

3. **Termination.** If

$$\frac{|\sum_{e \in} (c_e^o \Delta x_e + t_e^o \Delta u_e)|}{\sum_{e \in} (c_e^o x_e^o + t_e^o u_e^o)} < \epsilon$$

then quit, with current state approximating optimal state.

4. **Combination.** Find λ^{opt} such that

$$\lambda^{opt} = \arg \min_{\lambda \in [0,1]} \sum_{e \in \mathcal{E}} ((u_e^o + \lambda \Delta u_e) t_e(x_e^o + \lambda \Delta x_e)).$$

Then update current solution:

$$(x_e^o) \leftarrow (x_e^o + \lambda^{opt} \Delta x_e), \quad (u_e^o) \leftarrow (u_e^o + \lambda^{opt} \Delta u_e)$$

and return to Step 2.

Figure 10: Algorithm T2-ETA/Optimal Tolls

Algorithm T2-MPA. We call the algorithm that does all these conventional traffic assignments for one given origin node a "T2 min-path traffic assignment" (T2-MPA). Section (3.3.1) below describes T2-MPA. At this point, it suffices to say that each of the conventional assignments uses *fixed* link costs and times, and simultaneously assigns groups of trips whose VOT falls in a given finite *range*. Nevertheless and most important, it assigns every trip in the group to a path that minimizes that trip's particular GC.

Organization of This Section. Beginning with definitions of algorithm objects, the remainder of this section derives each step in Figure 10. To avoid more forward references, we describe the Direction step first, the Combination step next, then Initialization, and finally Termination.

3.2 Definitions

For future reference, this list includes the algorithm's key objects:

$$
\begin{aligned}
o &= \text{current origin node} \\
d &= \text{current destination node} \\
\mathcal{T}_o &= \text{origin node } o\text{'s current min-path tree} \\
p_j &= \text{(unique) path in } \mathcal{T}_o \text{ from node } o \text{ to node } j \\
\pi_j &= \text{GC of path in } \mathcal{T}_o \text{ from node } o \text{ to node } j \text{ in tree } \mathcal{T}_o \\
\alpha_o^{min} &= \text{minimum (effective) } \alpha \in \mathcal{A} \text{ for origin node } o \\
&= min\{\alpha \in \mathcal{A} : \sum_{d \in \mathcal{N}} v_{od} \int_0^{\alpha} f_{od}(\sigma) d\sigma > 0\} \\
\alpha_o^{max} &= \text{maximum (effective) } \alpha \in \mathcal{A} \text{ for origin node } o \\
&= max\{\alpha \in \mathcal{A} : \sum_{d \in \mathcal{N}} v_{od} \int_{\alpha}^{\infty} f_{od}(\sigma) d\sigma > 0\} \\
\alpha^{ub} &= \text{maximum VOT for which } \mathcal{T}_o \text{ is optimal} \\
\alpha^{lb} &= \text{greatest upper bound less than } \alpha^{ub} \text{ for which } \mathcal{T}_o \text{ is } \textit{not} \text{ optimal} \\
t_e^o &= \text{link } e\text{'s current (fixed) time} \\
c_e^o &= \text{link } e\text{'s current (fixed) cost} \\
g_e^o &= \text{link } e\text{'s current (fixed) GC for a trip with VOT } \alpha^{ub} \\
&= \alpha^{ub} t_e^o + c_e^o.
\end{aligned}
$$

The VOT bounds α_o^{min} and α_o^{max} are just the smallest (exclusive) and largest (inclusive) VOTs for trips out of node o. By a tree $\mathcal{T}_o$ being "optimal" with respect to a particular $\alpha \in \mathcal{A}$, we mean that its paths are min-GC paths for trips with origin o and VOT α. All these definitions become clear below.

3.3 Step 2: Direction

Finding the descent direction is the algorithm's most complicated step. In the next few paragraphs, we decompose this step into $|\mathcal{N}|$ subproblems, each solved with a finite sequence of *conventional* min-path assignments. This sequence of assignments we dub T2-MPA.

Call the current approximate solution $\mathbf{x}^o$ and define the associated link times and costs for $e \in \mathcal{E}$ as

$$\begin{aligned} t_e^o &= t_e(x_e^o) \\ c_e^o &= u_e^o t_e'(x_e^o). \end{aligned}$$

To determine the direction of steepest descent, Step 2 seeks the minimand of the directional derivative at $\mathbf{x}^o$. That is, it computes $\mathbf{x}^{min}$ such that

$$\mathbf{x}^{min} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \nabla V(\mathbf{x}^o)\mathbf{x} \tag{33}$$

where here

$$\nabla V(\mathbf{x}^o)\mathbf{x} = \int_{\mathcal{A}} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o) x_e(\alpha) d\alpha. \tag{34}$$

Theorem 4 (T2 Min-Path Assignment). $\mathbf{x}^{min}$ can be found with a sequence of conventional min-path traffic assignments.

Proof. Since arc times and costs are fixed at t_e^o and c_e^o, the right side of (34) is separable in α and:

$$\min_{\mathbf{x}\in\mathcal{X}} \int_{\mathcal{A}} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o) x_e(\alpha) d\alpha = \int_{\mathcal{A}} \left(\min_{\mathbf{x}(\alpha)\in\mathcal{X}(\alpha)} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o) x_e(\alpha) \right) d\alpha. \tag{35}$$

That is, the minimum is the integral of the solutions of a set of independent linear programs, one for each VOT α. Decomposing the problem yet further, each of these linear programs is itself $|\mathcal{N}|$ separable linear programs, one for each origin node:

$$\min_{\mathbf{x}(\alpha)\in\mathcal{X}(\alpha)} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o) x_e(\alpha) = \sum_{o\in\mathcal{N}} \min_{\mathbf{x}(\alpha)\in\mathcal{X}(\alpha)} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o) x_{oe}(\alpha). \tag{36}$$

Each summand on the rightmost of (36) can be solved with a simple min-GC path algorithm that gives each arc e a "length" of the link's GC $\alpha t_e^o + c_e^o$, then builds a min-path tree and assigns to its paths all trips with VOT α and origin node o. Each problem is trivial. Each is, in the jargon, an "all-or-nothing" assignment of the trips $v_{od}(\alpha)$. The bad news is there are an infinite number of these problems. The good news is there are only a finite number of origin nodes and each has only a finite number of min-path trees. Therefore, trips with different VOTs must share the same tree. Collectively then, these different min-path trees partition $\mathcal{A}$ into a finite set of intervals.

Lemma 4 (VOT Partitioning). Let $\alpha^{lb} < \alpha^{ub}$ be two VOTs for which the same min-path tree $\mathcal{T}_o$ is optimal; i.e., $\mathcal{T}_o$ contains min-GC paths for all trips originating at o with a VOT of either α^{lb} or α^{ub}. Then $\mathcal{T}_o$ is a min-path tree (i.e., is optimal) for any VOT $\alpha \in (\alpha^{lb}, \alpha^{ub})$.

Proof. Let $p_j \in \mathcal{T}_o$ be the unique path in $\mathcal{T}_o$ to node j and let $\pi_j(\alpha)$ be the "length" (GC) of this path as perceived by trips with VOT α:

$$\pi_j(\alpha) = \sum_{e \in p_j} \alpha t_e^o + c_e^o$$

Note that $\alpha t_e^o + c_e^o$ and therefore $\pi_j()$ are linear in α. It is well known [1] that $\mathcal{T}_o$ is a min-GC tree for α^{lb} and α^{ub} iff:

$$\pi_{j_e}(\alpha^{lb}) - \pi_{i_e}(\alpha^{lb}) \leq \alpha^{lb} t_e^o + c_e^o \tag{37}$$

$$\pi_{j_e}(\alpha^{ub}) - \pi_{i_e}(\alpha^{ub}) \leq \alpha^{ub} t_e^o + c_e^o. \tag{38}$$

Because $\alpha^{lb} < \alpha < \alpha^{ub}$, we can describe any such α as

$$\alpha = (1 - \lambda)\alpha^{lb} + \lambda\alpha^{ub}$$

for some $\lambda \in (0, 1)$. Multiplying both sides of (37) and (38) by $(1 - \lambda)$ and λ, respectively, then adding the results and rearranging yields

$$\pi_{j_e}\left((1-\lambda)\alpha^{lb} + \lambda\alpha^{ub}\right) - \pi_{i_e}\left((1-\lambda)\alpha^{lb} + \lambda\alpha^{ub}\right) \leq \left((1-\lambda)\alpha^{lb} + \lambda\alpha^{ub}\right) t_e^o + c_e^o$$

or $\pi_{j_e}(\alpha) - \pi_{i_e}(\alpha) \leq \alpha t_e^o + c_e^o$, which proves that if $\mathcal{T}_o$ is a min-path tree for α^{lb} and α^{ub}, it is also one for any α between these VOTs.

Lemma 5 (VOT Greatest Lower Bound). Let $\mathcal{T}_o$ be a tree that is a min-path tree for trips with VOT α^{ub} and let α^{lb} be the largest VOT less than α^{ub}, for which tree $\mathcal{T}_l$ is *not* a min-path tree is:

$$\alpha^{lb} \leftarrow \max\{\alpha \in [0, \alpha^{ub}) : e \notin \mathcal{T}_o \wedge \alpha t_e^o + c_e^o < \pi_{j_e}(\alpha) - \pi_{i_e}(\alpha)\}.$$

Proof. For tree $\mathcal{T}_o$ *not* to be optimal for α^{lb} means that for some $e \notin \mathcal{T}_o$,

$$\pi_{j_e}(\alpha^{lb}) - \pi_{i_e}(\alpha^{lb}) > \alpha^{lb} t_e^o + c_e^o. \tag{39}$$

Therefore, from Lemma 5, $\mathcal{T}_o$ is optimal for any VOT from α^{ub} down to the largest α^{lb} for which (39) is false. (Note that (39) is false for all in-tree arcs.)

Traffic Assignment. If a tree $\mathcal{T}_o$ is optimal for all VOTs in the interval $(\alpha^{lb}, \alpha^{ub}]$, then the contributions from trips on paths in $\mathcal{T}_o$ to the variables x_{oe} and u_{oe} show in the right side of the following two expressions:

$$x_e = \sum_{o \in \mathcal{N}} x_{oe} = \sum_{o \in \mathcal{N}} \sum_{e \in p_d} v_{od} \int_{\alpha^{lb}}^{\alpha^{ub}} f_{od}(\alpha) d\alpha \tag{40}$$

$$u_e = \sum_{o \in \mathcal{N}} u_{oe} = \sum_{o \in \mathcal{N}} \sum_{e \in p_d} v_{od} \int_{\alpha^{lb}}^{\alpha^{ub}} \alpha f_{od}(\alpha) d\alpha. \tag{41}$$

This contribution suggests a min-path assignments of two "trip" variables: the rightmost summands of (40) and (41). Doing these assignments over the (finite number of) trees for each origin computes the state variables x_e and u_e—without computing individual $x_{oe}(\alpha)$'s. This proves the theorem.

The theorem's implication is that an otherwise daunting problem with a nominally infinite number of link variables is solved by maintaining only two link "volume" arrays, (x_e) and (u_e)—only one more than for conventional traffic assignment.

3.3.1 Algorithm T2-MPA

Algorithm T2-MPA computes (x_{oe}) and (u_{oe}) for a given origin node o by generating a sequence of min-path trees, all rooted at the same node o. It builds and loads these trees sequentially, from the largest VOT down to the smallest, in finite steps of VOT.

T2-MPA (Figure 11) finds each individual tree $\mathcal{T}_o$ with a standard min-path routine, using arc lengths equal to the link GCs with respect to $\mathcal{T}_o$'s maximum VOT α^{ub}. It figures out $\mathcal{T}_o$'s lower VOT bound α^{lb} by scanning the out-of-tree links for the arc that would enter the tree with the largest VOT smaller than α^{ub}. Consequently, $\mathcal{T}_o$ contains a min-GC path to each destination node $d \in \mathcal{N}$ for its specific half-open VOT interval $(\alpha^{lb}, \alpha^{ub}]$.

T2-MPA loads these min-paths in $\mathcal{T}_o$ with trips originating at o whose VOT is in that interval. When all trips in the interval are loaded, the VOT's lower bound becomes the upper bound, and the algorithm iterates on to the next tree. Because the VOT intervals are exhaustive and mutually exclusive over the entire effective range of VOT, T-MPA's path loading properly accounts for all trips from origin node o, that is, (13) is satisfied. Since neither x_{oe}^{min} nor u_{oe}^{min} explicitly appears in the objective function, the algorithm does not save their values. Instead, it accumulates only x_e^{min} and u_e^{min}.

Algorithm T2-MPA $(o, (x_e), (u_e))$

1. **Initialization.** Set first tree's upper VOT bound to largest VOT for which trips originate at node o:

$$\alpha^{ub} \leftarrow \alpha_o^{max}.$$

2. **Min-Path Tree.** Build min-path tree $\mathcal{T}_o$ for trips with VOT α^{ub} by setting arc "lengths" g_e^o for $e \in \mathcal{E}$ to

$$g_e^o \leftarrow \alpha^{ub} t_e^o + c_e^o$$

and executing any conventional algorithm that builds a min-path tree rooted at node o, yielding $\pi_j()$ for all $j \in \mathcal{N}$.

3. **Lower Bound of VOT.** Find largest VOT smaller than α^{ub} for which $\mathcal{T}_o$ is *not* a min-GC path tree:

$$\alpha^{lb} \leftarrow \max\{\alpha \in [\alpha_o^{min}, \alpha^{ub}) : e \notin \mathcal{T}_o \wedge \alpha t_e^o + c_e^o < \pi_{j_e}(\alpha) - \pi_{i_e}(\alpha)\}.$$

4. **Path Loading.** Add trips and moments with VOT in the interval $(\alpha^{lb}, \alpha^{ub}]$ to the volume(s) of each link e in their path $p_d \in \mathcal{T}_o$:

$$\begin{aligned} x_e &\leftarrow x_e + \sum_{e \in p_d} v_{od} \int_{\alpha^{lb}}^{\alpha^{ub}} f_{od}(\alpha) d\alpha \\ u_e &\leftarrow u_e + \sum_{e \in p_d} v_{od} \int_{\alpha^{lb}}^{\alpha^{ub}} \alpha f_{od}(\alpha) d\alpha. \end{aligned}$$

5. **Termination.** If $\alpha^{lb} \leq \alpha_o^{min}$ quit with solution (x_e) and (u_e); otherwise continue to Step 6.

6. **Reduce Upper Bound.** Reduce $\mathcal{T}_o$'s upper bound on VOT:

$$\alpha^{ub} \leftarrow \alpha^{lb}$$

and return to Step 2 to build next tree.

Figure 11: Algorithm T2-MPA: Load Trips From Node o

Executing T2-MPA over all origin nodes completes the traffic assignment and T2-ETA's Step 2: direction. The values of the link state variables (x_e) and (u_e) provide the steepest descent direction for use in its Step 4: combination.

3.4 Step 4: Combination

With only x_e and u_e appearing in the objective function and having x_e^{min} and u_e^{min} from the Direction step, the Combination step becomes easy. Let

$$\Delta x_e(\alpha) = x_e^{min}(\alpha) - x_e^o(\alpha).$$

That is,

$$\begin{aligned}
\Delta x_e &= \int_{\mathcal{A}} x_e^{min}(\alpha)d\alpha - \int_{\mathcal{A}} x_e^o(\alpha)d\alpha = \int_{\mathcal{A}} \Delta x_e(\alpha)d\alpha \\
\Delta u_e &= \int_{\mathcal{A}} \alpha x_e^{min}(\alpha)d\alpha - \int_{\mathcal{A}} \alpha x_e^o(\alpha)d\alpha = \int_{\mathcal{A}} \alpha\Delta x_e(\alpha)d\alpha.
\end{aligned}$$

Lemma 6 (Convex Combinations). Let $\mathbf{x}^o$ and $\mathbf{x}^{min} \in \mathcal{X}$ be two feasible traffic assignments and let $\Delta\mathbf{x} = \mathbf{x}^{min} - \mathbf{x}^o$. Then

i. their convex combination is also feasible, i.e., for $\lambda \in [0,1]$,

$$\mathbf{x}^{cc} = \mathbf{x}^o + \lambda\Delta\mathbf{x} \in \mathcal{X}$$

ii. the convex combinations of their state variables x_e and u_e are the state variables of their convex combination:

$$\begin{aligned}
x_e^{cc} &= \int_{\mathcal{A}} \sum_{o\in\mathcal{N}} x_{oe}^{cc}(\alpha)d\alpha \\
&= \int_{\mathcal{A}} \sum_{o\in\mathcal{N}} (1-\lambda)x_{oe}^o(\alpha) + \lambda x_{oe}^{min}(\alpha)d\alpha \\
&= x_e^o + \lambda\Delta x_e \\
u_e^{cc} &= \int_{\mathcal{A}} \alpha \sum_{o\in\mathcal{N}} x_{oe}^{cc}(\alpha)d\alpha \\
&= \int_{\mathcal{A}} \alpha \sum_{o\in\mathcal{N}} (1-\lambda)x_{oe}^o(\alpha) + \lambda x_{oe}^{min}(\alpha)d\alpha \\
&= u_e^o + \lambda\Delta u_e.
\end{aligned}$$

Proof. Part (i) is true because the constraint set (13) is convex. Part (ii) follows because x_e and u_e are linear functions in $x_{oe}(\alpha)$.

The Combination step requires minimizing a convex function of a single variable λ:

$$\begin{aligned}
\lambda^{opt} &= \arg\min_{\lambda\in[0,1]} V(\mathbf{x}^o + \lambda\Delta\mathbf{x}) \\
&= \arg\min_{\lambda\in[0,1]} \sum_{e\in\mathcal{E}} (u_e^o + \lambda\Delta u_e)t_e(x_e^o + \lambda\Delta x_e).
\end{aligned}$$

This is easy prey for several common algorithms, e.g., Golden Section [2].

3.5 Step 1: Initialization

To launch FW we need starting values for the state variables (x_e^o) and (u_e^o). The simplest way to obtain initial feasible values for these variables is to set link costs and times at $t_e^o = t_e(0)$ and $c_e^o = 0$ and iterate Algorithm T2-MPA over all origin nodes.

3.6 Step 3: Termination

A common test for terminating FW traffic assignment algorithms is to compare the "relative gap" with a given positive threshold $\epsilon << 1$:

$$\frac{|\nabla V(\mathbf{x}^o)\Delta \mathbf{x}|}{\nabla V(\mathbf{x}^o)\mathbf{x}^{min}} < \epsilon. \tag{42}$$

To apply (42) conveniently here, we need to express it terms of the state variables x_e and u_e:

Lemma 7 (Compacted Directional Derivative).

$$\nabla V(\mathbf{x}^o)\Delta \mathbf{x} = \sum_{e\in\mathcal{E}}(t_e^o \Delta u_e + c_e^o \Delta x_e).$$

Proof.

$$\begin{aligned}
\nabla V(\mathbf{x}^o)\Delta \mathbf{x} &= \int_{\mathcal{A}} \sum_{e\in\mathcal{E}} (\alpha t_e^o + c_e^o)\, \Delta x_e(\alpha) d\alpha \ \ \{\text{from Lemma 2}\} \\
&= \sum_{e\in\mathcal{E}} \left(t_e^o \int_{\mathcal{A}} \alpha \Delta x_e(\alpha) d\alpha + c_e^o \int_{\mathcal{A}} \Delta x_e(\alpha) d\alpha \right) \\
&= \sum_{e\in\mathcal{E}} (t_e^o \Delta u_e + c_e^o \Delta x_e).
\end{aligned}$$

A practical termination test is therefore

$$\frac{|\sum_{e\in\mathcal{E}} (c_e^o \Delta x_e + t_e^o \Delta u_e)|}{\sum_{e\in\mathcal{E}} (c_e^o x_e^{min} + t_e^o u_e^{min})} < \epsilon.$$

Having clarified and justified all four steps in Figure 10, we terminate our derivation of this rendition of a Frank-Wolfe algorithm for finding optimal tolls on a network. The convexity of (1), assures that the algorithm will always converge on an optimal solution [2].

4 Conclusions

This paper obtained the optimal toll for each link in a network by finding the flow that minimizes the expected total cost of time

$$V(\mathbf{x}) = \sum_e \overline{\alpha}_e x_e t_e(x_e)$$

for a stochastically diverse population of trips. The solution reflects a state where each trip is user-optimal with respect to its particular value-of-time, while the resulting total effect is system-optimal. The value of two state variables derived from the decision variable $\mathbf{x}_e^{opt}$ yields the optimal toll for link e:

$$c_e^{opt} = \overline{\alpha}_e^{opt} t_e'(x_e^{opt}).$$

Despite the infinite dimension of the decision variable, the solution algorithm is efficient. It solves a convex programming problem with a finite number of conventional min-path traffic assignments and stores only two state variables per link. It can optimize flow in networks as large as those used for conventional equilibrium traffic assignment.

The model and the algorithm are robust. For example, some arcs may have a preset toll—including zero, and the model will honor these fixed tolls while finding optimal tolls over the remaining links. This feature permits rigorous integration of existing or invariant link costs, such as prespecified toll roads and parking fees. In addition, tolls can be rounded, say to whole dimes. However, in both these latter cases, conditions and claims for convergence and uniqueness could be problematic [6] and represent worthy subjects for future research.

4.1 Future Work.

Improvements leading to a dynamic version of the model and its algorithm are, I think, the most crucial areas for future work. Anyhow, future work should entail research, development and demonstrations that exploit T2's attributes while reducing deficiencies.

Dynamic Assignment. The model's most notable weakness is being static. That is, it ignores the time domain. Congestion is a temporal phenomenon, and a static model of congestion is an oxymoron. To predict travel times accurately, a dynamic traffic assignment is essential [8, 9]. Furthermore, any practical application of congestion pricing would levy different tolls on the same link at different times of day, and only a dynamic model lets trips delay their departure and/or arrival time to avoid tolls [4].

Fortunately, the introduction of the time domain into our model is no more difficult than into the conventional traffic assignment model. The classic time-staged network approach applies as easily here. Queue backups, turning delays at intersections, shock waves, and the like are as readily modeled inside our bicriterion model. Static bicriterion traffic assignment is a pure generalization of conventional static traffic assignment: its theory generalizes to a dynamic model identically.

Link Time-Volume Functions. It is axiomatic that before we can price congestion rationally, we must first model it accurately. The better we model it, the more efficient our tolls. Congestion, besides being temporal and depending on the additional delay factors alluded to above, is not so well behaved as "convex and twice differentiable" implies. A more sophisticated road pricing model would incorporate a more realistic albeit ill-behaved depiction of congestion, such as that proposed by Bernstein and Smith [5].

Elastic Demand. Another apparent weak point is that the model's trip matrix is fixed. Again, because it is a generalization of the conventional model, this shortcoming is easily overcome. A bicriterion traffic assignment model can be cast as a "combined distribution-traffic assignment model," as Leurent shows for the case where tolls are fixed [10].

VOT Density Estimation. Having important applicability beyond congestion pricing, the estimating and forecasting of value-of-time probability distributions are superb research topics. Presenting the T2 model in seminars, I am always asked where these distributions come from. My answer is always the same [6]: estimate them by observing path choice probability *vis a vis* path cost and time and fit curves, using parametric methods or the more modern approaches proposed by Silverman [15]. I would love to see someone follow up on this glib response.

4.2 Comment on Performance.

The direction step is clearly the performance bottleneck. It builds an unspecified, potentially large number of trees for each origin zone. Compared to the conventional traffic assignment, which builds only a single tree per origin, it would seem to portend a rapacious appetite for compute cycles.

Actually, the direction step does not perform as grotesquely as you might imagine. It indeed builds multitudinous trees, but fortunately each successive tree is very similar to its predecessor. One tree can change into the next in tiny fraction of the time needed to build a tree from scratch—being akin to swapping basis arcs in the network simplex algorithm.

An upcoming paper will describe this implementation and presenting running-time statistics for several large networks. It will report some impressive tree-building speeds. On a network of 4000 nodes and 16,000 heavily congested links, loaded with a 100-percent dense 1000-by-1000 trip matrix and a continuous VOT PDF with domain from zero to $1.00 per minute; my T2-ETA code running on a 33-MHz 486 builds and loads over 150 trees (600,000 min-GC paths) per *second.*

4.3 Acknowledgements

The U.S. Department of Transportation and the Environmental Protection Agency sponsored this work as part of their Transportation Model Improvement Program (TMIP). I am very grateful for the support of Mike Jacobs of the Volpe National Transportation Systems Center and for the patience and sympathy of my other colleagues Mark Bucciarelli, Doug Lee, Peter Mengert and Don Pickrell. I also owe thanks to David Bernstein of Princeton University for his several astute observations and helpful suggestions, which I employed without explicit credit. Finally, I want to thank Jim Orlin of the Massachusetts Institute of Technology. Without his parametric tree building algorithm [13], upon which Algorithm T2-MPA is based, there would be no blazing solution speeds to brag about.

4.4 Epilogue

In celebration of their success, the Oracle and Chairperson of the Marysville Board of Supervisors partied one night at the Oracle's home. After her sixth Rolling Rock, the Oracle began to display remarkable candor for someone in her profession:

"Look at this crystal ball," she belched. "I bought it at a yard sale. It's nothing but an upside down fish bowl." Then she lifted the bowl. To the Chairperson's amazement, it had concealed a notebook PC.

"While I stared into my crystal ball," she grinned, "I was running software that figured out the traffic Ed's tolls would produce."

"So that's how you did it," said the Chairperson. Then seizing the opportunity, she continued: "That explains how you predicted traffic for a given toll, but who gave you those tolls you recommended to me?"

"The computer gave them to me," slurred the Oracle. "Instead of me telling it tolls, I had it figure out the best ones."

"The *best* tolls?" the Chairperson asked.

"Yes," continued the Oracle. "The best tolls would cause trips to minimize the *total* perceived travel-time cost, while each trip minimized its own generalized cost: self interest working in society's interest!" Then unfolding an old drawing of hers (Figure 12) , she mumbled: "These figures here represent an approximate equilibrium, but the relative gap is less that 10^{-3}. If you divide the u_e by the x_e, you get the average value of time of trips using link e, and . . ."

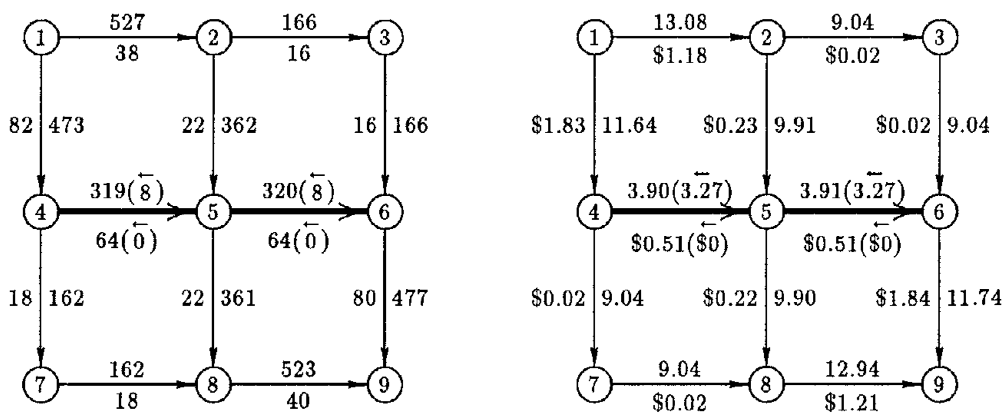


Figure 12(a): Volumes (x_e) and Moments (u_e) Figure 12(b): Times (t_e) and Tolls (c_e)

Figure 12: Optimal Tolls: State Variables at Equilibrium ($\epsilon = 10^{-3}$)

Seeking a return to familiar ground, the Chairperson interrupted: "So the tolls the computer came up with are the ones you recommended for Marysville."

"Yes and no," answered the Oracle. Running her hand across the drawing, she explained: "Here are the tolls my PC came up with (Figure 12(b)). The tolls I gave you are my tidier versions of these. I got rid of the tiny ones, rounded others to a nickel, and created some symmetry."

"But how did you know your tidy tolls would not get messy results?" asked the Chairperson.

"I checked them out with the computer," answered the Oracle. "I ran it again using my tolls. The results I got were great."

"They sure were," agreed the Chairperson. Smiling widely, the two ladies clinked their glasses.

References

[1] R. Ahuja, T. Magnanti and J. Orlin, *Network Flows,* Prentice Hall (1993).

[2] M. Bazaraa, H. Sherali and C. Shetty, *Nonlinear Programming: Theory and Algorithms,* John Wiley & Sons (1979).

[3] M. Beckmann, C. McGuire and C. Winston, *Studies in the Economics of Transportation,* Yale University Press (1956).

[4] D. Bernstein, Personal Correspondence, Massachusetts Instute of Technology (1993).

[5] D. Bernstein and T. Smith, "Equilibria for Networks with Lower Semicontinuous Costs: With an Application to Congestion Pricing." *Transportation Science* **28,** pp 221–235 (1994).

[6] R. Dial, " Multicriterion Equilibrium Traffic Assignment, Basic Theory and Elementary Algorithms, Part I, T2: The Bicriteria Model." *Transportation Science*, In press.

[7] M. Frank and P. Wolfe "An Algorithm for Quadratic Programming." *Naval Research Logistics Quarterly*, **3** pp 95–110 (1956).

[8] T. Friesz, D. Bernstein, T Smith, L. Tobin L., and B. Wie, "A Variational Inequality Formulation of the Dynamic Network User Equilibrium Problem." *Operations Research*, **41** No. 1 pp 179–191 (1993).

[9] B. Janson, "Dynamic Traffic Assignment for Urban Road Networks." *Transportation Research*, **25B**, pp 143–161. University of California Press, Berkely, CA.

[10] F. Leurent, "Cost Versus Time Equilibrium over a Network." Presented at the 1994 Annual Conference of the Transportation Research board. Washington D. C (1994).

[11] H. Moring, *Transportation Economics,* Ballinger Publishing Company (1976).

[12] S. Nguyen, "An Algorithm for the Traffic Assignment Problem." *Transportation Science,* **8**, No. 3 pp 205–218 (1974) .

[13] J. Orlin, "2-Criteria and 3-Criteria Shortest Path Problem: a Proposal," Personal Correspondence, Massachusetts Institute of Technology (1993).

[14] G. Roth, *Paying for Roads: the Economics of Traffic Congestion,* Penguin Books (1967).

[15] B. Silverman, *Density Estimation for Statistics and Data Analysis,* Chapman and Hall (1986).

[16] K. Small, *Urban Transport Economics,* Harwood Acadamic Publishers, (1992).

[17] J. Wardrop, "Some Theoretical Aspects of Road Traffic Research," *Proc. Inst. of Civil Engineers, Part 2,* pp 325–378 (1952).

NOTICE

This report is being distributed through the U.S. Department of Transportation's Technology Sharing Program.

DOT-T-95-20

DOT-T-95-20

Technology Sharing

A Program of the U.S. Department of Transportation