

---

# Multicriterion Equilibrium Traffic Assignment: Basic Theory and Elementary Algorithms

## Part I T2: The Bicriterion Model

*April, 1995*

# TMIIP

**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



# **MULTICRITERION EQUILIBRIUM TRAFFIC ASSIGNMENT: BASIC THEORY AND ELEMENTARY ALGORITHMS**

## **PART I T2: THE BICRITERION MODEL**

---

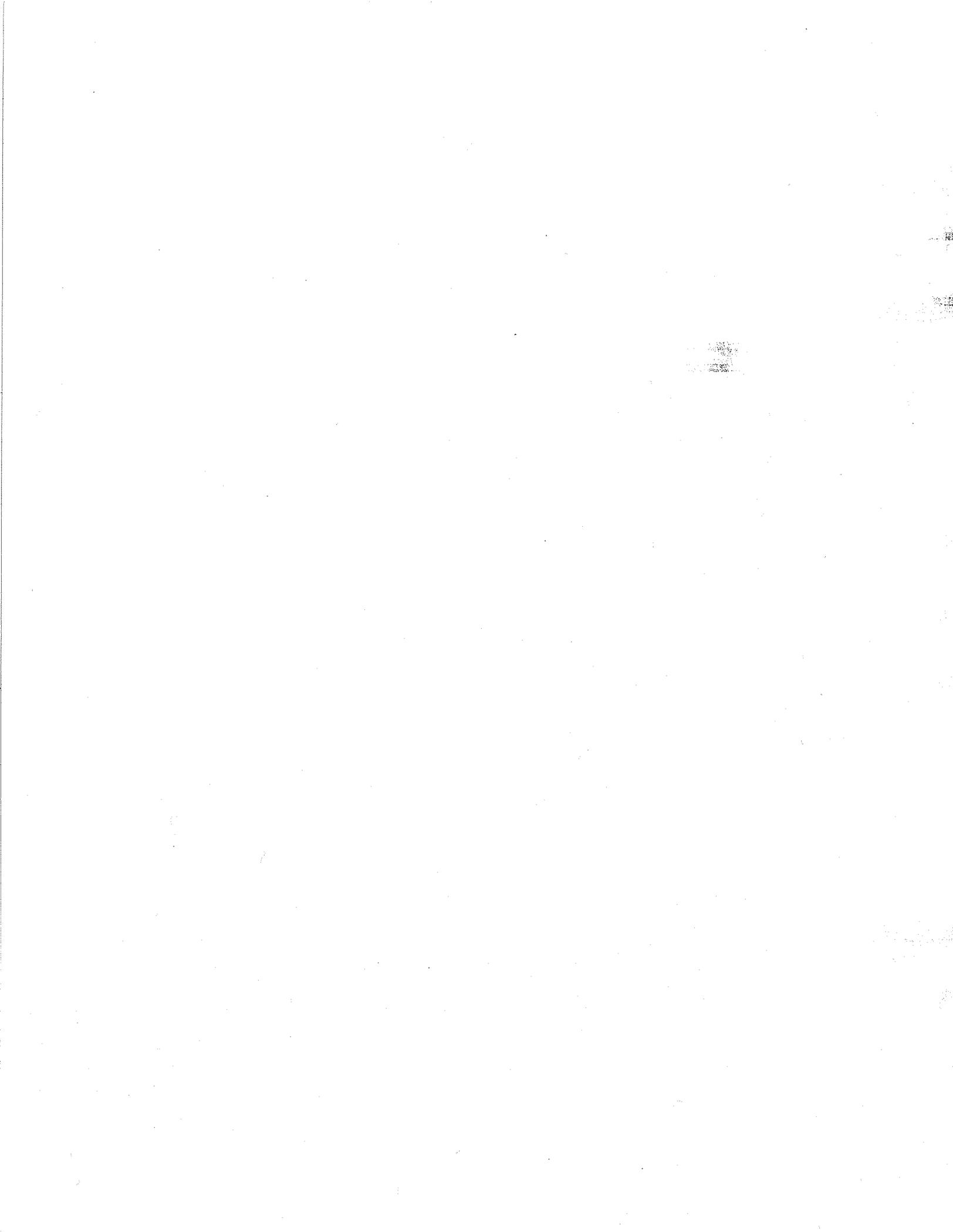
**Accepted for publication in  
Transportation Science  
April 1995**

### **Prepared by**

Robert B. Dial  
Volpe National Transportation Systems Center

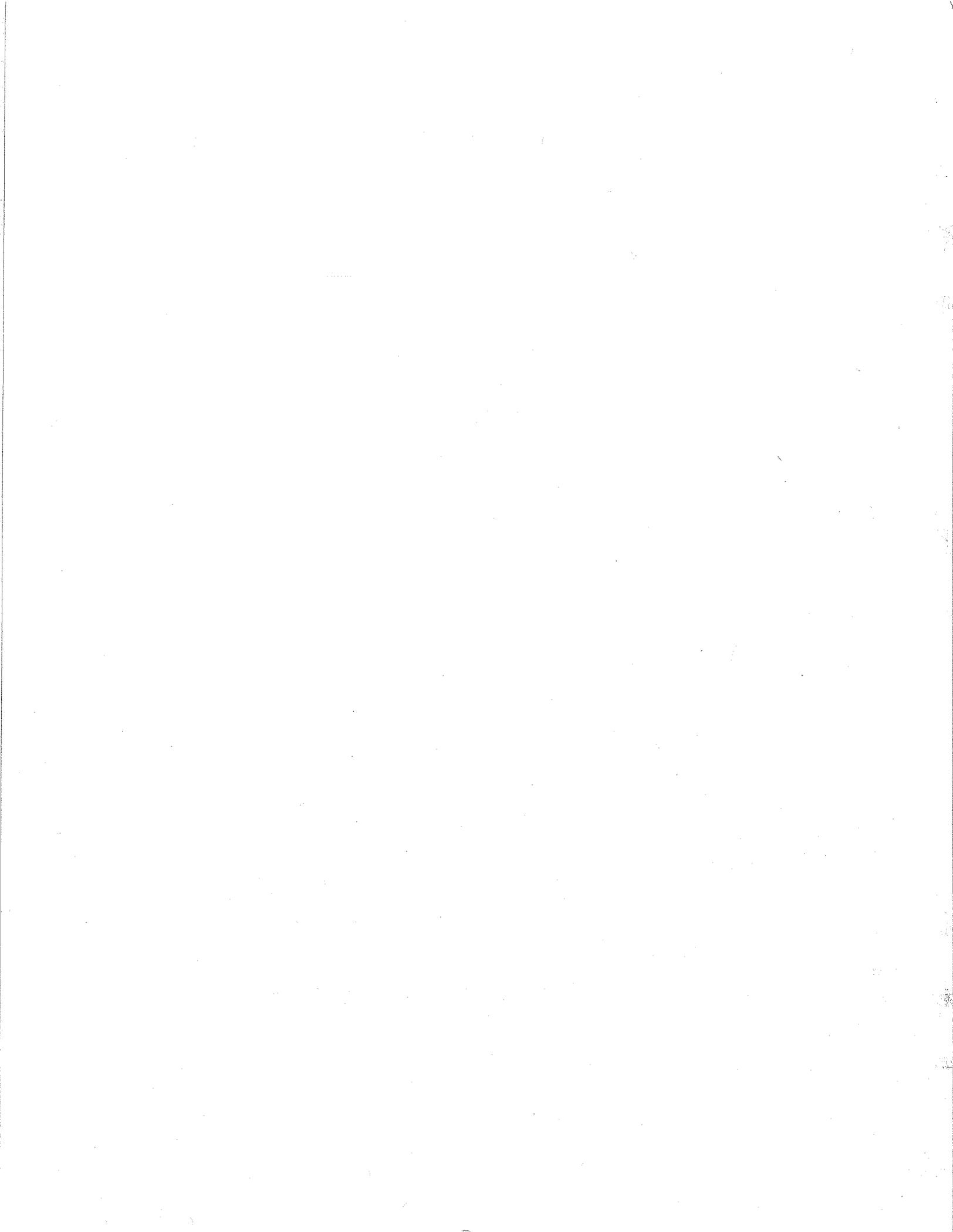
### **Prepared for**

U.S. Department of Transportation  
Federal Transit Administration  
Federal Highway Administration  
Office of the Secretary  
U.S. Environmental Protection Agency



## Abstract

T2 is a bicriterion equilibrium traffic assignment model that accurately forecasts path choices and consequent total arc flows for a stochastically diverse set of trips. Developed around a linear generalized cost model, T2 generalizes classical traffic assignment by relaxing the value-of-time parameter from a constant to a random variable with an arbitrary probability distribution. For the case where arc time and/or cost are flow-dependent, this paper formulates conditions and algorithms for stochastic bicriterion user-optimal equilibrium arc flows, which reflect every trip's exclusive use of a path that minimizes its particular perception of generalized cost.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Related Work . . . . .	3
1.2	Overview of the T2 Model . . . . .	3
1.3	Path Choice Probability . . . . .	5
1.4	Path-Finding Algorithm . . . . .	6
1.5	Traffic Equilibrium . . . . .	8
1.6	Remarks . . . . .	11
1.6.1	The VOT PDF . . . . .	11
1.6.2	Dynamic Traffic Assignment . . . . .	12
1.6.3	Organization of this Paper . . . . .	12
<b>2</b>	<b>T2 Min-Path Assignment</b>	<b>13</b>
2.1	T2 Ground Set X . . . . .	13
2.2	Total Generalized Trip Cost . . . . .	14
2.3	T2 Min-Path Traffic Assignment . . . . .	14
2.4	T2 Min-Path Assignment Algorithm . . . . .	15
2.4.1	Algorithm T2-MPA (Appendix A, Figure A2) . . . . .	15
2.4.2	Algorithm LikelyPaths (Appendix A, Figure A3) . . . . .	16
<b>3</b>	<b>T2 Equilibrium Assignment</b>	<b>17</b>
3.1	T2 Adaptation of Wardrop's Principle (T2-WP) . . . . .	17
3.2	T2 Equilibrium Traffic Assignment Theorem . . . . .	17
3.3	T2 Equilibrium Traffic Assignment Algorithm . . . . .	18
3.4	Algorithm T2-ETA (Appendix A, Figure A4) . . . . .	20
3.4.1	The Calculation of $u_e$ . . . . .	21
3.4.2	Example 1: Algorithm T2-ETA . . . . .	22
3.4.3	Example 2: Sampled Iterations of T2-ETA . . . . .	26
3.5	T1.5 Equilibrium Traffic Assignment . . . . .	27
3.6	Algorithm T1.5-ETA (Appendix A, Figure A5) . . . . .	28
<b>4</b>	<b>Conclusion</b>	<b>30</b>
4.1	Summary . . . . .	30
4.2	Additional Research and Development . . . . .	30
4.3	Acknowledgements . . . . .	32
	<b>Appendix A: Formal T2 Algorithms</b>	<b>34</b>
A1.	Data Types for All Algorithms . . . . .	34
A2.	Algorithm T2-MPA . . . . .	34
A3.	Likely Paths Algorithm . . . . .	35
A4.	Algorithm T2-ETA . . . . .	36
A5.	Algorithm T1.5-ETA . . . . .	37
	<b>References</b>	<b>38</b>

## List of Figures

1	Feasible Paths and $g_p(0.43)$ . . . . .	2
2	Efficient Frontier, Likely Path Probabilities . . . . .	4
3	Probability of Using Path 4 . . . . .	7
4	Arc Attributes at Free-Flow Volumes . . . . .	9
5	Arc Time vs Arc Flow (BPR) . . . . .	9
6	Value-of-Time Probability Density: $f_{19}()$ . . . . .	9
7	A T2 Equilibrium Traffic Assignment . . . . .	10
8	Two-Arc Equilibrium Example: Input . . . . .	22
9	Ex. 1: Three Iterations of Algorithm T2-ETA . . . . .	24
10	Ex. 2: Sampled Iterations of Algorithm T2-ETA . . . . .	26

## List of Tables

1	Ten Iterations of T2-ETA (Two-Arc Example) . . . . .	25
---	--	----

# 1 Introduction

The old saw “time is money” connotes the very *raison d’être* of transportation systems. Every transportation plan evaluation includes a users’ value of time (VOT), every modern mode-choice model hosts a VOT parameter, and every congestion pricing [26] scheme wagers that some travelers take slower routes to avoid tolls while others choose toll roads to save time.

Furthermore, it is a fact that each traveler has a different VOT, depending on how much money or time he is willing or able to spend on a particular trip. Conventional transportation planning models, however, acknowledge this fact poorly. Instead of using a realistic *distribution* of the VOT, these models use an *average* VOT. As a consequence, they invariably produce large estimation errors and inaccurate forecasts.

In a recent exception to this common practice, Professor Moshe Ben Akiva of M.I.T. estimated his logit mode choice model’s parameters by assuming a distribution  $f()$  of the VOT  $\alpha \in \mathcal{A}$ . That is, letting  $\alpha$  be the VOT and  $m$  a choice option, he fit the *expected* value:

$$\mathbf{E}[\mathbf{Prob}[m]] = \int_{\mathcal{A}} \frac{e^{g(m,\alpha)}}{\sum_k e^{g(k,\alpha)}} f(\alpha) d\alpha$$

instead of the usual simpler formulation, which assumes all user have the same VOT. His reward was an amazing improvement in goodness-of-fit [5].

This paper proposes a similar remedy to the same deficiency in classical traffic assignment. It presents the basic theory and elementary algorithms of a traffic assignment model that admits VOT distributions. Called T2, the model is a bicriteria user-optimal equilibrium traffic assignment model, which generalizes classical traffic assignment, by relaxing the VOT parameter in the generalized-cost function from a constant to a random variable with an arbitrary probability density function (PDF).

A traffic assignment model that is more robust than its conventional ancestor, T2’s potential use spans a wide spectrum of current and difficult problems. These include mode-route choice, parking policy planning, and congestion pricing. T2 can model mode choice by assigning trips to paths in a multi-modal (“hyper”) network, which combines walking, riding, transit and highway links. It can selectively route auto trips to parking lots that are cheap but require a long walk and to others that are closer to work but expensive. It is useful model for determining where to place tolls and what prices to levy in order to reduce congestion.

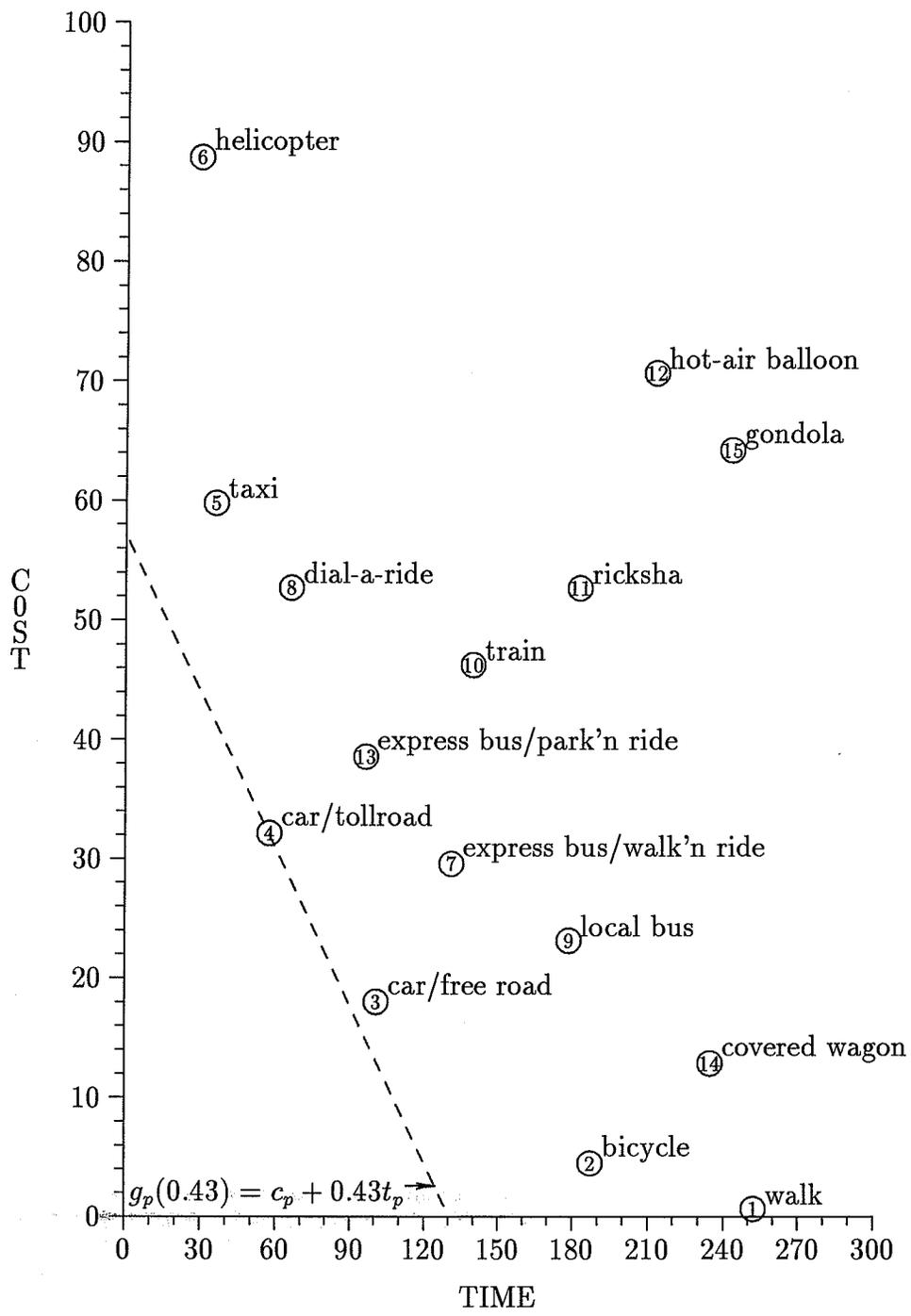


Figure 1: Feasible Paths and  $g_p(0.43)$

## 1.1 Related Work

T2's roots are in Quandt [25] and later Schneider [27]; however, this paper develops the model very differently than theirs. Its mathematical programming approach adds generality regarding the value-of-time probability density function (PDF) and formulates and solves the bicriteria user-optimal traffic equilibrium problem. It extends considerably the short note by Dial [9], which addressed only path finding. Dafermos [7] addresses T2's equilibrium problem; however, this paper makes no direct use of her results. Besides these earlier papers, this paper builds on the variational inequality work of Magnanti and Perakis [21] presented at the 1994 ORSA/TIMS conference in Boston, which inspired algorithms T2-ETA and T1.5-ETA.

The insightful work of Leurent [17] merits special comment. Leurent's formulation is at once more and less general than T2's. He deals with elastic demand, while T2 assumes a fixed total trip matrix; however, Leurent only permits one criterion, i. e., time, to be flow dependent, while T2 permits both. Moreover, his Monte Carlo solution approach [28] brings sampling errors, greatly restricts the VOT PDF and, compared to T2's algorithm, converges glacially [30].

Finally, it has been claimed [3] that a heuristic that uses a "small set of trip classes" to represent points on the PDF would accomplish T2's purpose. Frankly, such an approach would be an inefficient dead end. It is nothing more than a gross distortion of the original problem to fit the confines of existing software. To use it would risk fatal aliasing errors [17] that nullify the model's benefit. In addition, a naive application of this heuristic would save arc flow by trip class, a waste of memory since these flows are not unique.

## 1.2 Overview of the T2 Model

Combined traffic assignment and mode choice serves as a concrete example to introduce T2. Consider a trip from origin  $o$  to destination  $d$ . We wish to estimate the mode choice for this trip. Assume for the moment it is possible to enumerate all feasible paths for this trip and to know the time and cost of each. Now plot each path at a point in a graph according to its time and cost.

Figure 1 plots these time-cost points for fifteen feasible paths—ignore the dashed line for now. The horizontal axis is path time; the vertical axis is path cost. Each path label suggests its so-called "mode." Helicopter is the fastest and most expensive. Walking is the slowest and cheapest. Gondola is very expensive and very slow.

This example begs three general questions:

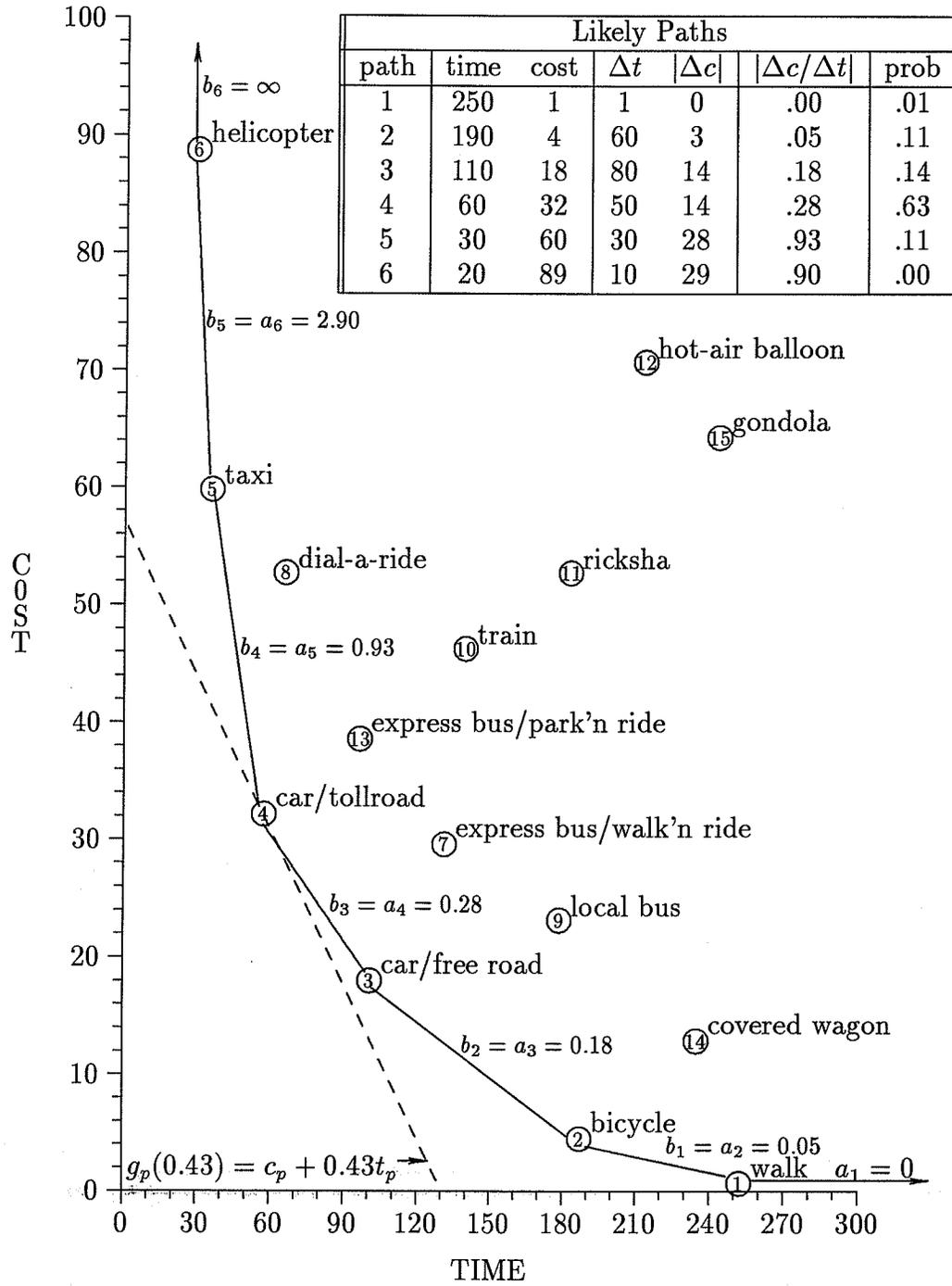


Figure 2: Efficient Frontier, Likely Path Probabilities

1. **path-choice probability:** what are the odds that a trip will choose each of these paths?
2. **path-finding algorithm:** how can a computer find these paths without enumerating all zillion feasible paths?
3. **traffic equilibrium:** how do we model the fact that path times depend on path choices, and vice versa?

We introduce these questions now and answer them later in greater detail.

### 1.3 Path Choice Probability

Answering this first question requires knowing how the trip decides among its feasible paths. Pretend for the moment that arc times and costs are *not* flow-dependent. Assume that a trip chooses a path  $p$  that minimizes its perceived generalized cost  $g_p(\alpha)$ , where:

$$\begin{aligned}
 g_p(\alpha) &= c_p + \alpha t_p; \\
 c_p &= \text{out-of-pocket ("dollar") cost of path } p; \\
 t_p &= \text{time of path } p; \\
 \alpha &\in [0, \infty) = \text{value-of-time, a random variable.}
 \end{aligned}$$

The probability of a trip choosing a certain path depends on that path's time and cost compared to all others'—together with the trip's particular VOT  $\alpha$ .

Referring again to Figure 1, the dashed line represents the objective function, which a trip with a VOT of \$0.43/min aims to minimize. The slope of this line is negative  $\alpha$ . The generalized cost  $g_p(\alpha)$  of path  $p$  is the line's intercept with the cost axis. Therefore, for a given  $\alpha$ , the best path is the first point that this line touches as it translates from left to right. Figure 1 shows path 4, "car/tollroad", is a trip's best choice if  $\alpha = 0.43$ . If the dashed line had a different slope, then path 4 might not be the best path.

Refer now to Figure 2. Whatever the slope  $\alpha$ , the only paths with any chance of selection are 1 through 6—those shown delimiting line segments. Among all feasible paths, only these six have a generalized cost (GC) that could be the minimand of *any* linear function of  $\alpha$ . The line segments that connect such path points, in decreasing sequence by path time, form the *efficient frontier (EF)*. Points where the EF changes direction are called *extreme points*. As we shall see momentarily, in the case of a continuous VOT PDF, only paths forming extreme points represent rational path choices.

Thus, for fixed arc times and costs, Figure 2 reveals how to compute the probability of using a path. For example, consider path 4, the toll road in

Figure 3(a), detailing the neighborhood of path 4 in Figure 2. Assume that the line segment connecting path 5 and path 4 has a slope of -2.8, and the line segment connecting path 4 and path 3 has slope -1.2. Then, path 4 will minimize any GC whose VOT is between 1.2 and 2.8. Hence, the probability of selecting path 4 is the probability that the random variable  $\alpha$  is between these values.

For example, if the probability density of  $\alpha$  for trips going from  $o$  to  $d$  is  $f_{od}()$ , then as Figure 3 shows, the probability of using path 4 is 0.63. In general, because the generalized cost is a linear function of  $\alpha$ , the probability of using a path off the EF is zero; while for a path  $p$  on the EF:

$$\mathbf{Prob}[p] = \int_{a_p}^{b_p} f_{od}(\alpha) d\alpha, \quad (1)$$

where  $a_p$  is the slope of the line segment connecting  $p$  to its right neighbor on the EF, and  $b_p$  is the slope of the segment connecting  $p$  to its left neighbor. Note that  $a_p = 0$  for the cheapest path, and  $b_p = \infty$  for the fastest path.

Any path that might be used we call *likely*; other paths we call *unlikely*. Technically, likely paths have a nonzero probability of use; unlikely paths have probability zero. A likely path is always on the EF but being on the EF does *not* guarantee a path is likely. For a continuous PDF, a likely path is always an extreme point on the EF, and a path not having *distinct* bounding  $\alpha$ 's is an *unlikely* path—whether on the EF or not: for  $\mathbf{Prob}[p]$  to be greater than zero,  $b_p$  must be strictly greater than  $a_p$ . The table inset in Figure 2 lists the probabilities for Figure 2's likely paths; all other, unlikely paths have probability zero.

In the case of a discrete PDF, other paths on the EF are equally likely, and entropy considerations [16] dictate that *any* path with a minimum GC merits an equal share of trips; we ignore this concern here; the “all-shortest-paths” algorithm in [8] handles this condition. Furthermore, as will be seen later, if path time is a function of arc flow, then these shares are not in general equal.

#### 1.4 Path-Finding Algorithm

By considering likely paths only, we reduce an otherwise daunting challenge to a practical chore: of all the paths connecting  $o$  to  $d$ , we need to find only those few that shape the EF. Presuming that a path's cost and time are the sums of the cost and time of the path's arcs, then the GC of a path is just the sum of the GC of its arcs. Consequently, finding these likely paths can be easy.

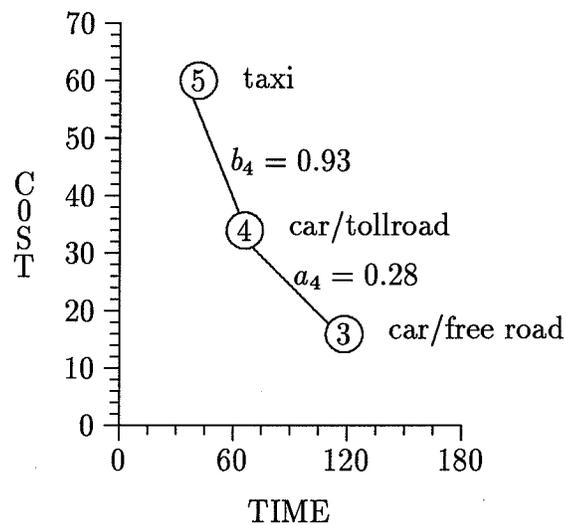


Figure 3(a): EF in Neighborhood of Path 4

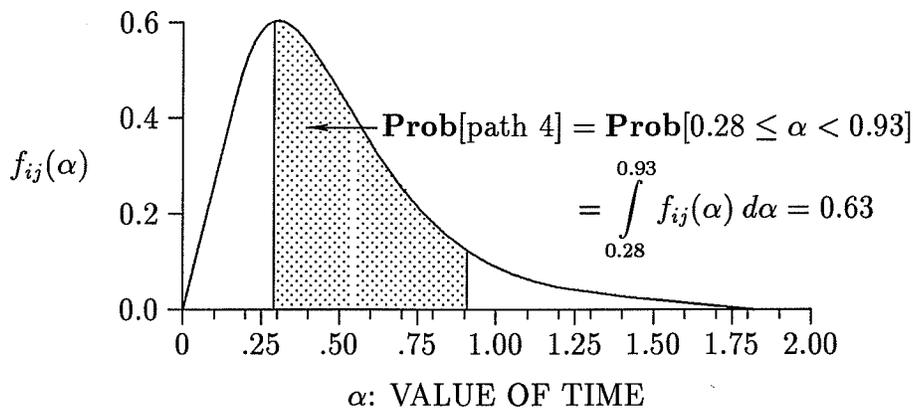


Figure 3(b): VOT PDF Share for Path 4

Figure 3: Probability of Using Path 4

Each likely path  $p$  on the EF minimizes  $g_p(\alpha)$  for some  $\alpha$ . Let the GC of an arc  $e$  be the sum of its cost plus  $\alpha$  times time:  $g_e = c_e + \alpha t_e$ . If we label each arc with its  $g_e$  for a fixed  $\alpha$  and find the minimum GC path, that min-GC path will be on the EF. Thus, to find the EF is equivalent to finding min-GC paths for an appropriate set of  $\alpha$ 's.

Remarkably, the computational complexity of an algorithm that finds *all* likely paths, is the same as one that finds a *single* min-GC path. This latter algorithm is virtually a standard min-path routine—its only complication is tie-breaking—to guarantee *extreme-point* paths on the EF. (Details for all algorithms appear in Appendix A.)

### 1.5 Traffic Equilibrium

A traffic assignment model accumulates trips for all  $o-d$  pairs on each arc comprising their connecting path. When all trips use a min-GC path, we call the result a *user-optimal* traffic assignment. When arc times are fixed, a single min-path (a.k.a., “all-or-nothing”) assignment performs this calculation by assigning each trip to its min-GC path. However, arc times may change, depending on how many trips use the arc—which in turn depends on arc times. The solution to this circular problem has the name *traffic equilibrium*.

All classical assignment models permit only a single, fixed VOT  $\alpha$ : they presume the perceived GC of a path is to all trips *identical*. In contrast, T2 assumes different trips perceive the same path as having *different* GCs by virtue of different VOTs. It represents these differences in  $\alpha$  by user-supplied probability distributions. T2's generalization of Wardrop's principle [31], states that each trip uses only paths that minimize their particular perceived GC. A simple example makes this clear.

Figures 5–7 show an example of T2 equilibrium in a nine-node grid network, a single  $o-d$  pair, and a simplistic three-spike VOT PDF. The network is in Figure 4. All arcs are two-way and of length ( $d$ ) one mile. The eight peripheral arcs are free roads; the four interior arcs are toll roads costing \$5.00. Figure 5 shows the time-flow (BPR) curves of the network's two arc types, toll arcs having higher practical capacity ( $k$ ) and speed limit ( $s$ ) than free arcs.

Figure 6 is the VOT PDF, consisting of three spikes, at \$0.10, \$1.00, and \$2.00 per minute and corresponding volumes of 800, 2400, and 800 trips. All trips go from node 1 to node 9.

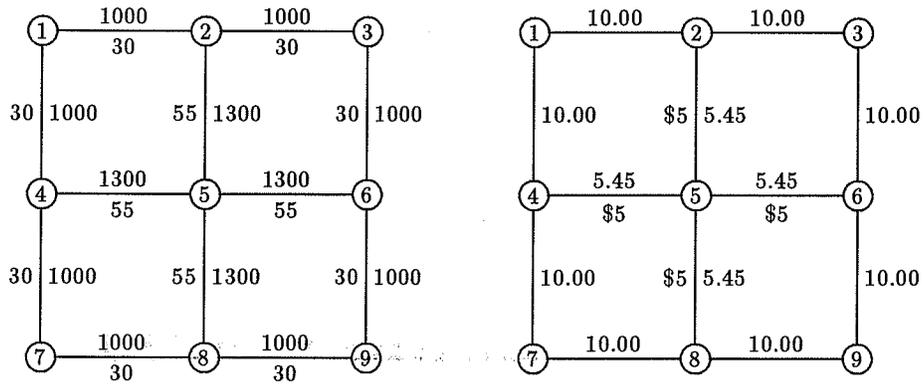


Figure 4(a): Capacity and Free-Flow Speed      Figure 4(b): Free-Flow Time and Cost

Figure 4: Arc Attributes at Free-Flow Volumes

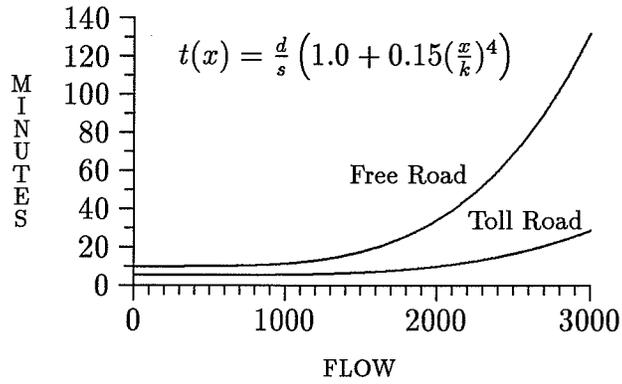


Figure 5: Arc Time vs Arc Flow (BPR)

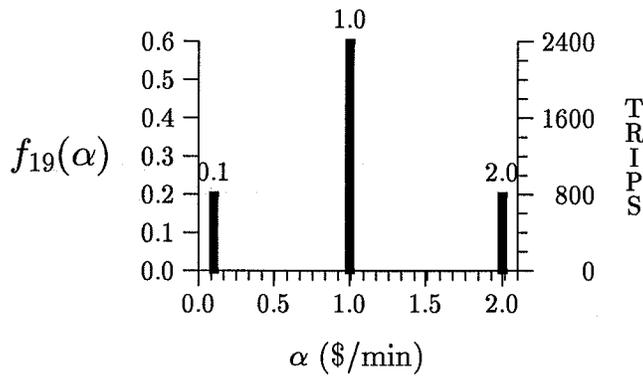
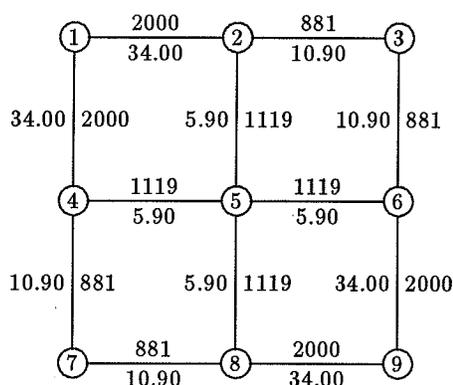


Figure 6: Value-of-Time Probability Density:  $f_{19}()$

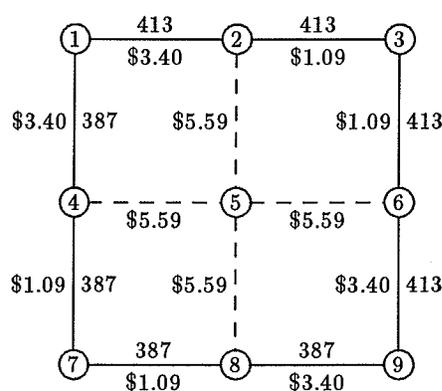
Figure 7(a) shows the associated T2 equilibrium flow. It labels each arc with an integer representing the total number of trips on the arc and a decimal number indicating the arc traversal time. For example, arc (1,2) has 2000 trips and requires 34.00 mins.

Figure 7(b) shows an *example* of the component of the total flow that represents trips with VOT of \$0.10/min. The dashed lines indicate unused arcs. The labels on the arc show trips and generalized cost. Note that all used paths from node 1 to node 9 have the same generalized cost, which is less than that of all unused paths: trips with VOT \$0.10/min are in equilibrium. As seen in Figures 7(c) and (d), the same is true for the trips with VOT \$1.00 and \$2.00.



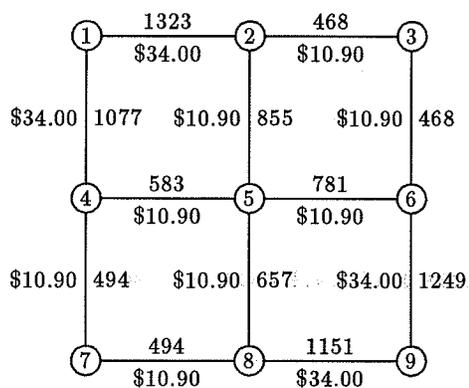
$$\sum_{\alpha} V_{1,9}^{\alpha} = 4000$$

Figure 7(a): T2-ETA Flow and Time



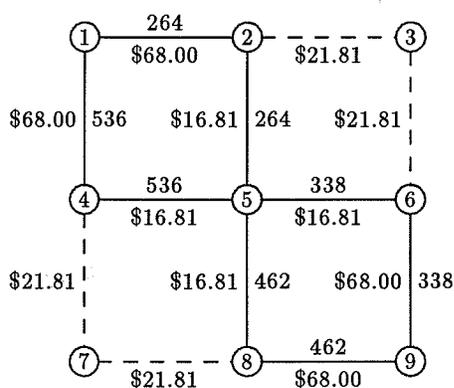
$$\alpha = \$0.10/\text{min}; V_{1,9}^{0.10} = 800$$

Figure 7(b): Flow and Generalized Cost



$$\alpha = \$1.00/\text{min}; V_{1,9}^{1.00} = 2400$$

Figure 7(c): Flow and Generalized Cost



$$\alpha = \$2.00/\text{min}; V_{1,9}^{2.00} = 800$$

Figure 7(d): Flow and Generalized Cost

Figure 7: A T2 Equilibrium Traffic Assignment

Figures 7(b)–(d) sum to those in 7(a). Note however that unlike the latter, these former flows are *not* unique: any reallocation of individual flows among their min-GC paths that summed to the same total in Figures 7(a) would also be in equilibrium for every individual VOT classes. For example, if in Figure 7(b) 413 trips shift from path 1-2-3-6-9 to path 1-4-7-8-9, while at the same time in Figure 7(c), 413 trips make the opposite shift; the result is another equilibrium flow. The total arc flows would be the same, but individual arc flows for VOTs \$.01 and \$1.00 would differ greatly.

## 1.6 Remarks

We conclude this introductory section with two remarks and a brief description of the organization of the remainder of this paper.

### 1.6.1 The VOT PDF

**Continuous vs Discrete.** T2 theory and algorithms developed below permit each directed  $o-d$  pair can have its own PDF  $f_{od}()$ , which can be continuous, discrete or any mixture of either. The integral sign used in this paper always means Riemann-Stieltjes integration [2]. Except in the simplistic case of a PDF consisting of only a few discrete points, T2’s algorithms’ complexity is unaffected.

**Estimation.** The best way to estimate a particular instance of the VOT PDF is a worthwhile research topic. As a first cut, we could estimate it by pairing likely paths with their observed probability of use, and fit a cumulative distribution function (CDF).

Fortunately, likely paths are a function of the network *only*: they are independent of the distribution of  $\alpha$ . The path finding algorithm discussed later finds every likely path  $p$  on the EF and the range  $[a_p, b_p]$  of  $\alpha$  associated with each.

An observed value of the probability of using any likely path  $p$  is an estimate of the CDF’s differential between  $a_p$  and  $b_p$ . That is, if  $F_{od}()$  is the (unknown) cumulative distribution function of  $\alpha$ , then

$$\mathbf{Prob}[p] = \mathbf{Prob}[a_p \leq \alpha < b_p] = F_{od}(b_p) - F_{od}(a_p).$$

Given a set of these observations, we may estimate  $F_{od}()$  with special statistical methods [29]. To forecast  $F_{od}()$  for other locations in space-time, its estimate can be correlated to traveler characteristics such as income.

### 1.6.2 Dynamic Traffic Assignment

The “bad news” is that the model presented here is static: T2 does *not* forecast arc-flow time profiles. Static traffic assignment models cannot accurately forecast congestion effects (i.e., travel time); therefore, they seriously compromise a planner’s forecasts. This shortcoming is particularly enfeebling when planning tolls for congestion pricing, since—besides making inaccurate link time estimates—the static model cannot forecast trips shifting their travel times to avoid or reduce toll costs [6].

The “good news” is that the temporal dimension is literally orthogonal. All the theory and algorithms presented in this paper support the development of even more accurate dynamic models. For example, all of the algorithms can be applied with minor modification to a discrete time-staged expanded network. As classical static traffic assignment provided ground work for recent work in dynamic traffic assignment [19] [13] [15], the results of this paper, by their extension into the time domain, invite the development of a dynamic T2. Although discussed no further in this paper, this extension is natural and its development urgent.

### 1.6.3 Organization of this Paper

The next two sections of this paper provide theoretical justification and algorithmic detail for the claims and procedures introduced above. Section 2 covers T2 min-path traffic assignment, which incorporates VOT as a random variable. It develops a model and algorithm for loading trips with a stochastic value of time on a network whose arc times and costs are fixed. Building on the concepts introduced in prior sections, Section 3 develops mathematical theory and algorithms for bicriteria equilibrium traffic assignment for trips with an arbitrary VOT PDF, where the network’s arc costs and/or times vary with flow. The final section briefly addresses a few proposed T2 R&D topics, then ends the paper by acknowledging people who helped produce it.

## 2 T2 Min-Path Assignment

Given a set of paths connecting all origin-destination  $o-d$  node pairs and given a trip matrix, which specifies the number of trips between each  $o-d$  pair, a traffic assignment algorithm simply accumulates on each arc the trip volume of every path using that arc. The output of a traffic assignment algorithm is this vector of accumulated arc flows, which we call a "traffic assignment." Depending on the paths the trips use, this vector has many possible values. We call the set of all (feasible) traffic assignments for a fixed trip table the *ground set*. Let

$$\begin{aligned}
 \mathcal{R}_+ &= \{\text{positive real numbers}\} \\
 \mathcal{N} &= \{\text{nodes in the network}\} \\
 \mathcal{E} &= \{\text{arcs in the network}\} \\
 v_{od} &= (\text{given}) \text{ number of trips going from } o \text{ to } d \\
 e &= \text{arc } (i_e, j_e), \text{ directed from node } i_e \text{ to node } j_e \\
 x_{oe} &= \text{flow on arc } e \text{ that originated at node } o. \\
 x_e &= \sum_{o \in \mathcal{N}} x_{oe} = \text{total flow on arc } e \\
 \bar{x} &= (x_{oe}) \in \mathcal{R}_+^{|\mathcal{N}| \times |\mathcal{E}|} = \text{a "traffic assignment."}
 \end{aligned}$$

In classical min-path traffic assignment,  $\bar{x}$  is in the ground set iff it is a non-negative solution to the following  $|\mathcal{N}|^2$  equations in  $|\mathcal{E}||\mathcal{N}|$  unknowns [1]:

$$v_{od} = \sum_{\{e \in \mathcal{E} | j_e = d\}} x_{oe} - \sum_{\{e \in \mathcal{E} | i_e = d\}} x_{oe}, \text{ for all } o, d \in \mathcal{N}.$$

### 2.1 T2 Ground Set $\mathcal{X}$

T2 is analogous to the classical model; however, it introduces the random variable  $\alpha$ , which partitions arc and o-d flows by qualifying the variables and augmenting the constraints given above as follows:<sup>1</sup>

$$\begin{aligned}
 \mathcal{A} &= \{\text{values of time (VOTs)}\} \subseteq \mathcal{R}_+ \\
 v_{od}(\alpha) &= \text{number of trips going from } o \text{ to } d \text{ with VOT } \alpha \\
 x_{oe}(\alpha) &= \text{flow on } e \text{ of trips that originated at } o \text{ with VOT } \alpha \\
 x_e(\alpha) &= \sum_{o \in \mathcal{N}} x_{oe}(\alpha) = \text{total flow on arc } e \text{ of trips with VOT } \alpha.
 \end{aligned}$$

Thus for all  $e \in \mathcal{E}$ ,  $o, d \in \mathcal{N}$ , and  $\alpha \in \mathcal{A}$ :

$$v_{od}(\alpha) = \sum_{\{e \in \mathcal{E} | j_e = d\}} x_{oe}(\alpha) - \sum_{\{e \in \mathcal{E} | i_e = d\}} x_{oe}(\alpha), \quad (2)$$

<sup>1</sup>The infinitesimal notation  $v_{od}(\alpha)d\alpha$ ,  $x_{oe}(\alpha)d\alpha$ ,  $x_e(\alpha)d\alpha$ , etc. might seem a more rigorous notation, but the visual burden exceeds its utility.

where as discussed in Section 1,

$$v_{od}(\alpha) = v_{od}f_{od}(\alpha). \quad (3)$$

Now letting

$$\begin{aligned} \bar{x}(\alpha) &= (x_{oe}(\alpha)) \in \mathcal{R}_+^{|\mathcal{N}| \times |\mathcal{E}|} \\ \bar{x} &= (\bar{x}(\alpha)) \in \mathcal{R}_+^{|\mathcal{N}| \times |\mathcal{E}| \times |\mathcal{A}|} \text{ (an "infinite vector"),} \end{aligned}$$

the ground set for T2 becomes

$$\mathcal{X} = \{\bar{x} \mid \text{equations (2) and (3) are satisfied}\}.$$

## 2.2 Total Generalized Trip Cost

In the case where arc costs and times are *fixed*, a very useful statistic describing a traffic assignment  $\bar{x} \in \mathcal{X}$  is its total *generalized trip cost*, which is the sum of products of each trip times the GC of the path it uses. For now, let these costs and times be fixed at  $\bar{c} = (c_e)$  and  $\bar{t} = (t_e)$ . For any traffic assignment  $\bar{x} \in \mathcal{X}$ , we define

$$G(\bar{x}|\bar{c}, \bar{t}) = \int_{\mathcal{A}} \sum_{e \in \mathcal{E}} (c_e + \alpha t_e) x_e(\alpha) d\alpha, \quad (4)$$

which is the *total generalized trip cost* of the  $\bar{x}$ .

## 2.3 T2 Min-Path Traffic Assignment

Clearly, if arc costs and times are *fixed* then a T2 min-path traffic assignment is a flow that reflects each trip taking its particular min-GC path, which would minimize the total generalized trip cost (4):

**Lemma 2.1** Flow  $\bar{y} \in \mathcal{X}$  is a T2 min-path traffic assignment iff

$$\bar{y} \in \arg \min_{\bar{x} \in \mathcal{X}} \int_{\mathcal{A}} \sum_e (c_e + \alpha t_e) x_e(\alpha) d\alpha. \quad (5)$$

**Proof.** ( $\Rightarrow$ ): Because  $c_e$  and  $t_e$  are fixed, the right-hand side of (4) is separable in  $\alpha$ . A T2 min-path traffic assignment certainly results in minimizing each integrand. Hence the integral is minimized. ( $\Leftarrow$ ): Proved by contradiction: if the integral is minimized, so is each integrand; hence, each must be a min-path assignment, which contradicts the hypothesis.

## 2.4 T2 Min-Path Assignment Algorithm

Referring to Lemma 1 and recalling from Section 2.1 the definitions

$$\begin{aligned}\bar{y} &= (\bar{y}(\alpha)) = ((y_{oe}(\alpha))) \\ y_e &= \int_{\mathcal{A}} \sum_{o \in \mathcal{N}} y_{oe}(\alpha) d\alpha = \int_{\mathcal{A}} y_e(\alpha) d\alpha,\end{aligned}$$

we now present a simple algorithm for finding  $(y_e)$ . That is, the algorithm produces only *total* arc flows on each arc—not flows for every VOT. The following two subsections are informal descriptions, which summarize the formal definitions in Figures A1 and A2 of Appendix A. (Ignore for now all references to variables  $u$  and  $u^o$  and the function  $H(\cdot)$ . They concern T2 equilibrium traffic assignment, and Section 3 explains them.)

### 2.4.1 Algorithm T2-MPA (Appendix A, Figure A2)

Compared to the classical model, a min-path assignment algorithm for T2 is more involved. Each  $o$ - $d$  pair may have several distinct min-GC paths depending on the value of  $\alpha$ . Each path must receive its fair share of trips, and the sum of the flow on each path must be the given  $v_{od}$ . Algorithm T2-MPA executed for each  $o$ - $d$  pair yields a T2 min-path traffic assignment. It uses algorithm LikelyPaths to find the paths to load.

To load the network with  $v_{od}$  trips going from  $o$  to  $d$  and having a VOT density  $f_{od}(\cdot)$ , do the following:

1. [likely paths] Using Algorithm LikelyPaths, find all likely paths from  $o$  to  $d$ .
2. [trip assignment] For each likely path  $p$ :
  - 2.1 [ $\alpha$ -ranges] associate with each likely path  $p$  its VOT range:  $a_p \leq \alpha < b_p$  for which  $p$  minimizes  $g_p(\alpha)$ ;
  - 2.2 [path probability] compute the probability of each  $p$ :

$$\mathbf{Prob}[p] = \mathbf{Prob}[a_p \leq \alpha < b_p] = \int_{a_p}^{b_p} f_{od}(\alpha) d\alpha;$$

- 2.2 [path share] multiply  $v_{od}$  by  $\mathbf{Prob}[p]$  to obtain  $p$ 's share of trips,  $v_p$ ;
- 2.4 [arc load] add  $v_p$  to the flow  $x_e$  of each arc  $e$  on  $p$ .

### 2.4.2 Algorithm LikelyPaths (Appendix A, Figure A3)

The following procedure finds all likely paths and determines the exclusive range of  $\alpha$  for which each path is optimal.

1. [**cheapest path**] Let  $\alpha^\circ = 0$  and solve this min-GC path problem; this finds the cheapest path—path 1 in Figure 2.
2. [**fastest path**] Let  $\alpha^\circ = \infty$  and solve this min-GC path problem; this finds the fastest path—path 6 in Figure 2.
3. [**between paths**] Let  $\alpha^\circ$  be the slope of the line connecting paths 1 and 6, and solve this min-GC path problem; this finds path 4. Now consider the two line segments connecting path 1 with 4, and path 4 with 6; their slopes find paths 5 and 3. Continue recursively until unable to find any more likely paths.

**Caveat.** Algorithm T2-MPA is *elementary*, intending only to demonstrate T2's solvability. For example, subroutine LikelyPaths() is a reasonably efficient *path* algorithm: its complexity is comparable to a single min-path algorithm. However, its enumeration of min paths renders Algorithm T2-MPA impractical for solving real-world problems—popular claims of limitless free computer time notwithstanding. A practical T2-MPA algorithm would not enumerate paths. Such an algorithm is the subject of a future paper.

### 3 T2 Equilibrium Assignment

Up to now, our technical discussion has addressed only the simple case, where the arc costs and times are fixed—i.e., do not depend on the arc flow. This section moves on to flow-dependent arc costs and times. Accordingly, we augment the notation  $c_e$  and  $t_e$  to allow for functions as well as scalars:

$$\begin{aligned} c_e(x_e) &= \text{cost to use arc } e \text{ when arc's total flow is } x_e \\ t_e(x_e) &= \text{time to traverse arc } e \text{ when arc's total flow is } x_e. \end{aligned}$$

#### 3.1 T2 Adaptation of Wardrop's Principle (T2-WP)

Wardrop's principle [31] extends readily to the stochastic- $\alpha$  case:

In a bicriterion equilibrium traffic assignment, where the VOT is a random variable  $\alpha$ , no trip (with its particular  $\alpha$ ) has another path with a smaller GC, i.e.,  $g_p(\alpha) = c_p + \alpha t_p$  than the one which it is using.

That is, a T2 equilibrium traffic assignment (T2-ETA) is the vector of arc flows  $\bar{x}^* \in \mathcal{X}$  reflecting a traffic assignment that puts every trip on a path having minimum generalized cost with respect to that trip's particular VOT  $\alpha$ . This generalization of classical traffic equilibrium admits a large, usually infinite, number of categories of trips in simultaneous equilibrium.

#### 3.2 T2 Equilibrium Traffic Assignment Theorem

All the above conditions reduce to a simple fact:

T2-WP implies that a T2 equilibrium traffic assignment has the same total generalized cost as a T2 min-path traffic assignment with the arc costs and times fixed at the level implied by the equilibrium flow.

We restate this fact as the

**T2-ETA Theorem.** The flow  $\bar{x}^* \in \mathcal{X}$  is a T2 equilibrium flow iff

$$G[\bar{x}^* | \bar{c}^*, \bar{t}^*] = G[\bar{y} | \bar{c}^*, \bar{t}^*] \quad (6)$$

where the fixed arc cost and time vectors are fixed at  $\bar{c}^* = (c_e(x_e^*))$  and  $\bar{t}^* = (t_e(x_e^*))$ , and as before:

$$\begin{aligned} G(\bar{x} | \bar{c}^*, \bar{t}^*) &= \int_{\mathcal{A}} \sum_e (c_e^* + \alpha t_e^*) x_e(\alpha) d\alpha \\ \bar{y} &\in \arg \min_{\bar{x} \in \mathcal{X}} G[\bar{x} | \bar{c}^*, \bar{t}^*]. \end{aligned}$$

**Proof.** We first state and prove a lemma regarding equilibrium for trips of a single given VOT  $\alpha^\circ \in \mathcal{A}$ :

**Lemma 3.1** Define a  $\mathcal{X}(\alpha)$  as a “projection” of  $\mathcal{X}$ :

$$\mathcal{X}(\alpha) = \{\text{feasible assignments of trips with VOT } \alpha\}$$

$$\bar{y}(\alpha) \in \arg \min_{\bar{x} \in \mathcal{X}(\alpha)} G(\bar{x}|\bar{c}, \bar{t}).$$

Now consider a single VOT  $\alpha^\circ$  and assume that for all the remaining  $\alpha \in \mathcal{A} - \{\alpha^\circ\}$ , the flows  $\bar{x}(\alpha) \in \mathcal{X}(\alpha)$ 's are equilibrium flows for their respective  $\alpha$ 's. Then the partial traffic assignment  $\bar{x}(\alpha^\circ) \in \mathcal{X}(\alpha^\circ)$  is in equilibrium iff

$$G[\bar{x}(\alpha^\circ)|\bar{c}^*, \bar{t}^*] = G[\bar{y}(\alpha^\circ)|\bar{c}^*, \bar{t}^*]. \quad (7)$$

**Proof.** Obvious: By definition, this is just the classical single fixed- $\alpha$  case: at equilibrium, all trips with the same  $o-d$  use a paths with same, minimum generalized cost.

We now prove Theorem T2-ETA. ( $\Rightarrow$ ): Follows from Lemma 3.1. ( $\Leftarrow$ ): Follows from the fact that the left-hand side of (7) can never be smaller than its right-hand side. Assume (6) is true and (7) is false for one or more  $\alpha$ 's; that is, one or more of the sums on the right are less than their partner on the left. Then the integral of the left-hand side of (7) has to be greater than that on the right-hand side: a contradiction.

**Corollary 3.1** The T2-ETA Theorem and Lemma 2.1 imply that the flow  $\bar{x}^* \in \mathcal{X}$  is a T2-ETA, i. e. all trips for all VOTs are simultaneously in equilibrium, iff

$$\int_{\alpha} \sum_{e \in \mathcal{E}} [c_e(x_e^*) + \alpha t_e(x_e^*)] x_e^*(\alpha) d\alpha = \int_{\alpha} \sum_{e \in \mathcal{E}} [c_e(x_e^*) + \alpha t_e(x_e^*)] y_e(\alpha) d\alpha.$$

### 3.3 T2 Equilibrium Traffic Assignment Algorithm

T2-ETA entails the simultaneous equilibration of an infinite number of trip classes. Fortunately, only the *total* flow  $x_e$  on each arc is interesting, since as seen in Section 1, the individual class arc flows  $x_e(\alpha)$  are not unique. This permits the construction of an efficient solution algorithm.

**Notation.** For purposes of symmetry, we effectively rename  $\bar{x}^*$  as  $\bar{x}^\circ$  and  $\bar{y}$  as  $\bar{x}$ . More precisely, the T2-ETA algorithm generates a sequence of  $\bar{x}^\circ$ 's approaching  $\bar{x}^*$  and in doing so, solves a series of T2 min-path traffic assignment problems—each solution being called  $\bar{x}$ .

**Lemma 3.2** If  $\bar{x}^\circ \in \mathcal{X}$  and

$$\begin{aligned} \bar{t}^\circ &= (t_e(x_e^\circ)), \quad \bar{c}^\circ = (c_e(x_e^\circ)) \\ \bar{x} &\in \arg \min_{\bar{y} \in \mathcal{X}} G[\bar{y} | \bar{c}^\circ, \bar{t}^\circ] \end{aligned} \quad (8)$$

$$\begin{aligned} u_e^\circ &= \int_{\mathcal{A}} \alpha x_e^\circ(\alpha) d\alpha, \quad u_e = \int_{\mathcal{A}} \alpha x_e(\alpha) d\alpha \\ \Delta x_e &= x_e - x_e^\circ, \quad \Delta u_e = u_e - u_e^\circ; \end{aligned} \quad (9)$$

then

$$G[\bar{x} | \bar{c}^\circ, \bar{t}^\circ] - G[\bar{x}^\circ | \bar{c}^\circ, \bar{t}^\circ] = \sum_{e \in \mathcal{E}} [c_e(x_e^\circ) \Delta x_e + t_e(x_e^\circ) \Delta u_e].$$

**Proof.**

$$\begin{aligned} G[\bar{x} | \bar{c}^\circ, \bar{t}^\circ] &= \int_{\alpha} \sum_{e \in \mathcal{E}} [c_e(x_e^\circ) + \alpha t_e(x_e^\circ)] x_e(\alpha) d\alpha \\ &= \sum_{e \in \mathcal{E}} \left[ \int_{\alpha} c_e(x_e^\circ) x_e(\alpha) d\alpha + \int_{\alpha} \alpha t_e(x_e^\circ) x_e(\alpha) d\alpha \right] \\ &= \sum_{e \in \mathcal{E}} \left[ c_e(x_e^\circ) \int_{\alpha} x_e(\alpha) d\alpha + t_e(x_e^\circ) \int_{\alpha} \alpha x_e(\alpha) d\alpha \right] \\ &= \sum_{e \in \mathcal{E}} [c_e(x_e^\circ) x_e + t_e(x_e^\circ) u_e]. \end{aligned} \quad (10)$$

$$\begin{aligned} G[\bar{x} | \bar{c}^\circ, \bar{t}^\circ] - G[\bar{x}^\circ | \bar{c}^\circ, \bar{t}^\circ] &= \sum_{e \in \mathcal{E}} [c_e(x_e^\circ)(x_e - x_e^\circ) + t_e(x_e^\circ)(u_e - u_e^\circ)] \\ &= \sum_{e \in \mathcal{E}} [c_e(x_e^\circ) \Delta x_e + t_e(x_e^\circ) \Delta u_e]. \end{aligned} \quad (11)$$

**Corollary 3.2** The T2-ETA Theorem says when the right-hand side of (11) vanishes,  $\bar{x}^\circ$  is a T2-ETA.

**Corollary 3.3** T2-ETA conditions can be stated as a variational inequality: The flow  $\bar{x}^* \in \mathcal{X}$  is T2-ETA iff

$$\sum_{e \in \mathcal{E}} [c_e(x_e^*)(x_e - x_e^*) + t_e(x_e^*)(u_e - u_e^*)] \geq 0.^2 \quad (12)$$

**Proof.** Define (for this proof only)  $\bar{x}^\circ$  as

$$\bar{x}^\circ \in \arg \min_{\bar{x} \in \mathcal{X}} G[\bar{x} | \bar{c}^*, \bar{t}^*]. \quad (13)$$

<sup>2</sup>Letting  $\bar{g}(\bar{v}) = (\bar{c}(\bar{v}), \bar{t}(\bar{v}))$  and  $\bar{v} = (\bar{x}, \bar{u})$  gives (12) a more familiar VI look:  $\bar{g}(\bar{v}) \cdot \Delta \bar{v} \geq 0$ .

That is, for all  $\bar{x} \in \mathcal{X}$ ,

$$\sum_{e \in \mathcal{E}} [c_e(x_e)x_e + t_e(x_e)u_e] - \sum_{e \in \mathcal{E}} [c_e(x_e^*)x_e^o + t_e(x_e^*)u_e^o] \geq 0. \quad (14)$$

By the Corollary 3.2, the flow  $\bar{x}^*$  is T2-ETA iff

$$\sum_{e \in \mathcal{E}} [c_e(x_e^*)x_e^o + t_e(x_e^*)u_e^o] - \sum_{e \in \mathcal{E}} [c_e(x_e^*)x_e^* + t_e(x_e^*)u_e^*] = 0. \quad (15)$$

Adding (14) and (15) yields

$$\sum_{e \in \mathcal{E}} [c_e(x_e^*)x_e + t_e(x_e^*)u_e] - \sum_{e \in \mathcal{E}} [c_e(x_e^*)x_e^* + t_e(x_e^*)u_e^*] \geq 0. \quad (16)$$

Or,

$$\sum_{e \in \mathcal{E}} [c_e(x_e^*)(x_e - x_e^*) + t_e(x_e^*)(u_e - u_e^*)] \geq 0. \quad (17)$$

Thus, the T2-ETA problem reduces to a variational inequality problem in *only two* state variables per arc,  $x_e$  and  $u_e$ . This permits its solution with existing VIP algorithms. One such algorithm is Magnanti and Perakis's Generalized Frank-Wolfe algorithm (GFW), which they have shown to converge if the variation's Jacobian is symmetric [20]. The following elementary algorithm adapts GFW to T2. Although the Jacobian of (17) is not necessarily symmetric, all my tests tend to converge—if slowly. (Later, in Section 3.5, we provide another algorithm, for a slightly easier problem, which always converges.)

### 3.4 Algorithm T2-ETA (Appendix A, Figure A4)

This algorithm starts with  $\bar{x}^o$  being a T2 min-path assignment for  $\bar{c}^o = \bar{c}(0)$  and  $\bar{t}^o = \bar{t}(0)$ , then modifies it to drive (11) towards 0.

0. **[initialization]** Do a T2 min-path assignment to obtain:

$$\bar{x}^o \in \arg \min_{\bar{x} \in \mathcal{X}} G[\bar{x} | \bar{c}^o, \bar{t}^o]$$

$$x_e^o \leftarrow \int_{\mathcal{A}} x_e(\alpha) d\alpha, \quad u_e^o \leftarrow \int_{\mathcal{A}} \alpha x_e(\alpha) d\alpha.$$

1. **[ascent direction]** Fix all arc costs and times:

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

and perform a T2 min-path assignment to obtain:

$$\bar{x} \in \arg \min_{\bar{y} \in \mathcal{X}} G[\bar{y} | \bar{c}^o, \bar{t}^o]$$

$$\Delta x_e \leftarrow \int_{\mathcal{A}} x_e(\alpha) d\alpha - x_e^o, \quad \Delta u_e \leftarrow \int_{\mathcal{A}} \alpha x_e(\alpha) d\alpha - u_e^o.$$

2. [termination test] Let:

$$L[\lambda, (x_e), (u_e)] = \sum_{e \in \mathcal{E}} [c_e(x_e^o + \lambda \Delta x_e)x_e + t_e(x_e^o + \lambda \Delta x_e)u_e].$$

If

$$\frac{L[0, (\Delta x_e), (\Delta u_e)]}{L[0, (x_e), (u_e)]} < \epsilon,$$

quit:  $(x_e^o)$  implies an  $(\epsilon)$ -approximate T2-ETA; else go to Step 3.

3. [step size:] Determine step size  $\lambda^*$ : Let

$$\lambda^* \leftarrow \begin{cases} 1 & \text{if } L[1, (\Delta x_e), (\Delta u_e)] \leq 0 \\ \arg_{\lambda \in (0,1)} L[\lambda, (\Delta x_e), (\Delta u_e)] = 0 & \text{otherwise.} \end{cases}$$

4. [improved solution] Update all  $x_e^o$  and  $u_e^o$ :

$$x_e^o \leftarrow x_e^o + \lambda^* \Delta x_e, \quad u_e^o \leftarrow u_e^o + \lambda^* \Delta u_e$$

and return to Step 1.

### 3.4.1 The Calculation of $u_e$

The calculation of the integral  $u_e$  (9) warrants comment. Recall that for each  $o$ - $d$  pair, the likely paths induce a partitioning on the range of  $\alpha$ , associating with each likely path  $p$  the exclusive range of  $\alpha$  for which that  $p$  is a min-GC path. Because there are a finite number of likely paths, this partitioning permits the exact calculation of the  $u_e$  integral as the sum of discrete quantities. Let:

$$\begin{aligned} f_{od}(\alpha) &= \text{VOT PDF for trips from } o \text{ to } d \\ v_{od} &= \text{total trips from } o \text{ to } d \\ o_p &= \text{path } p \text{'s origin node} \\ d_p &= \text{path } p \text{'s destination node} \\ \mathcal{P}_e &= \text{set of likely paths that use arc } e \\ a_p &= \text{lower limit of VOT's for which path } p \text{ is optimal} \\ b_p &= \text{upper limit of VOT's for which path } p \text{ is optimal} \\ x_{ep}(\alpha) &= \text{trips with VOT } \alpha \text{ assigned to path } p \in \mathcal{P}_e \\ &= \begin{cases} v_{o_p d_p} f_{o_p d_p}(\alpha) & \text{if } a_p \leq \alpha < b_p \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

When Algorithm T2-MPA assigns a path's flow to its arcs, it knows these values. This permits an alternative expression for  $u_e$ :

$$u_e = \int_{\mathcal{A}} \alpha x_e(\alpha) d\alpha$$

$$\begin{aligned}
&= \sum_{p \in \mathcal{P}_e} \int_{\mathcal{A}} \alpha x_{ep}(\alpha) d\alpha \\
&= \sum_{p \in \mathcal{P}_e} \int_{a_p}^{b_p} \alpha v_{o_p d_p} f_{o_p d_p}(\alpha) d\alpha \\
&= \sum_{p \in \mathcal{P}_e} v_{o_p d_p} \int_{a_p}^{b_p} \alpha f_{o_p d_p}(\alpha) d\alpha. \tag{18}
\end{aligned}$$

This last summand is the product of a known origin-destination trip volume times the first moment of  $\alpha$  from  $a_p$  to  $b_p$ . Computing  $(u_e)$  is, therefore, merely to accumulate “link loads” of this summand. This explains the reference to  $u_e$  in the procedure `loadPath()` and shows how to obviate storing any  $x_e(\alpha)$ .

### 3.4.2 Example 1: Algorithm T2-ETA

Figure 8 poses an example simple enough for manual calculation. The network has only two arcs: the upper arc is free and has a time equal to its flow; the lower arc has a fixed cost of 1 unit and has a time equal to twice its flow. Ten trips go from node 1 to node 2; their VOT PDF is triangular between zero and one.

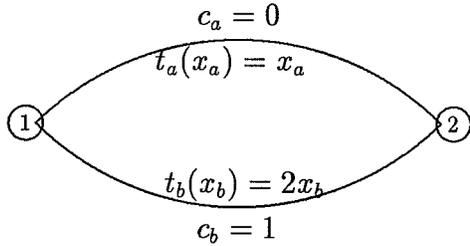


Figure 8(a): Two-Arc Network

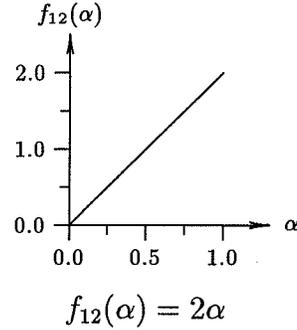


Figure 8(b): VOT PDF

Figure 8: Two-Arc Equilibrium Example: Input

Figures 9(a-c) show three iterations of Algorithm T2-ETA applied to the input data in Figures 8(a-b). Each iteration is described with four figures: The leftmost figure shows the beginning state of the algorithm:  $\bar{x}^o$ ,  $\bar{u}^o$  and  $\bar{t}(x^o)$  ( $c$  is constant). The next figure is the efficient frontier implied by the arc times and costs at the beginning state: the dot on the  $t$ -axis represents the upper arc and the dot on the  $c$ -axis the lower arc. The slope of the line connecting these two dots give  $\alpha_o$ , the threshold VOT at which trips shift from the upper arc to the lower. The third figure shows the PDF (*not* the CDF, hence its range is  $[0, 2]$ ) and the fraction of trips that the ascent direction assigns to

the upper arc (the remainder goes to the lower arc.) The fourth figure is the ascent direction, its arc flows  $x$  and moments  $u$ . Note that, in this simple example with only two arcs and one triangular PDF, there are easy formulae for the decision variables:

$$\begin{aligned}x_a &= v_{12} \int_0^{\alpha_o} f_{12}(\alpha) d\alpha = 10 \int_0^{\alpha_o} 2\alpha d\alpha = 10\alpha_o^2, \\x_b &= 10 - x_a \\u_a &= x_a \int_0^{\alpha_o} \alpha f_{12}(\alpha) d\alpha = x_a \int_0^{\alpha_o} 2\alpha^2 d\alpha = \frac{2}{3}x_a\alpha_o^3, \\u_b &= 6.7 - u_a.\end{aligned}$$

Furthermore, since  $\bar{c} = (0, 1)$  and  $\bar{t} = (x_a, 2x_b)$ , the solution to

$$\sum_{e \in \mathcal{E}} [c_e(x_e^o + \lambda^* \Delta x_e) \Delta x_e + t_e(x_e^o + \lambda^* \Delta x_e) \Delta u_e] = 0$$

is simply

$$\lambda^* = -\frac{x_a \Delta u_a + \Delta x_b + 2\Delta u_b x_b}{\Delta u_a \Delta x_a + 2\Delta u_b \Delta x_b}.$$

The subtitle shows the relative disequilibrium of the iteration's beginning state and the optimal fraction  $\lambda^*$  for combining the beginning state with the ascent direction to obtain an improved beginning state for the next iteration.

For example, the first iteration starts with all 10 trips on the upper arc, implying arc times of 10 and 0 (with fixed costs of 0 and 1) for the upper and lower arc respectively, and producing points (10,0) and (0,1) on the EF. A line connecting these points yields a threshold  $\alpha_o = 0.1$ . Thus, the ascent direction has

$$\begin{aligned}x_a &= 10\alpha_o^2 = 0.1 \\x_b &= 10 - x_a = 9.9\end{aligned}$$

total trips on the upper and lower arc respectively. Similarly,

$$\begin{aligned}u_a &= \frac{2}{3}x_a\alpha_o^3 = 0.0 \\u_b &= 6.7 - u_a = 6.7 \\ \lambda^* &= -\frac{10(-6.7) + 9.9 + 2(6.7)(9.9)}{(-6.7)(-9.9) + 2(6.7)(9.9)} = 0.287.\end{aligned}$$

By adding  $0.287\Delta\bar{x}$  to  $\bar{x}^o$  and  $0.287\Delta\bar{u}$  to  $\bar{u}^o$ , we obtain the beginning state of Iteration 2, whose relative error of 0.03 is significantly lower than the 5.69 of Iteration 1.

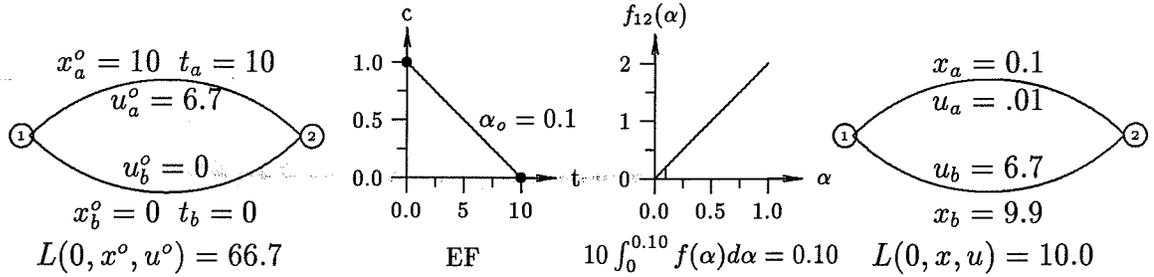


Figure 9(a): Iteration 1:  $s = 5.69$ ;  $\lambda^* = 0.287$

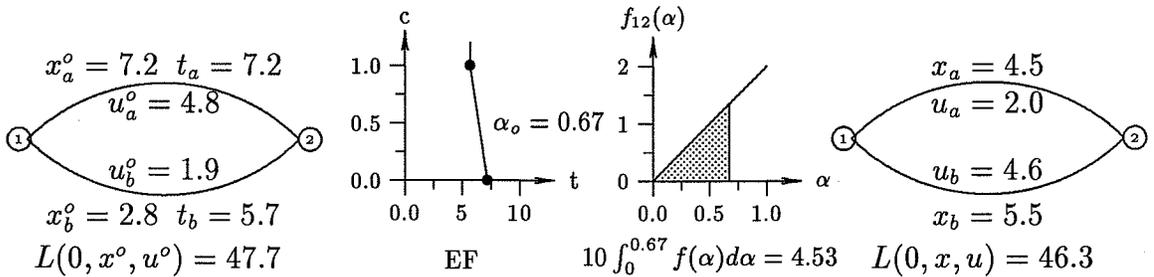


Figure 9(b): Iteration 2:  $s = 0.03$ ;  $\lambda^* = 0.066$

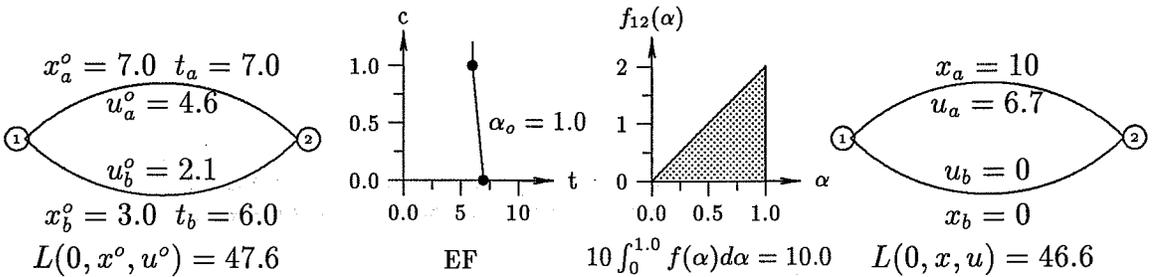


Figure 9(c): Iteration 3:  $s = 0.02$ ;  $\lambda^* = 0.053$

Figure 9: Ex. 1: Three Iterations of Algorithm T2-ETA

Table 1 below shows the results of the first ten iterations of this example in Figure 9. Equilibrium occurs at  $x_a = 7.06$ —which can be verified by noting that at the corresponding arc times, arc  $a$  is used by all trips with  $\alpha \leq 0.84$ ; and when  $\alpha = 0.84$ ,  $10\alpha^2 = 7.06 = x_a = t_a$ . Thus, the algorithm converges quickly to within a few percent of the right answer, but then begins to "tail." This is a common problem with the Frank-Wolfe algorithm. I have had the same experience running Algorithm T2-ETA on larger networks.

n	$x_a^o$ $x_a$	$x_b^o$ $x_b$	$u_a^o$ $u_a$	$u_b^o$ $u_b$	$L(0, x^o, u^o)$ $L(0, x, u)$	$c_a^o$ $\alpha$	$c_b^o$ $s$	$t_a^o$ $\lambda^*$	$t_b^o$
1	10.0	0.0	6.7	0.0	66.7	0	1	10.0	0.0
	0.1	9.9	0.0	6.7	10.0	0.10	5.69	0.287	
2	7.2	2.8	4.8	1.9	47.7	0	1	7.2	5.7
	4.5	5.5	2.0	4.6	46.3	0.67	0.03	0.066	
3	7.0	3.0	4.6	2.1	47.6	0	1	7.0	6.0
	10.0	0.0	6.7	0.0	46.6	1.00	0.02	0.053	
4	7.1	2.9	4.7	2.0	47.6	0	1	7.1	5.7
	4.8	5.2	2.2	4.4	46.4	0.69	0.03	0.070	
5	7.0	3.0	4.5	2.2	47.5	0	1	7.0	6.0
	10.0	0.0	6.7	0.0	46.6	1.00	0.02	0.050	
6	7.1	2.9	4.6	2.0	47.6	0	1	7.1	5.7
	5.1	4.9	2.4	4.2	46.5	0.71	0.02	0.077	
7	7.0	3.0	4.5	2.2	47.5	0	1	7.0	6.0
	10.0	0.0	6.7	0.0	46.5	1.00	0.02	0.048	
8	7.1	2.9	4.6	2.1	47.5	0	1	7.1	5.8
	5.4	4.6	2.6	4.0	46.6	0.73	0.02	0.087	
9	7.0	3.0	4.4	2.3	47.4	0	1	7.0	6.1
	10.0	0.0	6.7	0.0	46.5	1.00	0.02	0.047	
10	7.1	2.9	4.5	2.2	47.4	0	1	7.1	5.8
	5.6	4.4	2.8	3.8	46.6	0.75	0.02	0.102	

Table 1: Ten Iterations of T2-ETA (Two-Arc Example)

### 3.4.3 Example 2: Sampled Iterations of T2-ETA

To give a sense of Algorithm T2-ETA's behavior, Figure 10 shows the results of a sampling of four iterations of the algorithm applied to the example in Figures 5 and 4 in Section 1. Notice that the error in Figure 10(d): Iteration 10 is about 4 percent of the precise equilibrium flows in Figure 7(a). Due to the infamous "tailing problem" to duplicate them would require several more iterations. Fortunately, approximate equilibria suffice for most practical applications.

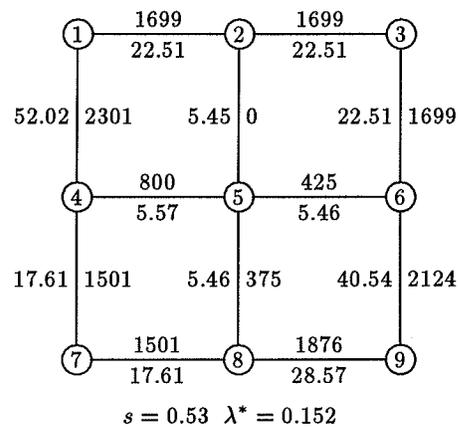
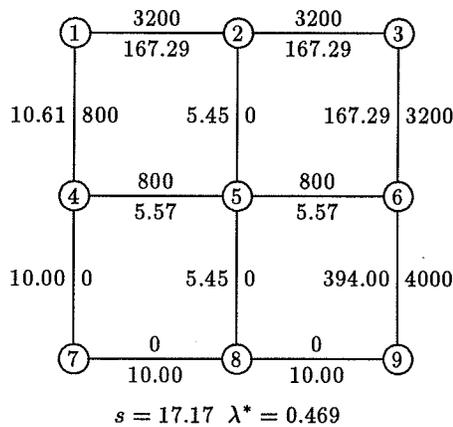


Figure 10(a): Iteration 1: Flow and Time

Figure 10(b): Iteration 2: Flow and Time

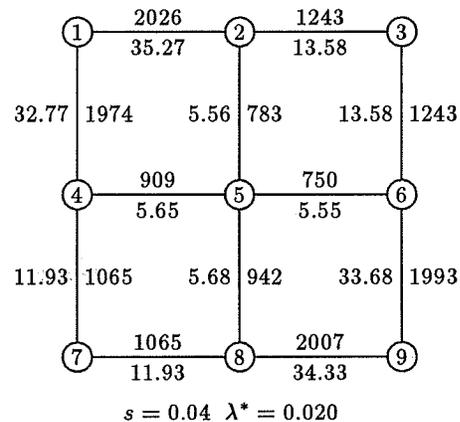
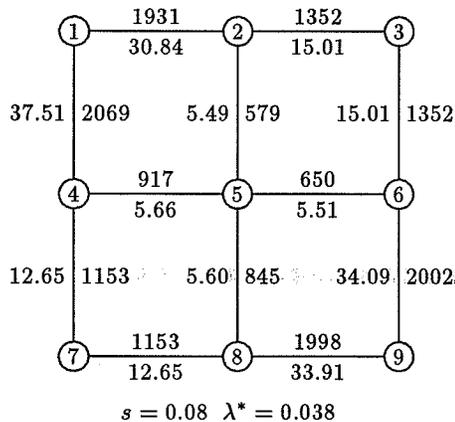


Figure 10(c): Iteration 6: Flow and Time

Figure 10(d): Iteration 10: Flow and Time

Figure 10: Ex. 2: Sampled Iterations of Algorithm T2-ETA

### 3.5 T1.5 Equilibrium Traffic Assignment

Because the proof of convergence of Algorithm T2-ETA is outside the scope of this paper, we offer for the skeptic an alternative algorithm, which solves a simpler problem, but whose convergence is readily proved. This simpler problem allows only arc *times* to vary with flow; arc costs are *fixed*; furthermore VOTs are strictly positive. Due to the major restriction on arc costs and minor constraint on  $\alpha$ , the model is not quite bicriterion; so, we call it T1.5 equilibrium traffic assignment—or T1.5-ETA.

**Lemma 3.3** If  $0 < \alpha \in \mathcal{A}$  then  $\bar{x}^*$  is a T1.5-ETA iff for all  $\bar{x} \in \mathcal{X}$ ,

$$\int_{\alpha} \sum_{e \in \mathcal{E}} \left[ \frac{c_e}{\alpha} + t_e(x_e^*) \right] [x_e(\alpha) - x_e^*(\alpha)] d\alpha \geq 0. \quad (19)$$

**Proof.** Corollary 3.1 implies equilibrium occurs at  $\bar{x}^*$  iff

$$\int_{\alpha} \sum_{e \in \mathcal{E}} \left[ \frac{c_e}{\alpha} + t_e(x_e^*) \right] [y_e(\alpha) - x_e^*(\alpha)] d\alpha = 0. \quad (20)$$

Where

$$\bar{y} \in \arg \min_{\bar{x} \in \mathcal{X}} \int_{\alpha} \sum_{e \in \mathcal{E}} \left[ \frac{c_e}{\alpha} + t_e(x_e^*) \right] x_e(\alpha) d\alpha,$$

i. e., for all  $\bar{x} \in \mathcal{X}$

$$\int_{\alpha} \sum_{e \in \mathcal{E}} \left[ \frac{c_e}{\alpha} + t_e(x_e^*) \right] (x_e(\alpha) - y_e(\alpha)) d\alpha \geq 0. \quad (21)$$

Adding (20) and (21) proves the lemma.

**T1.5-ETA Theorem (Leurent [17]).** Flow  $\bar{x}^*$  is a T1.5-ETA if

$$\bar{x}^* \in \arg \min_{\bar{x} \in \mathcal{X}} Q(\bar{x}),$$

where

$$Q(\bar{x}) = \sum_{e \in \mathcal{E}} \left[ c_e \int_{\mathcal{A}} \frac{x_e(\alpha)}{\alpha} d\alpha + \int_0^{x_e} t(v) dv \right] \quad (22)$$

**Proof.** Suppose inequality (19) were the directional derivative of  $Q()$  at  $\bar{x}^*$  in the direction  $\bar{x} - \bar{x}^*$ ; then if  $Q()$  were convex in  $\mathcal{X}$ , (19) would be a necessary and sufficient for  $\bar{x}^*$  to be  $Q()$ 's global minimand. Inequality (19) is  $Q()$ 's specified derivative if [4]

$$\frac{\partial Q}{\partial x_e(\alpha)} = \frac{c_e}{\alpha} + t_e(x_e),$$

which is clearly the case:

$$\begin{aligned}
\frac{\partial Q}{\partial x_e(a)} &= \frac{\partial}{\partial x_e(a)} \left( \sum_{e \in \mathcal{E}} \left[ \int_{\mathcal{A}} \frac{c_e x_e(\alpha)}{\alpha} d\alpha + \int_0^{x_e} t(v) dv \right] \right) \\
&= \frac{\partial}{\partial x_e(a)} \left( \int_{\mathcal{A}} \frac{c_e x_e(\alpha)}{\alpha} d\alpha \right) + \frac{\partial}{\partial x_e(a)} \left( \int_0^{x_e} t(v) dv \right) \\
&= \frac{c_e}{\alpha} + \frac{\partial}{\partial x_e} \left( \int_0^{x_e} t(v) dv \right) \frac{\partial x_e}{\partial x_e(a)} = \frac{c_e}{\alpha} + t_e(x_e^*).
\end{aligned}$$

That  $Q()$  is convex in  $\mathcal{X}$  is also clear. Its Hessian is zero everywhere except on its main diagonal, which has  $t'_e(x_e)$  as element  $(e, e)$ . Assuming  $\bar{t}()$  is nondecreasing and twice differentiable for  $(x_e(\alpha)) \geq 0$ , the Hessian is symmetric and positive semidefinite, making  $Q()$  convex in  $\mathcal{X}$ [4]. This completes the proof.

### 3.6 Algorithm T1.5-ETA (Appendix A, Figure A5)

Since  $Q()$  is convex, we can find its minimand using the well-know Frank-Wolfe (FW) algorithm [12] [18] [23]. However, let us use an indirect approach that reuses Algorithm T2-ETA. As before, rename  $\bar{x}^*$  as  $\bar{x}^o$  and  $\bar{y}$  as  $\bar{x}$ , but now redefine  $u_e$  as:

$$u_e = \int_{\mathcal{A}} \frac{x_e(\alpha)}{\alpha} d\alpha.$$

In the manner of Lemma 3.2, we can easily prove that  $x^o$  is a T1.5-ETA iff

$$\sum_{e \in \mathcal{E}} [c_e \Delta u_e + t_e(x_e^o) \Delta x_e] = 0.$$

Changing the definitions of the procedures L() and H() in Section 3.4 (and Figure A4, Appendix A to those in Figure A5, Appendix A), and changing (18) to read:

$$u_e = \sum_{p \in \mathcal{P}_e} v_{o_p d_p} \int_{a_p}^{b_p} \frac{f_{o_p d_p}(\alpha)}{\alpha} d\alpha$$

we can run this slightly revised Algorithm T2-ETA to perform in effect the appropriate FW.

Algorithm T1.5-ETA is an instance of what Perakis and Magnanti [20] [21] [22] call Generalized Frank-Wolfe (GFW), which solves variational inequalities directly, obviating a formula for a function being minimized. If the Jacobian of the GFW functional (i. e., Hessian of the FW objective) is symmetric, GFW is equivalent to FW.

Note that T1.5-ETA's ascent direction is the negative of FW's descent direction, and that its line search minimizes  $Q(\bar{x}^o + \lambda \Delta \bar{x})$  by finding a zero for its

derivative for  $\lambda \in (0, 1)$ :

$$\begin{aligned}
\frac{\partial Q}{\partial \lambda^*} &= \frac{\partial}{\partial \lambda^*} Q(\bar{x}^o + \lambda^* \Delta \bar{x}) \\
&= \frac{\partial}{\partial \lambda^*} \left( \sum_{e \in \mathcal{E}} \left[ c_e \int_{\mathcal{A}} \frac{x_e^o(\alpha) + \lambda^* \Delta x_e(\alpha)}{\alpha} d\alpha + \int_0^{x_e^o + \lambda^* \Delta x_e} t(v) dv \right] \right) \\
&= \sum_{e \in \mathcal{E}} \left[ c_e \int_{\mathcal{A}} \frac{\Delta x_e(\alpha)}{\alpha} d\alpha + t(x_e^o + \lambda^* \Delta x_e) \Delta x_e \right] \\
&= \sum_{e \in \mathcal{E}} [c_e \Delta u_e + t(x_e^o + \lambda^* \Delta x_e) \Delta x_e] \\
&= L(\lambda^*) = 0.
\end{aligned}$$

**T1.5-ETA Convergence Theorem.** Algorithm T1.5-ETA, in Figure A5 of Appendix A, converges to an equilibrium traffic assignment (in perhaps an infinite number of iterations).

**Proof.** Follows directly from Lemma 3.3 and the T1.5-ETA Theorem. Since Algorithm T2.5-ETA uses GFW in a case where GFW and FW are equivalent, it behaves like FW.

## 4 Conclusion

### 4.1 Summary

This paper covered the basic theory and elementary algorithms for a bicriterion user-optimal equilibrium traffic assignment model dubbed T2. It defined T2's behavioral assumptions, derived its mathematical formulation, and designed its solution procedures. Specifically, it extended Wardrop's Principle to include a stochastic value-of-time, produced a simple equation that reflected an equilibrium state, and presented four novel and space-efficient, but slow, algorithms:

- a bicriterion parametric min-GC path algorithm, LikelyPaths,
- a bicriterion min-GC path assignment algorithm, T2-MPA,
- a bicriterion equilibrium assignment procedure, T2-ETA, that solved problems where both arc time and cost are flow-dependent, and
- a one-and-a-half-criterion equilibrium procedure, T1.5-ETA, that permits only time to depend on flow.

T2-ETA is more robust than T1.5-ETA but lacks the latter's proof of convergence. LikelyPaths and T2-MPA provide support for either of these two equilibrium assignment procedures, which produce estimates of the total arc flow accruing when each trip uses its particular min-GC path

### 4.2 Additional Research and Development

Besides dynamic assignment and VOT PDF estimation, which were mentioned in Section 1.6, several other problem areas come to mind for T2-related additional research and development.

**Algorithm T2-ETA Convergence Proof.** The convergence of Algorithm T1.5-ETA's was easy to prove; however, it is beyond this writer's ability to prove that T2-ETA converges. Accordingly, I trust that a later paper will prove it or some variant does—as my tests would indicate.

**Algorithm Speedup.** Intended merely to show that T2 makes sense and has an accessible solution, the algorithms presented here are “elementary;” they are unsuitable for real-world applications. Accordingly their performance received no discussion.

The complications attending more sophisticated algorithms appropriate for “productions code” would obscure this paper's aim. Speedups from sophisticated algorithms are the subject of a future paper. For example, to find a

feasible descent direction, Algorithm T2-MPA enumerates likely paths, as opposed to using an approach based on min-path *trees*. T2 can, however, be implemented with a parametric *tree* building algorithm [24]. This improvement notwithstanding, the descent direction will still consume more computer cycles than the simpler classical model; therefore, T2 implementation must try to obviate steepest descent's tailing problem.

**Upper Bounds on Arc Flows.** If T2's model and algorithm incorporated prespecified upper bounds on arc flows in the manner proposed by Hearn [14], it would greatly improve its unique suitability for analyzing congestion pricing and, most especially, parking policy.

**Nonlinear Generalized Cost.** Every interesting result in this paper depends on the generalized cost of a path being a linear function of its cost and time. When GC is nonlinear, the GC of a path is no longer the sum of its arcs' GCs, making its min-path algorithm an interesting and worthy challenge.

**Elastic Demand.** T2 ignores the fact that the total trips  $v_{od}$  between an  $o$ - $d$  pair usually depend on the generalized cost of the trip, which in T2 is a random variable. Speaking theoretically, the inverse to this demand function maps  $v_{od}(\alpha)$  into  $g_{od}(\alpha)$ ; and the expected value of this inverse provides the expected generalized cost  $\mathbf{E}[g_{od}|v_{od}]$  among all trips from  $o$  to  $d$ . Hence, at equilibrium

$$\sum_e \mathbf{E}[g_e|x_e^*]x_e^* = \sum_e \mathbf{E}[g_e|x_e^*]y_e = \sum_{od} \mathbf{E}[g_{od}|v_{od}^*]v_{od}^*,$$

where  $y \in \arg \min_x G[x|c(x^*), t(x^*)]$ . These equations could be solved with a two-stage algorithm using T2-ETA. Built upon the work of Evans [10] and Leurent [17], the resulting algorithm would, for example, yield simultaneous destination-mode-route choice equilibrium.

**Tn: Beyond Two Criteria.** While time and cost are paramount, they certainly do not comprise all path-choice criteria. In various applications, other determinants of discomfort and inconvenience are influential, e.g., schedule reliability, number of transfers, safety, etc. Even the simplest mode-choice model would probably require time, cost, and inconvenience. Such a "T3" model uses two parameters: a value-of-time  $\alpha$  and a value-of-inconvenience  $\beta$ . A path's GC becomes  $g_p = c_p + \alpha t_p + \beta d_p$ , where  $d_p$  is path  $p$ 's inconvenience measure. For each  $o$ - $d$  pair, T3 uses a joint PDF  $f_{od}(\alpha, \beta)$ . The efficient frontier is a triangulated convex surface, with each vertex representing a path. The path-choice probability is the double integral of  $f_{od}(\alpha, \beta)$  over the values of  $\alpha$  and  $\beta$  that define the vertex's subgradient. T2's extension to handle more than two criteria is a fruitful subject for research.

**Applications.** The use of T2 and/or Tn to solve actual transportation planning problems presents many interesting potentialities for applications-oriented research and development. Four examples are the following:

- **Road pricing** is the most obvious application of T2. To plan toll roads, T2 could be used with an auto-driver trip table to determine traffic volume on arcs at various toll levels. It is the tool necessary to apply congestion pricing: using cost as a flow-dependent variable, an arc's "toll" can be related to the marginal cost of an additional trip; thus, users would pay an amount that reflected not only the direct cost to themselves but also the delay they cause other users.
- **Parking** receives simplistic treatment in the traditional approach to transportation planning, which ignores the fact that auto trips begin and end with a walk to and from a parking place. In most cities, there are several different opportunities to park, each with a different cost and walk time. The choice among these opportunities depends on the trip maker's value-of-time with respect to that trip. Besides walk links, T2 would allow links representing parking spaces with various costs and capacities. (An *elastic* parking costs, which increased with demand, could forecast a particular parking cost.) It would, then reasonably route a trip that parks and then walks to its destination.
- **Mode choice**, as already mentioned, could become more streamlined by using a T3 model with time, cost and inconvenience. The VOT PDF could be correlated with origin-destination parameters to provide necessary socio-demographic sensitivity. The model would be run as a combined mode-choice traffic assignment, which would expedite processing and assure continuity of travel time estimates. It would be interesting to compare the results of this simple model with a conventional logit model.
- **Land use forecasts** typically use a gravity-model approach. It is well known that a gravity model can be integrated into an equilibrium traffic assignment model[10]. These forecasts would likely improve if they used T2's path finding for determining the *expected cost*, time, and generalized cost separating zones. Furthermore, the cost of "settling" in a zone could be made elastic, depending on zone's attraction potential, and T2-ETA would solve this equilibrium problem.

### 4.3 Acknowledgements

Individuals from several institutions contributed directly and indirectly to the production of this paper:

**Academe.** I thank Tom Magnanti and Rob Freund of the Massachusetts Institute of Technology for their lucid math programming lectures; David Boyce of the University of Illinois at Chicago, who read several drafts this paper; David Bernstein of Princeton University for his helpful comments; Mike Florian and Patrice Marcotte of the University of Montreal, whose counter example [11] prompted my finding and repairing a serious error in an earlier draft; and particularly Georgia Perakis of MIT for introducing me to the generalized Frank-Wolfe algorithm and for her many helpful suggestions.

**Volpe Center.** I am very grateful for my excellent working environment and generous colleagues: the assistance Sarah Maccalous, the opinions of Mark Bucciarelli, the insights of Peter Mengert, and most importantly the encouragement of Mike Jacobs.

**Transportation Science.** Feedback from the Editor, Associate Editor, and two astute referees greatly improved this paper's form and content. I am indebted to their thoughtful reviews and very helpful suggestions.

**Department of Transportation.** Finally thanks go to the U. S. DOT, who with the Environmental Protection Agency, sponsored this work as part of its Transportation Model Improvement Program (TMIP); and to its technical representative, Fred Ducca, without whose support this paper would not have been written.

## Appendix A: Formal T2 Algorithms

<b>type</b> <i>Net</i>			data describing network
<i>Node Set</i>	$\mathcal{N}$		nodes in network
<i>Arc Set</i>	$\mathcal{E}$		arcs $(i, j)$ in network
<b>type</b> <i>Path</i>			data for min path $p^*$ (for VOT $\alpha$ )
<i>VOT</i>	$\alpha$		value-of-time used to find $p^*$
<i>Cost</i>	$c$		cost of min path: $c_{p^*}$
<i>Time</i>	$t$		time of min path: $t_{p^*}$
<i>GC</i>	$g$		GC of min path: $g_{p^*} = c_{p^*} + \alpha t_{p^*}$
<b>type</b> <i>Trips Set</i>			data describing trip matrix
<i>Node</i>	$o$		origin
<i>Node</i>	$d$		destination
<i>Real</i>	$v$		number of trips from $o$ to $d$
<b>type</b> <i>FlowPair Vector</i>			a traffic assignment (or differential)
<i>Real Array</i> [[ $\mathcal{E}$ ]]			arc total-flow vector (e.g. $\bar{x}, \bar{x}^o, \Delta\bar{x}$ )
<i>Real Array</i> [[ $\mathcal{E}$ ]]			arc first-moment vector (e.g. $\bar{u}, \bar{u}^o, \Delta\bar{u}$ )

**Figure A1: Data Types for All Algorithms**

```

FlowPair T2MPA(Net  $n$ , Trips  $\mathcal{V}$ , Real Function Array  $H$ )
     $(\bar{t}, \bar{c}^o) \leftarrow (\bar{t}(\bar{x}^o), \bar{c}(\bar{x}^o))$ 
     $(\bar{x}, \bar{u}) \leftarrow (0, 0)$ 
    for  $(o, d, v) \in \mathcal{V}$ 
         $(k, a) \leftarrow (\text{likelyPaths}(n, o, d), 0)$            (see Figure A3)
        for  $m \in \{1, \dots, k\}$ 
             $b \leftarrow (c_{p_m} - c_{p_{m-1}}) / (t_{p_{m-1}} - t_{p_m})$ 
             $\text{loadPath}(n, o, d, \alpha_{p_{m-1}}, a, b, v)$ 
             $a \leftarrow b$ 
         $\text{loadPath}(n, o, d, \alpha_{p_0}, a, \infty, v)$ 
    return  $(\bar{x}, \bar{u})$ 

loadPath(Net  $n$ , Node  $o, d$ , VOT  $\alpha$ ,  $a, b$ , Real  $v$ )
     $\text{minPath}(n, o, d, \alpha)$            (see Figure A3)
     $e \leftarrow q_d$ 
    while  $e \neq 0$ 
         $(x_e, u_e, e) \leftarrow (x_e + vF_{od}(a, b), u_e + vH_{od}(a, b), q_{i_e})$ 
    return.

```

**Figure A2: Algorithm T2-MPA**

**Integer likelyPaths**(*Net*  $n$ , *Node*  $o$ ,  $d$ , *Path Array*  $p$ )

$k \leftarrow 1$ ;	(count of likely paths)
$\hat{p} \leftarrow \text{minPath}(n, o, d, \infty)$	(fastest path)
$p_1 \leftarrow \text{minPath}(n, o, d, 0)$	(cheapest path)
$\text{midPaths}(n, o, d, \hat{p}, p, k)$	(likely paths between)
<b>return</b> ( $k + 1$ )	(count of $p$ found)

$\text{midPaths}$ (*Net*  $n$ , *Node*  $o$ ,  $d$ , *Path*  $\hat{p}$ , *Path Array*  $p$ , **Integer ref**  $k$ )

$\alpha \leftarrow (c_{\hat{p}} - c_{p_k}) / (t_{p_k} - t_{\hat{p}})$	(slope of connecting line)
$\tilde{p} \leftarrow \text{minPath}(n, o, d, \alpha)$	(find path in between)
<b>if</b> $g_{\tilde{p}} < c_{\hat{p}} + \alpha t_{\tilde{p}}$	(check if done)
$\text{midPaths}(n, o, d, \tilde{p}, p, k)$	(paths between)
$\text{midPaths}(n, o, d, \hat{p}, p, k)$	(paths between)
<b>else</b> $k \leftarrow k + 1$	(bump path count)
$p_k \leftarrow \hat{p}$	(stash $\hat{p}$ in its spot)

*Path*  $\text{minPath}$ (*Net* ( $\mathcal{N}, \mathcal{E}$ ), *Node*  $o$ ,  $d$ , *VOT*  $\alpha$ )

<b>for</b> $i \in \mathcal{N}$ $g_i \leftarrow \infty$	( $c_i$ is cost to node $i$ )
$(c_o, t_o, g_o, q_o, \mathcal{S}) \leftarrow (0, 0, 0, 0, \{o\})$	( $t_i$ is cost to node $i$ )
<b>for</b> $i \in \mathcal{S}$	( $g_i$ is GC to node $i$ )
<b>for</b> $e = (i, j) \in \mathcal{E}$	( $q_i$ is $i$ 's predecessor arc)
$(\hat{g}, \hat{c}, \hat{t}) \leftarrow (g_i + c_e + \alpha t_e(x_e^o), c_i + c_e, t_i + t_e)$	
<b>if</b> $\hat{g} < g_j \vee (\hat{g} = g_j \wedge \hat{c} < c_j) \vee (\hat{g} = g_j \wedge \hat{c} = c_j \wedge \hat{t} < t_j)$	
$(g_j, c_j, t_j, q_j, \mathcal{S}) \leftarrow (\hat{g}, \hat{c}, \hat{t}, e, \mathcal{S} \cap \{j\})$	
<b>return</b> $(\alpha, c_d, t_d, g_d)$ .	

**Figure A3: Algorithm LikelyPaths**

**FlowPair T2ETA**(*Net*  $n$ , *Trips*  $\mathcal{V}$ , **Real**  $\epsilon$ )

$(\bar{x}^o, \bar{u}^o) \leftarrow (0, 0)$  (start with zero flow)

$(\bar{x}^o, \bar{u}^o) \leftarrow T2MPA(n, \mathcal{V}, H)$  (T2 min-path assignment, Fig. A2)

**do**

$(\bar{x}, \bar{u}) \leftarrow T2MPA(n, \mathcal{V}, H)$  (T2 min-path assignment, Fig. A2)

$(\Delta\bar{x}, \Delta\bar{u}) \leftarrow (\bar{x} - \bar{x}^o, \bar{u} - \bar{u}^o)$  (ascent direction)

$s \leftarrow \frac{L(0, \Delta\bar{x}, \Delta\bar{u})}{L(0, \bar{x}, \bar{u})}$  (relative error)

**if**  $\epsilon \leq s$  (if not done, get step size)

**if**  $L(1, \Delta\bar{x}, \Delta\bar{u}) \leq 0$

$\lambda \leftarrow 1$

**else**  $\lambda \leftarrow \arg \min_{\lambda \in (0,1)} L(\lambda, \Delta\bar{x}, \Delta\bar{u}) = 0$

$(\bar{x}^o, \bar{u}^o) \leftarrow (\bar{x}^o + \lambda\Delta\bar{x}, \bar{u}^o + \lambda\Delta\bar{u})$  (and update flows)

**until**  $s < \epsilon$  (quit when rel. error small)

**return**  $(\bar{x}^o, \bar{u}^o)$ . (send back updated flow)

**Real**  $L(\mathbf{Real} \lambda, x, u)$

**return**  $\left( \sum_{e \in \mathcal{E}} [c_e(x_e^o + \lambda\Delta x_e)x_e + t_e(x_e^o + \lambda\Delta x_e)u_e] \right)$ .

**Real**  $H_{od}(VOT a, b)$

**return**  $\left( \int_a^b \alpha f_{od}(\alpha) d\alpha \right)$ .

**Real**  $F_{od}(VOT a, b)$  (**Prob** $[\alpha \in [a, b]]$ )

**return**  $\left( \int_a^b f_{od}(\alpha) d\alpha \right)$ .

**Figure A4: Algorithm T2-ETA**

*FlowPair* T2ETA(*Net*  $n$ , *Trips*  $\mathcal{V}$ , **Real**  $\epsilon$ )

$(\bar{x}^o, \bar{u}^o) \leftarrow (0, 0)$  (start with zero flow)

$(\bar{x}^o, \bar{u}^o) \leftarrow T2MPA(n, \mathcal{V}, H)$  (T2 min-path assignment, Fig. A2)

**do**

$(\bar{x}, \bar{u}) \leftarrow T2MPA(n, \mathcal{V}, H)$  (T2 min-path assignment, Fig. A2)

$(\Delta\bar{x}, \Delta\bar{u}) \leftarrow (\bar{x} - \bar{x}^o, \bar{u} - \bar{u}^o)$  (ascent direction)

$s \leftarrow \frac{L(0, \Delta\bar{x}, \Delta\bar{u})}{L(0, \bar{x}, \bar{u})}$  (relative error)

**if**  $\epsilon \leq s$  (if not done, get step size)

**if**  $L(1, \Delta\bar{x}, \Delta\bar{u}) \leq 0$

$\lambda \leftarrow 1$

**else**  $\lambda \leftarrow \arg_{\lambda \in (0,1)} L(\lambda, \Delta\bar{x}, \Delta\bar{u}) = 0$

$(\bar{x}^o, \bar{u}^o) \leftarrow (\bar{x}^o + \lambda\Delta\bar{x}, \bar{u}^o + \lambda\Delta\bar{u})$  (and update flows)

**until**  $s < \epsilon$  (quit when rel. error small)

**return**  $(\bar{x}^o, \bar{u}^o)$ . (send back updated flow)

**Real**  $L(\mathbf{Real} \lambda, \bar{x}, \bar{u})$  (different from Algo T2-ETA)

**return**  $\left( \sum_{e \in \mathcal{E}} [c_e u_e + t_e(x_e^o + \lambda \Delta x_e) x_e] \right)$ .

**Real**  $H_{od}(VOT a, b)$  (differnt from Algo T2-ETA)

**return**  $\left( \int_a^b \alpha^{-1} f_{od}(\alpha) d\alpha \right)$ .

**Real**  $F_{od}(VOT a, b)$  (**Prob** $[\alpha \in [a, b]]$ )

**return**  $\left( \int_a^b f_{od}(\alpha) d\alpha \right)$ .

**Figure A5: Algorithm T1.5-ETA**

## References

- [1] R. Ahuja, T. Magnanti and J. Orlin, *Network Flows*, Prentice Hall (1993).
- [2] T. Apostol, *Mathematical Analysis*, Addison-Wesley (1968).
- [3] Anon, Presented as formal commentary at the 1994 Annual Conference of the Transportation Research Board by a well-known academic, who preferred that his name not be mentioned in connection with his statement.
- [4] M. Bazaraa, H. Sherali and C. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons (1979).
- [5] M. Ben Akiva, Personal Correspondence, Massachusetts Institute of Technology (1994).
- [6] D. Bernstein, Personal Correspondence, Massachusetts Institute of Technology (1993).
- [7] S. Dafermos, "A Multicriteria Route-Choice Traffic Equilibrium Model." Paper prepared for Frontiers in Transportation Equilibrium and Supply Models Symposia, University of Montreal (1981).
- [8] R. Dial, "A Probabilistic Multipath Assignment Model that Obviates Path Enumeration," *Transportation Research* **5**, pp 83-111 (1971).
- [9] R. Dial, "A Model and Algorithm for Multicriteria Route-Mode Choice." *Transportation Research, Part B. Methodological*, December (1979).
- [10] Evans, S. (1976) "Derivation and Analysis of Some Models for Combining Trip Distribution and Assignment." *Transportation Research* **10**, pp 37-57.
- [11] M. Florian and P. Marcotte, Personal Correspondence, University of Montreal (1994).
- [12] M. Frank and P. Wolfe "An Algorithm for Quadratic Programming." *Naval Research Logistics Quarterly*, **3** pp 95-110 (1956).
- [13] 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).
- [14] D. Hearn, "Bounded Flow Equilibrium Problems by Penalty Methods." *Proceedings of the IEEE International Conference on Circuit and Computers*, **1**, pp 162-166 (1980).
- [15] B. Janson, "Dynamic Traffic Assignment for Urban Road Networks." *Transportation Research*, **25B**, pp 143-161.

- [16] J. Kapur and H. Kesavan, *Entropy Optimization Principles with Applications*, Academic Press (1992). University of California Press, Berkely, CA.
- [17] F. Leurent, "Cost Versus Time Equilibrium over a Network." Presented at the 1994 Annual Conference of the Transportation Research board. Washington D. C (1994).
- [18] L. LeBlanc, *Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems*, Ph. D. thesis, Northwestern University (1973).
- [19] L. LeBlanc, B. Ran and D. Boyce, "Dynamic Travel Choice Models for Urban Transportation Networks." To appear in *Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetic* (1992).
- [20] T. Magnanti, and G. Perakis, "A Unifying Geometric Solution Framework and Complexity Analysis for Variational Inequalities." Working Paper OR 276-93, Operations Research Center, MIT, Cambridge, MA (1993).
- [21] T. Magnanti and G. Perakis, "On the Convergence of Classical Variational Inequality Algorithms." Working Paper OR 282-93, Operations Research Center, MIT, Cambridge, MA (1993).
- [22] T. Magnanti and G. Perakis, "From Frank-Wolfe to Steepest Descent: A Descent Framework for Solving VIPs." Presentation at TIMS/ORSA Joint National Meeting, Boston (1994).
- [23] S. Nguyen, "An Algorithm for the Traffic Assignment Problem." *Transportation Science*, 8, No. 3 pp 205-218 (1974) .
- [24] J. Orlin, "2-Criteria and 3-Criteria Shortest Path Problem: a Proposal," Personal Correspondence, Massachusetts Institute of Technology (1993).
- [25] R. Quandt, "A Probabilistic Abstract Mode Model." *Studies in Travel Demand, VIII*," Mathematica, Inc., Princeton, N. J. pp 127-149 (1967).
- [26] G. Roth, *Paying for Roads: the Economics of Traffic Congestion*, Penguin Books (1967).
- [27] M. Schneider, "Access and Land Development," *Proceedings of the conference on Urban Development Models*, Dartmouth College (1968).
- [28] Y. Sheffi and C. Daganzo, "On Stochastic Models of Traffic Assignment," *Transportation Science* 11, N. 3, pp 253-274 (1977).
- [29] B. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall (1986).

- [30] H. Slavin, Personal Correspondence, Caliper Corporation (1993).
- [31] J. Wardrop, "Some Theoretical Aspects of Road Traffic Research,"  
*Proc. Inst. of Civil Engineers, Part 2*, pp 325-378 (1952).