Computational Methods for Congestion Toll Pricing Models

D. W. Hearn, M. B. Yildirim, M. V. Ramana and L. H. Bai

Abstract— **In earlier work a toll pricing framework for congestion pricing has been developed for both fixed and elastic demand traffic assignment models as an alternative to traditional marginal social cost pricing. Within the framework it is possible to define alternative objectives such as minimizing toll costs, minimizing the number of toll booths, etc. This paper will report the results of recent computational experiments on methods designed for various toll pricing objectives.**

Keywords— **Congestion Toll Pricing, Marginal Social Cost Pricing, Toll Set Approximation, Cutting Plane Method.**

Introduction

In this paper, we present two computational methods for fixed demand (FD) toll pricing problems based on the toll pricing framework proposed by Hearn and Ramana [9]. The first method uses a toll set approximation to obtain alternative toll vectors. The second one is a cutting plane method for the minimum toll revenue problem.

Background

Let $\mathcal{G}$ denote the network model of a transportation system which consists of streets, $\mathcal{A}$, and intersections, $\mathcal{N}$. Mathematically, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ is a network, where $\mathcal{N}$ is the node set and $\mathcal{A}$ is the link set. Let A be the node-arc incidence matrix of $\mathcal{G}$. We define a commodity by an origin, p, and a destination, q. Let $\mathcal{K}$ represent the set indexing all such origin-destination pairs, $k = (p, q)$. The demand between an origin-destination pair is expressed as $\bar{t}_k$. The kth *commodity flow* vector is denoted by x^k and the sum of all the commodity flow vectors is the *aggregate flow* vector v. We assume that a continuously differentiable **cost map** $s : \mathcal{A} \rightarrow \mathcal{A}$ is given. ∇s denotes the Jacobian of s. When the aggregate flow on the network is v, the travel time (or cost) for a user on arc a is given by $s_a(v)$. We assume that $s_a(v)$ is strictly monotone and $s(v)^T v$ is strictly convex. The system defining the feasible flows is given by:

$$\begin{array}{rcll} v & = & \sum_{k \in \mathcal{K}} x^k & \\ Ax^k & = & E_k \bar{t}_k & \forall k \in \mathcal{K} \\ x^k & \geq & 0 & \forall k \in \mathcal{K} \end{array}$$

where $E_k = e_p - e_q$, a column incidence vector for commodity k, and e_p and e_q are unit vectors. The first constraint is the **aggregate flow** constraint and the second set of constraints are **network balance** constraints. We define V as the set of all feasible flows.

The user problem is defined by Wardrop's first principle [13]: For each origin-destination pair, the travel times on all utilized routes are equal, and less than those which would be experienced by a single vehicle on any unused route. Mathematically, a feasible aggregate flow vector $\bar{v}$ is a tolled user equilibrium (UOPT-FD) solution if and only if

$$(s(\bar{v}) + \beta)^T (v - \bar{v}) \geq 0 \qquad \forall v \in V$$

where users are tolled by an amount of β_a on link a. The untolled user problem has $\beta = 0$.

In the system problem (SOPT-FD), the objective changes to minimizing the total travel cost. Mathematically,

$$\begin{array}{l} \min s(v)^T v \\ \quad \text{s. t.} \quad v \in V. \end{array}$$

It has been observed that the untolled user and system solutions can differ significantly [9]. As a result, a traffic planner might toll the users in order to obtain a better utilization of the transportation network. In the next section, we will give the set of all tolls that will make the tolled user problem have the same solution as the system solution.

Toll Pricing Methods for Fixed Demand Networks

For a given system optimal flow vector $v^* \in V$, let $\mathcal{T}$ be the set of all tolls that ensure that v^* is a solution to the tolled user equilibrium problem. Hearn and Ramana [9] prove the following lemma:

Lemma 1: $\mathcal{T} = W_{FD}(v^*)$, the polyhedron given by the β part of the following linear inequality system in β and ρ variables:

$$\begin{array}{rcll} s(v^*) + \beta & \geq & A^T \rho^k & \forall k \in \mathcal{K} \\ (s(v^*) + \beta)^T v^* & = & \sum_{k \in K} \bar{t}_k E_k^T \rho^k. & \end{array}$$

From the optimality conditions of the system problem and Lemma 1, there always exists a toll vector, $\beta_{MSCP} = \nabla s(v^*) v^*$, of traditional *marginal social cost pricing* tolls. Thus the toll set is not empty. By possibly adding additional constraints and defining specific objectives, a traffic planner can obtain alternative toll vectors by solving linear and mixed integer programs. Examples for the objectives are (1) minimizing the total (nonnegative) toll revenue collected (MINSYS), (2) minimizing the largest nonnegative

D. W. Hearn, M. B. Yildirim and L. H. Bai are at the Center for Applied Optimization of Industrial and Systems Engineering Department at University of Florida. Email: {hearn}, {ybayram}, {lbai}@ise.ufl.edu

M. V. Ramana is an OR analyst at United Airlines. Email: ramana@cao.ise.ufl.edu

TABLE I
Alternative Optimization Formulations

TOLL	Objective (Z)	Extra Constraints ($\hat{\mathcal{T}}$)
MINSYS	$\beta^T v^*$	$\beta \geq 0$
MINMAX	z	$z \geq \beta_a,\ \forall a \in \mathcal{A}, \beta \geq 0$
MINTB	$\sum_{a \in \mathcal{A}} y_a$	$\beta_a \leq M y_a\ \forall a \in \mathcal{A}, y_a \in \{0,1\}, \beta \geq 0$

TABLE II
Alternative Toll Pricing Schemes for the Stockholm Network

Model	**Toll Booths**	**Total Tolls**	$\sum_{a \in \mathcal{A}} \bar{\beta}_a / \lvert\mathcal{A}\rvert$	**Max. Toll**	$\frac{\lvert\lvert \bar{v} - \tilde{v}^* \rvert\rvert}{\lvert\lvert \tilde{v}^* \rvert\rvert}\%$
MINSYS	192	2638640	1.59	131.57	2.03
MINMAX	301	5755080	4.13	27.51	2.08
MINTB	177	23417900	13.83	263.14	2.18
MSCP	914	34963959	36.04	423.41	0.82

toll collected (MINMAX) and (3) minimizing the number of toll booths (MINTB).

As shown in Table I, let $\hat{\mathcal{T}}$ be the set of additional (side) constraints and Z be the objective function for each toll pricing problem. The feasible region for each toll problem is the intersection of $\hat{\mathcal{T}}$ with $\mathcal{T}$. The objective is linear in each case, thus each toll pricing problem is either a linear program or a linear integer program.

To summarize, a traffic planner can solve the system problem to obtain the optimal flows and demands and define the toll set. Then using the problems defined above, alternative toll vectors can be obtained. This procedure is known as the *toll pricing framework* [9]. However, it requires that the system solution be exact. If this is not the case, the aggregate complementary constraint $((s(v^*) + \beta)^T v^* = \sum_{k \in K} \bar{t}_k E_k^T \rho^k$ will not hold and the toll set will be infeasible.

For example, consider the Stockholm network in [9] which has 417 nodes, 962 arcs and 1664 OD-pairs (46 commodities). As a result, the MINSYS problem has 20,145 variables and 44,298 inequalities. When the direct solution method using GAMS [6] and CPLEX [3] was applied, the MINSYS linear program was observed to be infeasible. Thus, Hearn and Ramana [9] penalized the aggregate complementarity constraint to obtain an approximate solution. Put in the general context, this yields the problem,

$$\begin{array}{ll} \min & u_1 Z + u_2 \left((s(v^*) + \beta)^T v^* - b^T \rho\right) \\ \text{s.t.} & s(v^*) + \beta \geq A^T \rho^k,\ \ \forall k \in \mathcal{K} \\ & (\beta, \rho, y, z) \in \hat{\mathcal{T}} \end{array}$$

where u_1 and u_2 are positive scalars. $\hat{\mathcal{T}}$ and Z are from Table I.

However, this method requires guessing values for u_1 and u_2 in order to find a good solution which minimizes the total infeasibility in the complementarity constraint and at the same time gives a good approximation of the toll pricing objective. For example, Hearn and Ramana [9], after some experimentation, chose $u_1 = 1$ and $u_2 = 1000$ to solve the MINSYS problem. Repeating the estimation of u_1 and u_2 for each toll pricing problem, proved to be difficult, or impossible, for the other objectives. However, it has been found that this can be avoided by employing an approximation to the toll set. $W_{FD}(\tilde{v}^*, \epsilon)$ is defined by the following set of the inequalities:

$$\begin{array}{rcl} s(\tilde{v}^*) + \beta & \geq & A^T \rho^k,\ \ \forall k \in \mathcal{K} \\ -\epsilon & \leq & (s(\tilde{v}^*) + \beta)^T \tilde{v}^* - b^T \rho \\ \epsilon & \geq & (s(\tilde{v}^*) + \beta)^T \tilde{v}^* - b^T \rho. \end{array}$$

The toll pricing framework which combines the penalty method and toll set approximation idea is summarized below:

Step 1: Solve the system optimum problem to obtain an approximately optimal solution $\tilde{v}^*$ and $s(\tilde{v}^*)$.

Step 2: Solve the MINSYS problem by penalizing the aggregate complementarity constraint. Let

$$\bar{\epsilon} = (s(\tilde{v}^*) + \bar{\beta}_{MINSYS})^T \tilde{v}^* - b^T \bar{\rho}$$

be the amount of violation in this constraint.

Step 3: Define the toll set approximation $W_{FD}(\tilde{v}^*, \bar{\epsilon})$.

Step 4: Define and optimize an a toll pricing problem using the toll set, $W_{FD}(\tilde{v}^*, \bar{\epsilon})$.

To illustrate, we present the results on the Stockholm network. Using the $\bar{\epsilon} = 5734.80$ obtained from MINSYS solution, $W_{FD}(\tilde{v}^*, \bar{\epsilon})$ is used to solve the other toll pricing problems. The results are presented in Table II. In this table, MINSYS tolls gives the minimum total toll revenue with 192 toll booths having an average toll of 1.59. The minimum number of toll booths is 177. However, the average toll for MINTB is significantly higher than the average toll for MINSYS. All of the alternative toll vectors outperform MSCP tolls in the total number of toll booths, the total tolls collected, the maximum toll, and the average toll. The relative error between the tolled user optimal solution, $\bar{v}$, and the approximate system optimal flows for each toll pricing problem is also listed in Table II. MSCP tolls have the smallest relative error, 0.08% of $||\tilde{v}^*||$ and this increases to 2.18% for MINTB tolls.

The MINSYS Cutting Plane Algorithm

For solving the large sized mathematical programming problems, several methods including column and row generation methods have been proposed. These methods usually take advantage of special structure induced by the resulting master and/or subproblem.

In this section, we present a cutting plane algorithm (CPA) for solving the MINSYS problem. The master problem is a linear program and the subproblem is a multicommodity shortest path problem. However, the multicommodity shortest path problem is decomposed into $|\mathcal{K}|$ shortest path problems. The CPA can be stated as:

1: Solve the SOPT problem to obtain the aggregate solution vector $v^* = \sum_{k \in \mathcal{K}} x^{k*}$ and the corresponding cost vector $s(v^*)$.

2: Start with any outer approximation of $\mathcal{T}$, the toll set, say, $\mathcal{T}^0 = \emptyset$. Let the iteration counter l be zero and $\epsilon \geq 0$ be the tolerance limit.

3: Solve the linear program which has the objective $\beta^T v^*$ and constraints $\mathcal{T}^l$

$$\min\{\beta^T v^* | \beta \in \mathcal{T}^l\}.$$

Let $\bar{\beta}_l$ be a solution.

4: Compute the shortest paths in the network with the cost vector being $s(v^*) + \bar{\beta}_l$. Let $v_l = \sum_{k \in \mathcal{K}} x_l^k$ be the extreme point solution obtained.

5: If

$$(s(v^*) + \bar{\beta}_l)^T (v_l - v^*) \geq -\epsilon,$$

then the tolled user equilibrium conditions are approximately satisfied. STOP.

6: If not, add the constraint(s) to $\mathcal{T}^l$ to obtain $\mathcal{T}^{l+1}$. In this step, either a single constraint,

$$(s(v^*) + \beta)^T (v_l - v^*) \geq 0,$$

or multiple constraints,

$$(s(v^*) + \beta)^T (x_l^k - x^{k*}) \geq 0, k \in S$$

can be added. S can be either $S = \{1, ..., K\}$ or

$$S = \{k' : (s(v^*) + \bar{\beta}_l)^T (x_l^{k'} - x^{k'*}) < -\epsilon\}.$$

7: Let $l = l + 1$, and repeat from Step 3.

To summarize, the master problem computes tolls, and the subproblem finds shortest paths having the tolled cost vector in the objective. In each iteration, new constraints are introduced to the master problem and the only change in the subproblem is the change in the cost vector. Thus both master problem and subproblem can be "warm started," since an initial basis is readily available when a new row is added to the master problem. Except for the cutting planes generated, the memory requirements and the size of the linear programs (LP) solved is $O(m)$, where m is the number of arcs in the network. A nice convergence property of the method is given in the following theorem.

Theorem 1: The cutting plane method converges finitely.

Proof: Let v_l be the vector generated and added at the l^{th} iteration. Obviously, there are finitely many of those points. The new cut

$$(s(v^*) + \beta)^T (v_l - v^*) \geq 0$$

is added since the equation

$$(s(v^*) + \bar{\beta}_l)^T (v_l - v^*) \leq -\epsilon$$

is violated. In the $(l+1)^{st}$ iteration, by solving the shortest path problem, the new flow vector can either be $v_{l+1} = v_m$ where $m < l + 1$ or $v_{l+1} \neq v_m$. In the first case, the algorithm terminates since

$$(s(v^*) + \bar{\beta}_{l+1})^T (v_{l+1} - v^*) \geq -\epsilon$$

is already satisfied. In the second case a new cut is added. By the finiteness of the such v_l the algorithm terminates in finite number of iterations, i.e., the method converges finitely. ■

Numerical Example

The CPA have been tested on the Nine Node problem [9] which has nine nodes, 18 arcs and four origin-destination pairs, and the New Sioux Falls Problem [1] which has 24 nodes, 76 arcs and 552 origin-destination pairs. Table III presents the system and user solutions. Note that there is a 8.96% difference between the system and user optimal solutions for the Nine Node problem. This difference is 3.97% for the Sioux Falls problem.

In Table IV, we present the performance of CPA method on Nine Node and Sioux Falls problems. For the small Nine Node problem, the CPA only shows an advantage in the number of variables and constraints because the total CPU time increases. However for the larger New Sioux Falls, the multicut CPA outperforms the direct solution approaches (using CPLEX [3] to solve the MINSYS directly) and the single cut CPA. The total CPU time for the multicut CPA is 15% less than the direct solution approaches. Further experimentation is needed to determine whether this improvement continues as the network size increases.

We have also tested if dropping nonbinding constraints improves the performance of the CPA or not, because we have observed that finding the nonbinding constraints consumes extra CPU time. From Table IV, the increase in CPU time does not justify the cut dropping strategy for the Sioux Falls problem since the problem size decreases negligibly. Apparently, dropping the nonbinding constraints does not improve the performance since most of the constraints in the master problem are binding.

Note that the multiple cut version of the cutting plane algorithm requires optimal commodity flow vectors x^{k*} such that $v^* = \sum_k x^{k*}$. However, the commodity flow vectors are not unique. Furthermore, it should be recognized that

TABLE III
The User and System Solutions for the Nine Node Problem and Sioux Falls Problem

	NINE NODE	**NEW SF***
Nodes	9	24
Arcs	18	76
OD Pairs	4	552
System Optimal Objective	2253.918	71.943
User Optimal Objective	1820.427	42.313
System Optimal Objective(User Solution)	2455.870	74.802
Difference between These Solutions.	8.96%	3.97%
$\|\|v^*\|\|$	98.806	112.787
$\|\|v^U\|\|$	105.661	108.677
$\|\|v^U - v^*\|\|$	25.951	13.791

TABLE IV
Experimental Results for CPA and Cut Dropping Strategies

Nine Node Problem						
	Number of Iterations	Shortest Path CPU	Master Pr. CPU	Total CPU	Number of Variables	Number Of Constraints
MINSYS				0.01	37	38
MULTICUT	7	0.03	0.01	0.04	17	13
MULTIDROP	7	0.00	0.06	0.06	17	13
SINGLECUT	9	0.03	0.03	0.06	17	9
SINGLEDROP	11	0.03	0.13	0.16	17	8
Nine Node Problem-Version II						
	Number of Iterations	Shortest Path CPU	Master Pr. CPU	Total CPU	Number of Variables	Number Of Constraints
MINSYS				0.02	37	38
MULTICUT	6	0.05	0.03	0.08	16	11
MULTIDROP	5	0.05	0.00	0.05	16	9
SINGLECUT	7	0.03	0.07	0.10	16	7
SINGLEDROP	7	0.02	0.13	0.15	16	7
New Sioux Falls Problem						
	Number of Iterations	Shortest Path CPU	Master Pr. CPU	Total CPU	Number of Variables	Number Of Constraints
MINSYS				1.73	653	1826
MULTICUT	10	0.46	1.01	1.47	77	217
MULTIDROP	10	0.46	1.98	2.43	77	217
SINGLECUT	343	19.37	162.07	181.44	77	343

nonlinear network algorithms such as Frank-Wolfe and simplicial decomposition methods [8] do not give the exact solution v^* and commodity flow vector x^{k*}, but, instead, give only an approximate total flow vector $\tilde{v}^*$. Due to rounding, there can also be small infeasibilities. The following LP both corrects the infeasibilities and gives the optimal commodity flow vectors.

$$
\begin{array}{lrcl}
\min & \multicolumn{3}{l}{\sum_{a\in\mathcal{A}} \epsilon_a + \varepsilon_a} \\
\text{s.t.} & & & \\
& \tilde{v}^* & = & \sum_{k\in\mathcal{K}} x^k - \epsilon + \varepsilon \\
& Ax^k & = & b_k, \quad \forall k \in \mathcal{K} \\
& x^k & \geq & 0, \quad \forall k \in \mathcal{K} \\
& \epsilon & \geq & 0 \\
& \varepsilon & \geq & 0.
\end{array}
$$

Given that $(\hat{x}^k, \hat{\epsilon}, \hat{\varepsilon})$ solves the LP above,

$$\hat{v} = \tilde{v}^* + \hat{\epsilon} - \hat{\varepsilon} = \sum_{k\in\mathcal{K}} \hat{x}^k$$

and $(\hat{v}, \hat{x}^k)$ replaces (v^*, x^{k*}) in the CPA.

Extensions

An immediate extension of the toll pricing framework for solving moderate size problems is for the elastic demand (ED) toll pricing problems. Below, we will present the user and system problems and then give the toll set.

We define a commodity by an origin, p, and a destination, q. Let $\mathcal{K}$ represent the set indexing of all such origin-destination pairs, $k = (p, q)$. The demand between an origin-destination pair is expressed as $t_k(c_k)$ where t_k is a nonnegative invertible demand function of c_k, the generalized cost of travel for commodity k. Denote the inverse of t_k by $w_k(t_k)$. Assume that $w_k(t_k)$ is monotonically decreasing. The system defining the feasible flows and demands is given by:

$$\begin{array}{rcll} v & = & \sum_{k\in\mathcal{K}} x^k & \\ Ax^k & = & E_k t_k & \forall k \in \mathcal{K} \\ x^k & \geq & 0 & \forall k \in \mathcal{K} \\ t_k & \geq & 0 & \forall k \in \mathcal{K}. \end{array}$$

Define V as the set of all feasible flows and demands.

For ED networks, the tolled user equilibrium problem, UOPT-ED is defined as [10]:

Find $(\bar{v}, \bar{t}) \in V$ such that

$$(s(\bar{v}) + \beta)^T (v - \bar{v}) - w(\bar{t})^T (t - \bar{t}) \geq 0 \quad \forall (v,t) \in V.$$

The goal of the system model is to maximize the net user benefit [10] which is the difference between the total network user benefit, $\sum_{k\in\mathcal{K}} \int_0^{t_k} w_k(z)dz$, and the system cost, $s(v)^T v$. We can state the system problem (SOPT-ED) as

$$\begin{array}{ll} \max & \sum_{k\in\mathcal{K}} \int_0^{t_k} w_k(z)dz - s(v)^T v \\ \text{s. t.} & (v,t) \in V. \end{array}$$

The elastic demand toll set is characterized by the following lemma [10]:

Lemma 2: $\mathcal{T} = W_{ED}(v^*, t^*)$, the polyhedron given by the β part of the following linear inequality system in β and ρ^k variables:

$$\begin{array}{rcll} s(v^*) + \beta & \geq & A^T \rho^k & \forall k \in \mathcal{K} \\ w_k(t_k^*) & \leq & E_k^T \rho^k & \forall k \in \mathcal{K} \\ (s(v^*) + \beta)^T v^* & = & w(t^*)^T t^* & \end{array}$$

where $(v^*, t^*) \in V$.
As it is in the fixed demand case, the MSCP tolls form a valid toll vector for elastic demand networks. Furthermore, the ED toll set has an interesting property: The total toll revenue $\beta^T v^* = w(t^*)^T t^* - s(v^*)^T v^*$, so it is constant, which, from Table 1, clearly does not hold for the FD toll pricing problems.

The toll pricing framework described in Section 3 can be extended easily to the elastic demand case, and, in fact, it simplifies because a suitable ϵ can be determined in Step 2 of the toll pricing framework from an approximate system solution. Specifically,

$$\tilde{\epsilon}^* = w(\tilde{t}^*)^T \tilde{t}^* - (s(\tilde{v}^*) + \tilde{\beta}_{MSCP})^T \tilde{v}^*$$

can be used and it is not necessary to solve the MINSYS linear program. This is because $\tilde{\beta}_{MSCP}$ is an approximate valid toll vector.

Define $W_{ED}(\tilde{v}^*, \tilde{t}^*, \tilde{\epsilon}^*)$ as

$$\begin{array}{rcl} s(\tilde{v}^*) + \beta & \geq & A^T \rho^k, \ \forall k \in \mathcal{K} \\ w_k(\tilde{t}_k^*) & \leq & E_k^T \rho^k, \ \forall k \in \mathcal{K} \\ w(\tilde{t}^*)^T \tilde{t}^* - (s(\tilde{v}^*) + \beta)^T \tilde{v}^* & \leq & \tilde{\epsilon}^*. \end{array}$$

Then, the toll pricing framework for the elastic demand TA problems can be stated as:

Step 1: Solve the system problem to obtain an approximately optimal solution $(\tilde{v}^*, \tilde{t}^*)$, $(s(\tilde{v}^*), w(\tilde{t}^*))$ and $\tilde{\beta}_{MSCP} = \nabla s(\tilde{v}^*)\tilde{v}^*$.

Step 2: Let

$$\tilde{\epsilon}^* = w(\tilde{t}^*)^T \tilde{t}^* - (s(\tilde{v}^*) + \tilde{\beta}_{MSCP})^T \tilde{v}^*$$

be the amount of violation in the aggregate complementary constraint.

Step 3: Define the toll set approximation $W_{ED}(\tilde{v}^*, \tilde{t}^*, \tilde{\epsilon}^*)$.

Step 4: Define and optimize an a toll pricing problem using this approximation.

Conclusion and Future Research

In this paper, we have defined an $\epsilon-$approximation of the toll pricing framework in [9] which solves moderate sized toll pricing problems.

We also present a cutting plane method for the MINSYS problem. Experimentations show that the multicut version seems to improve the solution time and therefore we have given a method to compute the optimal commodity flow vectors.

Note that these methods can be implemented using the algebraic modeling languages GAMS [6] and AMPL [2], and then mathematical programming solvers CPLEX [3], MINOS [11] and OSL [12] can be utilized to compute alternative toll vectors for both fixed and elastic demand traffic assignment problems. The resulting LPs usually can be solved in a reasonable amount of time. For example, on an IBM SP2, 300 Mhz computer with 512 GB of RAM using CPLEX 6.6, the fixed demand Stockholm Network takes 304 CPU seconds for the MINSYS problem and 204 CPU seconds for the MINMAX problem. However, the Stockholm MINTB problem, which has 962 binary variables (thus making it a very large mixed integer program), requires 131273 CPU seconds (36.5 hours). We are currently developing a Branch & Bound algorithm which takes into account the special structure of the LP relaxation of the MINTB problem.

As an alternative to the methods given here, there can be combinatorial methods for solving certain toll pricing problems. For example, Dial [4] and Hagstrom [7] have shown that the single commodity fixed demand MINSYS problem can be solved by using a max-path algorithm. Furthermore, Dial [5] extends this result to the multicommodity case. Comparison of these alternative methods is another topic of future experimentation.

References

[1] Abdulaal, M., LeBlanc, L.J., "Methods for combining Modal Split and Equilibrium Assignment Models," *Transportation Science*, 13, 292-314, 1979.

[2] Fourer, R., Gay, D. M., and Kernighan, B., "AMPL: A Modeling Language for Mathematical Programming," 1993.

[3] CPLEX, CPLEX Optimization, Inc., Incline Village, NV, 1996.

[4] Dial, R., "Minimal Revenue Congestion Pricing Part I: A Fast Algorithm for the Single-Origin Case," *Transportation Research-B*, Vol. 33, pp. 189-202, 1999.

[5] Dial, R., "Minimal Revenue Congestion Pricing Part II: A Efficient Algorithm for the General Case," *Transportation Research-B*, Vol. 34, pp. 645-665, 2000.

[6] GAMS, General Algebraic Modeling System, GAMS Development Corporation, 1995.

[7] Hagstrom, J. N., "Computing Tolls and Checking Equilibrium for Traffic Flows," *Working Paper*, University of Illinois at Chicago, 1998.

[8] Hearn, D. W., Lawphongpanich, S. and Ventura J. A., "Restricted Simplicial Decomposition: Computation and Extensions," *Mathematical Programming*, Vol. 31, pp. 99-118, 1987.

[9] Hearn, D. W. and Ramana, M. V., "Solving Congestion Toll Pricing Models," *Equilibrium and Advanced Transportation Modeling,* P. Marcotte and S. Nguyen (Eds.), Kluwer Academic Publishers, pp. 109-124, 1998.

[10] Hearn, D. W. and Yildirim, M. B., "A Toll Pricing Framework for Traffic Assignment Problems with Elastic Demand," To appear in the edited volume *Current Trends in Transportation and Network Analysis* papers in honor of Michael Florian, 2001.

[11] Murtagh, A. and Saunders, M. A., "MINOS 5.0 Users Guide," Stanford University, 1983.

[12] OSL, IBM Optimization Subroutine Library, http://www.research.ibm.com/osl/.

[13] Wardrop, J. G., "Some Theoretical Aspects of Road Traffic Research," *Proceedings of the Institute of Civil Engineers, Part II,* pp. 325-378, 1952.