Chapter 2

ON TRAFFIC EQUILIBRIUM MODELS WITH A NONLINEAR TIME/MONEY RELATION

Torbjörn Larsson
tl@mai.liu.se

Per Olov Lindberg
polin@mai.liu.se
Department of Mathematics, Linköping University
SE-581 83 Linköping, Sweden

Michael Patriksson
mipat@math.chalmers.se
Department of Mathematics, Chalmers University of Technology
SE-412 96 Gothenburg, Sweden

Clas Rydergren
clryd@mai.liu.se
Department of Mathematics, Linköping University
SE-581 83 Linköping, Sweden

Abstract We consider a traffic equilibrium problem in which each route has two attributes, time delay and monetary outlay, which are combined into a generalized time through a nonlinear relation. It is shown that this problem can be stated as a convex optimization model. Two simplicial decomposition type methods are proposed for its solution. The subproblem of these methods, which is a two-attribute shortest route problem, can be efficiently solved by the multi-labelling technique which has previously been applied to resource-constrained shortest path problems. Our numerical experiments show that both methods are feasible approaches to the equilibrium problem.

Keywords: Traffic equilibrium, nonlinear value-of-time

1. Introduction

In models for traffic equilibria one usually assumes that travellers choose the routes from their origins to their destinations that are preferred according to a cost criterion (the Wardrop principle, e.g., [10, Section 2.1], or [11]). The routes typically have several attributes, such as time, monetary outlays, distance etc. Usually these attributes are combined linearly to a *generalized cost*.

However, empirical studies, e.g., [7] and [6], indicate that travelers value travel time changes *nonlinearly* rather than linearly, in that short changes have lower value of time than longer ones.

At first sight, this might seem puzzling, but it is in fact easy to envisage situations where this is the case. Consider, for example, a young commuter heading for his workplace. Arriving a quarter of an hour late to work is probably then more than three times worse than coming five minutes late. Conversely, having to wake up a quarter of an hour earlier, similarly, is experienced to be more than three times worse than waking up five minutes earlier.

In the present paper, we will study a situation with two route attributes: *time* and *money*. In a subsequent paper, we will hopefully return to the case of more than two attributes. The two-attribute case has already been studied by Bernstein and Gabriel (B&G for short) in [4] and [2]. They assume that route choice is based on a (nonlinear) generalized cost. In particular they assume that the (generalized) cost for route r is

$$c_r = m_r + g_r(t_r), \tag{1}$$

with m_r and t_r being monetary outlay and travel time, respectively, and g_r the nonlinear value of time for route r.

To compute equilibria of their model, B&G utilize a specialized code based on a Gauss-Newton type method for nonlinear complementarity problems.

It is not a priori obvious why route choice should be based on (generalized) cost rather than on time. Thus, in the present paper we consider to instead base route choice on time rather than on cost,

$$\tau_r = t_r + G_r(m_r). \tag{2}$$

At first sight, the time-based cost (2) seems to be just a reformulation of the money-based one (1), with $g_r = G_r^{-1}$. It turns out, however, that they are not equivalent.

For the case with flow independent monetary outlays, the time based equilibrium problem can be stated as an equivalent optimization problem. This will give the benefit of usually faster computational methods as well as better control of convergence. The development of this equivalent optimization model and the techniques to solve it are the theme of this paper.

The paper is structured as follows. In Section 2 we state the time-based traffic equilibrium problem. Section 3 is devoted to the simplicial decomposition approaches to solve this problem; on the one hand a disaggregate version, where all route flows are stored, on the other hand an aggregate version, where only link flows are stored. In Section 4 we describe how to generate new routes to be used in the simplicial decomposition schemes, using a multi-label shortest path method [3]. Section 5, finally, is devoted to some computational testing on the example of B&G and on the classical Sioux Falls network.

2. The time-based traffic equilibrium problem

We will first introduce some further notation. The set of commodities (i.e., the origin-destination pairs) is denoted by $\mathcal{K}$, and d_k denotes the traffic demand for commodity k. (More general, k could denote a segment of travellers in a certain travel relation.) Let $\mathcal{R}^k$ be the set of simple routes for commodity k, and let $\mathcal{R} = \cup_{k \in \mathcal{K}} \mathcal{R}^k$ be the set of all routes. Similarly, let $\mathcal{R}_\ell \subseteq \mathcal{R}$ denote the routes that pass link $\ell \in \mathcal{L}$, with $\mathcal{L}$ being the set of links in the network. Let h_r denote the flow along route r, and let $t_\ell(v_\ell)$ be the travel time at link flow v_ℓ on link ℓ. Thus, the travel time for route r will be $t_r = \sum_{\ell \in r} t_\ell(v_\ell)$. Similarly, let $m_\ell \geq 0$ be the monetary outlay for link ℓ, which is assumed to be independent of the traffic flow. If the monetary outlay for route r is assumed to be additive, then $m_r = \sum_{\ell \in r} m_\ell$. (We can allow for more complex structures for the monetary outlay for a route. The important thing is that they are monotone and can be computed recursively.)

Travellers are assumed to base their route choice on some generalized times on the routes, and based on these times they always choose the minimal time route from origin to destination. The situation can be expressed by the well known equilibrium condition, the Wardrop principle (e.g., [10]),

$$\forall k \in \mathcal{K}, \forall r \in \mathcal{R}^k : h_r > 0 \quad \Longrightarrow \quad \tau_r(v) = \min_{p \in \mathcal{R}^k} \{\tau_p(v)\}, \tag{3}$$

where τ_r is the generalized time on route r.

If travel time is the only measure of generalized time, we have $\tau_r = t_r = \sum_{\ell \in r} t_\ell(v_\ell)$, and it is well known that the flows v and h that fulfill (3) can be obtained by solving the following convex optimization problem (see e.g., [10], Section 2.2).

$$\min_{h,v} \sum_{\ell \in \mathcal{L}} \int_0^{v_\ell} t_\ell(s) ds \tag{P}$$

subject to

$$v_\ell = \sum_{r \in \mathcal{R}_\ell} h_r, \qquad \ell \in \mathcal{L},$$

$$\sum_{r \in \mathcal{R}^k} h_r = d_k, \qquad k \in \mathcal{K},$$

$$h_r \geq 0, \qquad r \in \mathcal{R}$$

If we want to include a monetary cost component m_r for each route, and to express a situation with a nonlinear relation between travel time and monetary cost, we can express the generalized time on route r, for given link flows v, according to

$$\tau_r(v) = \sum_{\ell \in r} t_\ell(v_\ell) + G_k(m_r). \tag{4}$$

Since G_r, the time equivalent of money, probably is dependent on socio-economic factors, it is natural to assume that it depends not on r but on k_r, the commodity corresponding to r. The function G_k is naturally assumed to be nonnegative and increasing. The Wardrop conditions for the time-based traffic equilibrium problem are expressed by (3) using the definition of τ_r according to (4).

An equivalent optimization model for this problem can be formulated as

$$\min_{h,v} \sum_{\ell \in \mathcal{L}} \int_0^{v_\ell} t_\ell(s) ds + \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}^k} h_r G_k(m_r) \tag{P_D}$$

subject to

$$v_\ell = \sum_{r \in \mathcal{R}_\ell} h_r, \qquad \ell \in \mathcal{L},$$

$$\sum_{r \in \mathcal{R}^k} h_r = d_k, \qquad k \in \mathcal{K},$$

$$h_r \geq 0, \qquad r \in \mathcal{R}.$$

The only difference in comparison with problem (P) is the extra linear term in the objective. Thus, (P_D) is a convex problem, since (P) is. Further, since (P_D) has a compact feasible set it attains its minimum, and since the constraints are linear an Abadie constraint qualification is fulfilled, whence there are Lagrange multipliers such that the Karush–Kuhn–Tucker (KKT) conditions are satisfied (see [1], Theorem 5.1.3 and Lemma 5.1.4).

Letting γ_k be the multiplier for the k:th demand constraint and substituting the expression for v_ℓ into the objective, we get for $r \in \mathcal{R}^k$ the KKT condition

$$\sum_{\ell \in r} t_\ell(v_\ell) + G_k(m_r) - \gamma_k \geq 0,$$

with equality if $h_r > 0$. Thus

$$\gamma_k = \min_{r \in \mathcal{R}^k} \left\{ \sum_{\ell \in r} t_\ell(v_\ell) + G_k(m_r) \right\} = \min_{r \in \mathcal{R}^k} \{\tau_r(v)\}.$$

Hence, the KKT conditions state, for $r \in \mathcal{R}^k$, that $\tau_r(v) = \min_{p \in \mathcal{R}^k}\{\tau_p(v)\}$ if $h_r > 0$, that is (h, v, γ) fulfills the Wardrop conditions.

Conversely, assuming that (h, v) fulfills the Wardrop condition and letting $\gamma_k = \min_{r \in \mathcal{R}^k}\{\tau_r(v)\}$, it follows that (h, v, γ) fulfills the KKT conditions, and hence is an optimal solution to (P_D).

If the link travel time functions t_ℓ are strictly increasing, then the solution to the time-based traffic equilibrium problem has unique link flows. This follows from the fact that the objective of (P_D) is in this case strictly convex in the link flows.

In summary we have shown the following.

Proposition 1 (Optimization formulation) *The time-based traffic equilibrium problem admits an equivalent optimization formulation.* □

3. Solution approaches

In practice, the route set $\mathcal{R}$ is too large to be handled explicitly. Instead, one has to generate profitable routes systematically and iteratively. We propose to solve the problem (P_D) using either the disaggregate (DSD) or aggregate form (ASD) of the simplicial decomposition method (cf. [10, 8]). The DSD method, which alternates between a route generation phase and a master problem phase where the problem is solved using the routes generated so far, is analogous to the original DSD method for the standard traffic equilibrium problem proposed in [8].

Suppose we have generated a subset $\bar{\mathcal{R}} = \bigcup_{k \in \mathcal{K}} \bar{\mathcal{R}}^k$ of the routes, where $\bar{\mathcal{R}}^k$ is the subset of the routes for commodity k. We then solve a restriction of (P_D), replacing $\mathcal{R}$ with $\bar{\mathcal{R}}$. This problem (the master problem) is denoted $(\bar{P}_D)$, and it gives the solution $(\bar{h}, \bar{v})$, giving an upper bound to the optimal value of (P_D). Next, we want to find out whether we need to include more routes. This is accomplished through the solution of a subproblem, a linearized version of (P_D). The solution of this subproblem (the route generation problem) amounts to checking whether some non-generated route $r \in \mathcal{R}^k$ has a reduced cost less than zero, i.e., to check whether

$$\tau_r(\bar{v}) - \gamma_k = \sum_{\ell \in r} t_\ell(\bar{v}) + G_k(m_r) - \gamma_k < 0$$

where $\gamma_k = \min_{r \in \bar{\mathcal{R}}^k} \{\tau_r(\bar{v})\}$ is the multiplier for demand constraint k. Thus, we have to solve

$$\min_{r \in \mathcal{R}^k} \left\{ \sum_{\ell \in r} t_\ell(\bar{v}) + G_k(m_r) \right\},$$

which is to find the route with minimal generalized time, and then we have to check whether this generalized time is lower than γ_k. A solution procedure for finding the new routes is given in next section.

Since there are only finitely many routes for each O/D–pair we have the following result.

Proposition 2 (Convergence) *The DSD method will terminate with an optimal solution after a finite number route generations.* □

Similarly to classic traffic assignment problems, the shortest path subproblems in the route generation arises from a linearization of the objective of $(\bar{P}_D)$, e.g., ([10], Section 4.1.2). If we linearize the objective, then the cost coefficients will be flow independent, and all traffic can be sent along the shortest paths. Due to the convexity of the objective function of (P_D), the solution of the linearized problem yields a lower bound of the optimal value, as stated below.

Proposition 3 (Lower bound) *Let $\{r_k\}_{k \in \mathcal{K}}$ be the set of generated shortest paths in the route generation for $(\bar{P}_D)$. Then,*

$$\sum_{\ell \in \mathcal{L}} \int_0^{\bar{v}_\ell} t_\ell(s)ds - \sum_{\ell \in \mathcal{L}} t_\ell(\bar{v}_\ell)\bar{v}_\ell + \sum_{k \in \mathcal{K}} d_k \left(t_{r_k}(\bar{v}) + G_k(m_{r_k}) \right)$$

is a lower bound on the optimal value of (P_D). □

The lower bound together with the upper bound from solving the master problem $(\bar{P}_D)$ gives the possibility to terminate the algorithm when a prescribed error is achieved rather than when all routes with nonzero flows are generated.

In large networks, the DSD strategy may be too memory consuming. Hence, there is need for more aggregate versions. In classical (aggregate) simplicial decomposition (ASD) methods for traffic assignment, one takes convex combinations of link flow patterns arrived at by assigning all demand in each commodity to the shortest path obtained from the route generation subproblems (e.g., [10]). Exactly the same can be done in the current situation.

To be specific, assume that the shortest route for commodity k in the i'th iteration is r_k^i, and that this route has monetary outlay m_k^i. Assigning all demand in each commodity to the shortest route, we get the link flow pattern vector v^i, where for each link ℓ we have

$$v_\ell^i = \sum_{\{k|\ell \in r_k^i\}} d_k.$$

Further, let $G^i = \sum_{k\in\mathcal{K}} d_k G_k(m_{r_k^i})$ be the total time equivalent of the monetary outlays for this flow pattern. The aggregate master problem $(\bar{P}_A)$ for the time based traffic assignment problem is then to minimize the objective of the original problem (P_D) over convex combinations of generated link flow patterns up to the current iteration I, that is

$$\min_{h,v} \sum_{\ell\in\mathcal{L}} \int_0^{v_\ell} t_\ell(s)ds + \sum_{i=1}^{I} G^i \lambda_i \qquad (\bar{P}_A)$$

subject to

$$\begin{aligned} v_\ell = \sum_{i=1}^{I} \lambda_i v_\ell^i, &\qquad \ell \in \mathcal{L}, \\ \sum_{i=1}^{I} \lambda_i = 1, & \\ \lambda_i \geq 0, &\qquad i = 1, ..., I. \end{aligned}$$

This problem gives new link flows $\bar{v}_\ell$ and hence travel times $t_\ell(\bar{v}_\ell)$. Given these travel times, we solve a new route generation subproblem. Assigning all flow to the generated routes, we get a new link flow pattern v^{I+1}, etc.

The theoretical results for the ASD method are totally in parallel with those of the DSD method. The solution to $(\bar{P}_A)$, whose feasible set corresponds to a restriction of (P_D), gives an upper bound to the optimal value of (P_D).

Proposition 4 *The ASD method will terminate after a finite number of route generations.* □

Proof Follows from the fact that there are only a finite number of link flow patterns that can be generated, since they are composed of flows along a finite number of routes. □

The observation that the generated shortest paths solve a linearization of the true objective, is valid also for the aggregate master $(\bar{P}_A)$, the only difference being that we linearize at link flows that are convex combinations of generated link flow patterns.

In practice, the number of generated link flow patterns will become very large. Hence, if we have a limit on how many patterns we can store, we will have to drop patterns (possibly after aggregation), i.e., we will use restricted simplicial decomposition [5]. This will destroy the finite convergence, giving convergence in the limit.

4. A route generation algorithm

The minimization problem to solve for finding new routes is a generalized shortest path problem, where the links have multiple attributes, in our case travel time and monetary outlay. Due to the nonlinear transformation of money to time, it cannot be solved using standard procedures for shortest paths. In particular it does not adhere to the optimality principle of dynamic programming. If the "shortest" path from an origin to a destination passes node n, that does *not* imply that the sub-path from the origin to node n is the shortest path between the origin and node n. Our shortest path problem can however be solved using a multi-label shortest path method, known from the solution of resource constrained shortest path problems [3].

Each node gets labels (usually several) of the form (t, m, p), where t and m are the accumulated travel time and monetary outlay to get to the node from the origin via predecessor node (or or more generally, label) p. For a given origin, one starts by giving the origin node the label (0, 0, $-$). For each labeled node, one labels all nodes that can be reached from it. If for instance node n' can be reached via a link ℓ with time delay t_ℓ and monetary outlay m_ℓ from node n with label (t, m, p) then node n' gets the label $(t + t_\ell, m + m_\ell, n)$. Thus, the labels will indicate the total time and total monetary outlay to get to the nodes along paths indicated by the predecessor indices. The embryonic version of multi-label shortest path methods just described, will in fact generate all paths in a network. Possibly, a node can get one label for each predecessor. Since the number of labels in practice can be very large, we need to use domination tests to keep down the number of labels.

If in a given node there are two labels (t, m, p) and (t', m', p') with $t' \geq t$ and $m' \geq m$ then the label (t', m', p') can be deleted, since any path continuing from that label will be no better than the corresponding path from (t, m, p).

Since both travel time delays and monetary outlays for the links are nonnegative, domination implies that no path corresponding to the labels will contain cycles. Hence, the labeling process is finite. When the process stops, each destination node has received a set of labels (t_p, m_p, p). It is then easy to find the best label, i.e., the one minimizing $t_p + G_k(m_p)$, and the best O/D path can be traced using the predecessor indices. The multi-label shortest path method is summarized in the rudimentary pseudo code in Table 2.1.

Table 2.1. Multi-label shortest path method

Initialization:
  Give the origin O the label $(0, 0, -)$.
Scanning:
  While there are unscanned node labels.
    Choose an unscanned node label (t, m, p) at some node n.
    For each outgoing link ℓ at n do
      Give the terminal node n' of ℓ the label $(t + t_\ell(\bar{v}_\ell), m + m_\ell, n)$,
      possibly using domination to reduce the set of labels at n'.
    Mark the label (t, m, p) as scanned.
Route tracking:
  For each destination D, i.e., for each commodity k
    Find the label (t_p, m_p, p) at D that minimizes $t_p + G_k(m_p)$.
  Trace the corresponding path using the predecessor labels.

5. Numerical tests

We have tested the disaggregate as well as the aggregate version of the proposed simplicial decomposition method on two small test networks which are summarized in Table 2.2.

Network 1 is taken from [2]; we have used the inverse of their time-to-money transformation,

$$g_r(t_r) = \frac{t_r}{30} + 3\left(\frac{t_r}{30}\right)^2,$$

to describe the nonlinear relation between monetary outlay and travel time, i.e. $G_r = g_r^{-1}$. Moreover, since G&B in contrast to us use elastic demand, we have used their equilibrium demand as our demand.

Network 2 is a modification of the classical Sioux Falls network, where we have used the tolled links of [9] and the same function G as for Network 1.

Our experiments were not aiming at obtaining computational efficiency, but rather at verifying the validity of the algorithms. Hence, for both networks, and for both methods, we have solved the master problems to a quite small relative error ($10^{-4}\%$) before generating new routes. This accuracy bound is computed in a similar way as for the global accuracy. For the minimization in the master problem, we have used gradient projection in the DSD method and Newton's method in the ASD method. We initialize the algorithm by assigning all travel demand to the time optimal routes.

Table 2.2. Basic data for test networks.

	Network 1	Network 2
nodes	9	24
links	28	76
toll links	4	12
O/D–pairs	72	528

Table 2.3. Iteration history for Network 1. DSD method.

Iteration	# generated routes	gap (%)
0	72	10^{15}
1	51	4.5
2	8	$4.9 \cdot 10^{-5}$
3	0	$-$

Table 2.4. Iteration history for Network 2. DSD method.

Iteration	# generated routes	gap (%)
0	528	$2.5 \cdot 10^{12}$
1	394	15
2	177	.80
3	48	$1.1 \cdot 10^{-3}$
4	24	$9.7 \cdot 10^{-4}$
5	0	$-$

The disaggregate version terminates, for both test networks, in a few iterations, as shown in Tables 2.3 and 2.4. (Note that the relative gap, or error, in a given iteration is in fact not determined until the route generation of the next iteration.) The solution procedure generates 131 and 1175 routes for Network 1 and 2, respectively, and the number of routes with positive flow in the optimal solutions is 84 and 663 for Network 1 and 2, respectively.

In the tests of the aggregate version we have used non-restricted simplicial decomposition, i.e., no traffic patterns were discarded. As expected, convergence is much slower for the aggregate version. Figures 2.1 and 2.2 display the iteration histories, giving iteration-wise upper and lower bounds of the objective. We achieve a gap of 0.1% at iterations 23 and 45 for the network 1 and 2, respectively.

Gabriel and Bernstein give no computing times and do not mention how many routes they have generated. However, in order to compare their results to ours, we chose our money-to-time transform G as the inverse of the time-

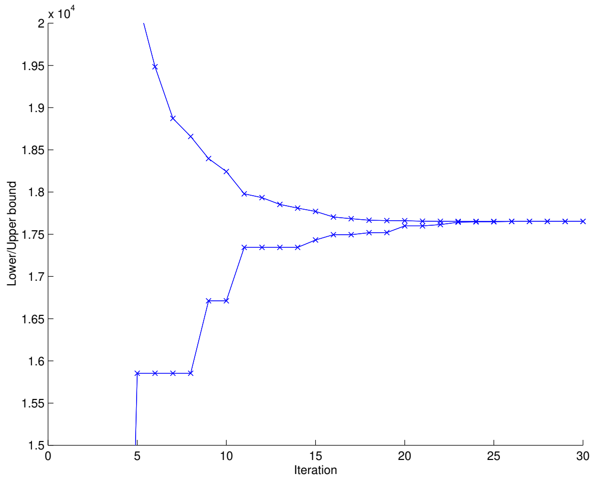

Figure 2.1. Iteration history for Network 1. ASD method.

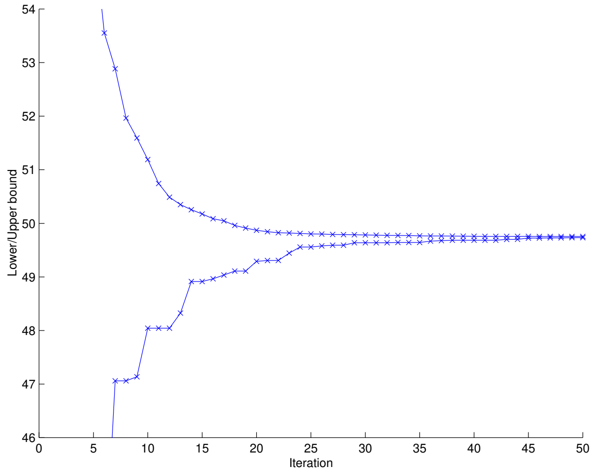

Figure 2.2. Iteration history for Network 2. ASD method

to-money transform g in [2]. To our initial surprise, the solutions did not coincide.

A closer look shows that this is in fact quite natural. Consider the example in Figure 2.3.

Using $g(t) = t^2$, we get a money-based cost of $c_r = m_r + (t_r(v))^2$ and a generalized time of $\tau_r = t_r(v) + \sqrt{m_r}$. It is then easy to verify that the money

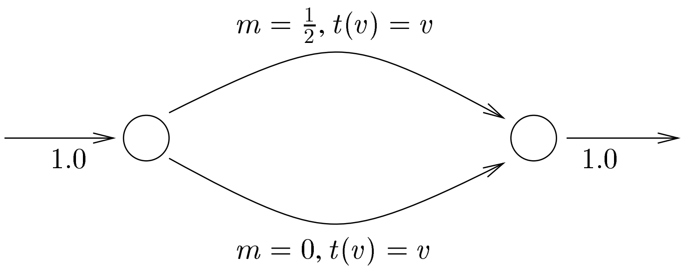

Figure 2.3. Example network.

based equilibrium is $v_1 = \frac{1}{4}$, $v_2 = \frac{3}{4}$, and that the time based equilibrium is $v_1 = \frac{2-\sqrt{2}}{4}$, $v_2 = \frac{2+\sqrt{2}}{4}$.

Observation 1 (Non-equivalence of time- and money-based equilibrium problems) *Evaluating routes in a money based way using generalized cost $\mu_r = m_r + g(t_k)$ does not necessarily give the same equilibrium as evaluating routes in a time based way, using generalized time $\tau_r = t_r + G(m_r)$, with $G = g^{-1}$.* □

In a later paper we will come back to how one can utilize the equivalent optimization formulation of the time based approach to solve money based problems and more general multi attribute traffic assignment problems.

References

[1] M.S. Bazaraa, H.D. Sherali and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, NY, 1993.

[2] D. Bernstein and S.A. Gabriel. Solving the nonadditive traffic equilibrium. In P.M. Pardalos, D.W. Hearn, and W. Hager, editors, *Network Optimization*, volume 450 of *Lecture Notes in Economics and Mathematical Systems*, pages 72–102. Springer–Verlag, 1997.

[3] J. Desrosiers, Y. Dumas, M.M. Solomon, and R. Soumis. Time constrained routing and scheduling. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbook in Operations Research and Management Science*, Network Models. North–Holland, 1995.

[4] S.A. Gabriel and D. Bernstein. The traffic equilibrium problem with nonadditive path costs. *Transportation Science*, 31:337–348, 1997.

[5] D.W. Hearn, S. Lawphongpanich, and J.A. Ventura. Restricted simplicial decomposition: Computation and extensions. *Mathematical Programming Study*, 31:99–118, 1987.

[6] D.A. Hensher and T.P. Truong. Valuation of travel savings. *Journal of Transport Economics and Policy*, pages 237–260, 1985.

[7] L. Hultkranz and R. Mortazavi. The value of travel time changes in a random nonlinear utility model. CTS working paper 1997:16., Submitted.

[8] T. Larsson and M. Patriksson. Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science*, 26:4–17, 1992.

[9] T. Larsson, M. Patriksson, and A-B. Strömberg. Ergodic, primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming*, 86:283–312, 1999.

[10] M. Patriksson. *The Traffic Assignment Problem – Models and Methods*. VSP, Utrecht, 1994.

[11] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice–Hall, Englewood Cliffs, NJ, 1985.